

УНИВЕРЗИТЕТ У БЕОГРАДУ

ГРАЂЕВИНСКИ ФАКУЛТЕТ

Никола Д. Обрадовић

**РАЗВОЈ “НАSP” КОНСТИТУТИВНОГ МОДЕЛА ЗА
ПРЕКОНСОЛИДОВАНЕ ГЛИНЕ СА ПРИМЕНОМ У
НУМЕРИЧКОЈ АНАЛИЗИ МЕТОДОМ КОНАЧНИХ
ЕЛЕМЕНАТА**

Докторска дисертација

Београд, 2024

UNIVERSITY OF BELGRADE
FACULTY OF CIVIL ENGINEERING

Nikola D. Obradović

**DEVELOPMENT OF “HASP” CONSTITUTIVE MODEL
FOR OVERCONSOLIDATED CLAYS WITH
APPLICATION IN FINITE ELEMENT NUMERICAL
ANALYSIS**

Doctoral dissertation

Belgrade, 2024

Никола Д. Обрадовић

РАЗВОЈ “NASP” КОНСТИТУТИВНОГ МОДЕЛА ЗА ПРЕКОНСОЛИДОВАНЕ ГЛИНЕ СА ПРИМЕНОМ У НУМЕРИЧКОЈ АНАЛИЗИ МЕТОДОМ КОНАЧНИХ ЕЛЕМЕНАТА

Ментор:

др Сања Јоцковић, доцент,
Универзитет у Београду, Грађевински факултет

Чланови комисије:

др Сања Јоцковић, доцент,
Универзитет у Београду, Грађевински факултет

др Снежана Мараш-Драгојевић, доцент,
Универзитет у Београду, Грађевински факултет

др Милош Марјановић, доцент,
Универзитет у Београду, Грађевински факултет

др Вељко Пујевић, доцент,
Универзитет у Београду, Грађевински факултет

др Лидија Здравковић, редовни професор,
Imperial College London, Faculty of Engineering, Department of Civil and Environmental
Engineering

Датум одбране

Захвалност

Докторска дисертација је резултат истраживања које сам спровео на Грађевинском факултету Универзитета у Београду.

Велику захвалност дугујем својим менторима, проф. др Мирјани Вукићевић и доц. др Сањи Јоцковић, на подршци, уложеном времену, стручности и саветима током истраживања и писања дисертације.

Захваљујем се колегама са Катедре за грађевинску геотехнику за сву подршку коју су ми пружили приликом израде докторске дисертације.

Највећу захвалност упућујем својој породици и пријатељима за стрпљење и разумевање током израде докторске дисертације.

Резиме

Описивање механичког понашања преконсолидованих глина и формулисање конститутивног модела који може да репродукује све карактеристике овог типа тла представља изазов у геотехничком инжењерству. HASP конститутивни модел, који је заснован на концепту критичног стања тла и концепту граничне површи, користи параметар стања као битан показатељ понашања тла и може да опише понашање преконсолидоване глине услед монотоног оптерећења у условима триаксијалног опита. Примена HASP модела у анализи граничних проблема, као и при описивању понашања глина услед цикличног оптерећења је ограничена. Због тога је неопходно проширивање релација HASP модела, како би овај модел могао да се користи за решавање практичних геотехничких проблема.

У оквиру ове дисертације, извршен је одабир релација за проширење HASP модела и формулисан је унапређени модел под називом HASP-E модел. Проширење модела извршено је тако да релације унапређеног модела буду релативно једноставне за имплементацију у рачунарски програм заснован на методи коначних елемената, да основни принципи на којима је развијен HASP модел буду задржани, да сви параметри модела буду јасно дефинисани и да се могу одредити из конвенционалних лабораторијских опита. Релације HASP модела су проширене на генерални напонски простор укључивањем треће напонске инваријанте ($Lode$ -овог угла). Утицај $Lode$ -овог угла је узет преко релација које описују облик површи течења и граничне површи у девијаторској равни тако да облик ових површи одговара експерименталним резултатима. За површ пластичног потенцијала је усвојен облик круга у девијаторској равни, тако да HASP-E модел користи неасоцијативан закон течења. Такође, релације HASP модела су проширене тако да је узета у обзир промена модула смицања са променом смичућих деформација, прикључењем "Brick" модела релацијама HASP модела.

Основни HASP модел и проширени HASP-E модел су имплементирани у рачунарски програм PLAXIS заснован на методи коначних елемената преко UDSM потпрограма написаних у рачунарском језику FORTRAN. За имплементацију оба конститутивна модела развијени су експлицитни и имплицитни алгоритми за нумеричку интеграцију конститутивних релација. Верификација поступка имплементације је извршена поређењем резултата симулације триаксијалних опита у PLAXIS-у 2D са резултатима за Euler-ову интеграцију деформација дуж познате путање напона и са резултатима предвиђања других расположивих конститутивних модела за преконсолидоване глине у PLAXIS-у. Извршена је анализа ефикасности, тачности и робустности развијених алгоритама. Према добијеним резултатима, експлицитни Euler-ов поступак са аутоматском субинкрементацијом и контролом грешке је најробустнији и најтачнији, иако захтева највећи утрошак рачунарског времена.

Валидација HASP-E модела је извршена кроз симулацију дренираних и недренираних цикличних триаксијалних опита и кроз примену конститутивног модела у анализи граничног проблема – понашање слоја преконсолидоване глине услед изградње насипа. Из приказаних резултата цикличних триаксијалних опита, HASP-E модел може да се користи за немонотона оптерећења, као што је растерећење тла и циклично оптерећење, може да опише хистерезисно понашање при растерећењу и поновном оптерећењу и да на адекватан начин опише величину хистерезиса. Такође, HASP-E

модел може у цикличним триаксијалним опитима да квалитативно и уз адекватну калибрацију материјалних параметара и квантитативно добро опише понашање тла. Из приказаних резултата симулације понашања преконсолидоване глине услед изградње насипа, HASP-E модел предвиђа релативно добро слегања тла, као и генерисане порне притиске. За разлику од MCC модела, HASP-E модел предвиђа понашања преконсолидоване глине у складу са резултатима мерења.

Кључне речи: конститутивни модел, преконсолидоване глине, HASP модел, Brick модел, крутост тла при веома малим деформацијама, MKE, PLAXIS, метода водећег параметра, циклично понашање, нумеричка интеграција

Научна област: Грађевинарство

Ужа научна област: Механика тла, Фундирање и Геоеколошко инжењерство

УДК:

Abstract

Describing the mechanical behavior of overconsolidated clays and formulating a constitutive model that can reproduce all the characteristics of this type of soil is a challenge in field of geotechnical engineering. The HASP constitutive model, developed in the framework of critical state soil mechanics and bounding surface concept, uses the state parameter as an important indicator of soil behavior and can describe the behavior of overconsolidated clay under monotonic loading under triaxial test conditions. The application of the HASP model in the analysis of boundary problems, as well as in describing the behavior of clay due to cyclic loading, is limited. Therefore, it is necessary to expand the relations of the HASP model, so that this model can be used to solve practical geotechnical problems.

Within this dissertation, new relations for the extension of the HASP model were selected and an improved model named HASP-E model was formulated. The criteria for selection of those relations were: new relations of the HASP-E model should be simple enough to be implemented in computer program based on finite element method, the basic principles of the HASP model should be retained, all model parameters should be clearly defined and that they can be determined from conventional laboratory tests. The dependency on the Lode angle is introduced in the HASP-E model in the expressions for yield surface and bounding surface. The shape of yield surface and bounding surface match the shapes determined from experimental results. The shape of plastic potential is the same as in the HASP model. As the different shapes for yield function and plastic potential are used, the flow rule of the HASP-E model is non-associated. Small strain stiffness is included in the HASP-E model by introducing the Brick small strain overlay model to the relations of the HASP model.

The basic HASP model and the HASP-E model are implemented in computer program PLAXIS based on finite element method via UDSM subroutines written in FORTRAN computer language. Explicit and implicit algorithms for numerical integration of constitutive relations have been developed. Verification of implementation procedures is performed by comparing the results of simulations of triaxial tests in PLAXIS 2D with the results of Euler's integration of deformations for known stress paths and with the predictions of other constitutive models for overconsolidated clay available in PLAXIS 2D. The developed algorithms are further analyzed in terms of their efficiency, accuracy, and robustness. According to the analysis results, the explicit Euler's procedure with automatic sub-incrementation and error control is the most accurate and robust although it requires the highest CPU times.

The validation of the HASP-E model is done against published results of cyclic triaxial drained and undrained tests and against published results of analysis of boundary value problem – overconsolidated clay layer behavior due to embankment loading. According to the results of the simulations of triaxial tests, the HASP-E model can be used for non-monotonic loading such as unloading and cyclic loading, it can reproduce hysteretic behavior due to unloading and reloading, and it is capable to reproduce the adequate size of hysteretic loop in one cycle of unloading/reloading. Also, the HASP-E model can qualitatively and quantitatively describe soil behavior in cyclic triaxial tests with adequate calibration of material parameters. From the presented results of the overconsolidated clay behavior simulation due to the construction of embankments, the HASP-E model predicts relatively well the soil settlement, as well as the generated pore pressures. Unlike the MCC model, the HASP-E model predicts the behavior of overconsolidated clay in accordance with the measurement results.

Keywords: constitutive modelling, overconsolidated clay, HASP constitutive model, Brick model, small strain stiffness, FEM, PLAXIS, governing parameter method, cyclic loading, numerical integration

Scientific field: Civil Engineering

Scientific subfield: Soil Mechanics, Foundation Engineering and Geoenvironmental Engineering

UDC:

Садржај

Садржај.....	I
Попис слика.....	V
Попис табела	XI
Попис симбола и скраћеница	XIII
Поглавље 1 Увод.....	1
1.1 Опште.....	1
1.2 Задаци и циљеви истраживања.....	2
1.3 Структура дисертације	3
Поглавље 2 Теоријске основе и примењене методе.....	5
2.1 Увод.....	5
2.2 Анализа напона	5
2.2.1 Принцип ефективних напона.....	9
2.3 Анализа деформација	9
2.4 Конститутивне релације за тло	13
2.4.1 Еластично понашање тла.....	13
2.4.2 Еласто-пластично понашање тла	14
2.4.3 Концепт критичног стања тла.....	16
2.5 Понашање преконсолидованих глина	19
2.6 Преглед конститутивних модела за глине	23
2.7 Нумеричке методе за решавање геотехничких проблема	25
2.7.1 Опште	25
2.7.2 Метод коначних елемената	25
2.7.3 Алгоритми за нумеричку интеграцију конститутивних релација	29
2.7.3.1 Експлицитне методе интеграције.....	29

2.7.3.2	Имплицитне методе интеграције.....	31
2.7.4	PLAXIS – рачунарски програм заснован на МКЕ за анализу геотехничких проблема.....	32
Поглавље 3 Проширени HASP конститутивни модел за преконсолидоване глине – HASP-E модел		37
3.1	Увод.....	37
3.2	Напонско-деформацијске релације HASP модела	38
3.2.1	Параметри HASP модела.....	42
3.3	Проширење HASP модела - HASP-E модел	47
3.3.1	Проширивање релација HASP модела на генерални напонски простор узимањем у обзир утицај треће напонске инваријанте (Lode-ов угао)	47
3.3.2	Проширивање релација HASP модела узимањем у обзир промене модула смицања са променом смичућих деформација	55
3.3.3	Напонско-деформацијске релације HASP-E модела	64
3.3.4	Параметри HASP-E модела	68
3.4	OC_CLAY конститутивни модел.....	69
Поглавље 4 Имплементација HASP и HASP-E модела		75
4.1	Увод.....	75
4.2	Интеграција конститутивних релација основне формулације HASP модела.....	75
4.2.1	Метода водећег параметра	75
4.2.1.1	Избор водећег параметра	76
4.2.1.2	Поступак тражења водећег параметра (нуле водеће функције)	77
4.2.1.3	Релације нумеричке интеграције HASP модела.....	81
4.2.1.4	Алгоритам нумеричке интеграције	85
4.2.2	Алгоритам (Ворја & Lee, 1990).....	87
4.2.2.1	Алгоритам нумеричке интеграције	90
4.2.3	Модификовани Euler-ов поступак са аутоматском субинкрементацијом и контролом грешке.....	92
4.2.3.1	Алгоритам нумеричке интеграције	100

4.2.4	Runge-Kutta-Dormand-Prince поступак са аутоматском субинкрементацијом и контролом грешке.....	102
4.2.4.1	Алгоритам нумеричке интеграције	106
4.3	Интеграција конститутивних релација HASP-E модела	108
4.3.1	Метода водећег параметра	110
4.3.1.1	Алгоритам нумеричке интеграције	112
4.3.2	Модификовани Euler-ов поступак са аутоматском субинкрементацијом и контролом грешке.....	114
4.3.2.1	Алгоритам нумеричке интеграције	119
4.3.3	Runge-Kutta-Dormand-Prince поступак са аутоматском субинкрементацијом и контролом грешке.....	120
4.3.3.1	Алгоритам нумеричке интеграције	121
4.4	Верификација поступка имплементације конститутивних модела.....	122
4.4.1	Модел триаксијалног опита у PLAXIS-у 2D	123
4.4.2	CU триаксијални опити на Cardiff глини	125
4.4.3	CD триаксијални опити на Fujinomori глини.....	132
4.4.4	CD циклични триаксијални опит на Fujinomori глини.....	139
4.4.5	CU циклични триаксијални опити на Newfield глини.....	141
4.5	Анализа ефикасности, тачности и робустности алгоритама за интеграцију конститутивних релација.....	144
4.5.1	Нумерички модел.....	144
4.5.2	Реултати и дискусија.....	147
Поглавље 5 Валидација HASP-E модела		157
5.1	Увод.....	157
5.2	CD циклични триаксијални опити на Fujinomori глини.....	157
5.2.1	Резултати и дискусија.....	159
5.3	CU циклични триаксијални опити на Newfield глини.....	175
5.3.1	Резултати и дискусија.....	176

5.4	Предвиђање понашања слоја глине услед изградње насипа – Saga Highway Embankment.....	187
5.4.1	Нумерички модел.....	188
5.4.2	Параметри конститутивних модела	189
5.4.3	Резултати и дискусија.....	192
5.5	Предвиђање понашања слоја глине услед изградње насипа – Teven Road Trial Embankment.....	198
5.5.1	Нумерички модел.....	199
5.5.2	Параметри конститутивних модела	200
5.5.3	Резултати и дискусија.....	202
Поглавље 6 Закључна разматрања и препоруке за даља истраживања		209
6.1	Закључна разматрања.....	209
6.1.1	Проширивање релација HASP модела и формулација HASP-E модела	209
6.1.2	Развој алгоритама за нумеричку интеграцију конститутивних релација.	210
6.1.3	Анализа ефикасности, тачности и робустности алгоритама за нумеричку интеграцију конститутивних релација.....	211
6.1.4	Валидација HASP-E модела	212
6.2	Препоруке за даља истраживања.....	213
Литература		215
Прилог 1 Структура UDSM за PLAXIS.....		225
Прилог 2 UDSM for PLAXIS – Fortran code for HASP model.....		229
Прилог 3 UDSM for PLAXIS – Fortran code for HASP-E model		291

Попис слика

Слика 2-1 Стање напона у тачки - преузето из (Maksimović, 2008)	6
Слика 2-2 Напонске инваријанте у простору главних напона – преузето из (Jocković, 2017).....	8
Слика 2-3 Понашање тла при изотрпоној компресији – преузето из (Jocković, 2017)	17
Слика 2-4 Путање напона у CU и CD опиту – преузето из (Vukićević & Jocković, 2019).....	18
Слика 2-5 Линија критичног стања у $p' - v$ равни - преузето (Vukićević & Jocković, 2019)	18
Слика 2-6 Гранична површ у $p' - q - e$ простору – преузето из (Schadlich & Schweiger, 2014).....	18
Слика 2-7 Комплетна гранична површ у нормализованој равни $p' - q$ – преузето из (Maksimović, 2008)	19
Слика 2-8 Карактеристично понашање преконсолидоване глине у дренираним условима – преузето из (Vukićević & Jocković, 2019).....	21
Слика 2-9 Путање ефективних напона у недренираном триаксијалном опиту – преузето из (Gu, et al., 2016).....	21
Слика 2-10 Промена модула смицања са порастом смичућих деформација (S – крива) – преузето из (Vukićević & Jocković, 2019).....	22
Слика 2-11 Newton-Raphson-ов итеративни поступак	29
Слика 3-1 (a) Концепт граничне површи; (b) Параметар стања – преузето из (Jocković & Vukićević, 2017)	39
Слика 3-2 Максимални градијент промене запремине – преузето из (Jocković, 2017) ...	40
Слика 3-3 Линија критичног стања у $p' - q$ дијаграму – преузето из (Vukićević & Jocković, 2019).....	43
Слика 3-4 Одређивање параметра Γ - адаптирано из (Bentley, 2019).....	45
Слика 3-5 Класични критеријуми лома у девијаторској равни – преузето из (Vukićević & Jocković, 2019).....	48
Слика 3-6 (Matsuoka & Nakai, 1974) и (Lade & Duncan, 1975) критеријуми лома – преузето из (Georgiadis, et al., 2004).....	49
Слика 3-7 SPM критеријум у девијаторској равни: (a) простор главних напона; (b) простор трансформисаних напона – преузето из (Matsuoka, et al., 1999)	52

Слика 3-8 Облик критеријума лома у девијаторској равни за различите односе M_e/M_c за усвојену зависност $M(\theta)$ и $Z = -0.229$	54
Слика 3-9 Облик критеријума лома у девијаторској равни за $\phi_C = \phi_E = \phi_{cv} = 30^\circ$ за: (a) усвојена зависност $M(\theta)$ и $Z = -0.229$, (b) LMN критеријум - израз (3-65), (c) (Zdravković, et al., 2020) са $Z = -0.1$, (d) (Sheng, et al., 2000) израз (3-67), (e) Mohr-Coulomb-ов хексагон.....	55
Слика 3-10 Аналогија особе која вуче за собом сет цигли помоћу неистегљивих канапа – преузето из (Simpson, 1992).....	59
Слика 3-11 Степенаста деградација крутости тла у “Brick” моделима – преузето из (Simpson, 1992).....	59
Слика 3-12 Илустрација особе која вуче цигле у HS-Brick моделу: (a) почетни положај цигли, (b) почетак деградације крутости када је један канап затегнут, (c) максимална деградација крутости јер су сви канапи затегнути, (d) промена правца кретања, сви канапи су опуштени и крутост поново има максималну вредност – преузето из (Cudny & Truty, 2020).....	61
Слика 3-13 Степенаста S-крива у HS-Brick моделу за $Nb = 10$ цигли – преузето из (Cudny & Truty, 2020).....	62
Слика 3-14 Облик површи течења и граничне површи у девијаторској равни за преконсолидовану глину и $\phi_C = \phi_E = \phi_{cv} = 30^\circ$ у HASP-E моделу.....	65
Слика 3-15 Степенаста S-крива у HASP-E моделу за $Nb = 20$ цигли, $G_{0ref} = 50$ МПа, $G_{urref} = 5$ МПа и $\gamma_{0,7} = 5 \cdot 10 - 4$. (Легенда: G_t - референтни тангентни модул; G_s - референтни секантни модул).....	67
Слика 3-16 Линија нормалне консолидације и рекомпресије у $\ln p' - \epsilon_v$ равни – преузето из (Bentley, 2019).....	70
Слика 3-17 Шематски приказ напонско-деформацијског стања у OC_CLAY моделу – преузето из (Bentley, 2019).....	71
Слика 3-18 Путање напона у $p' - q$ равни и напонско-деформације криве за лондонску глину при истој иницијалној вредности OCR за различите вредности параметра h - преузето из (Bentley, 2019).....	72
Слика 3-19 Промена генерисаних порних притисака са порастом аксијалних деформација за лондонску глину при истој иницијалној вредности OCR за различите вредности параметра h - преузето из (Bentley, 2019).....	72
Слика 4-1 Метода бисекције за тражење нуле функције – преузето из (Kojić & Bathe, 2003).....	78
Слика 4-2 Метода <i>regula falsi</i> – преузето из (Chapra & Canale, 2021).....	79
Слика 4-3 Newton-ов поступак тражења нуле функције – преузето из (Kojić & Bathe, 2003).....	79

Слика 4-4 Секантна метода за тражење нуле функције – преузето из (Chapra & Canale, 2021).....	80
Слика 4-5 Модел триаксијалног опита у PLAXIS-у 2D – преузето из (Surarak, et al., 2012)	124
Слика 4-6 Мрежа коначних елемената за модел триаксијалног опита у PLAXIS-у 2D..	125
Слика 4-7 Напонско-деформацијске криве за CU триаксијалне опите на Cardiff глини	128
Слика 4-8 Генерисани порни притисци за CU триаксијалне опите на Cardiff глини	131
Слика 4-9 Напонско-деформацијске криве за CD триаксијалне опите на Fujinomori глини	135
Слика 4-10 Запреминске деформације за CD триаксијалне опите на Fujinomori глини	138
Слика 4-11 Напонско-деформацијске криве за CD циклични триаксијални опит на Fujinomori глини	140
Слика 4-12 Запреминске деформације за CD циклични триаксијални опит на Fujinomori глини	141
Слика 4-13 Напонско-деформацијске криве за CU циклични триаксијални опит на Newfield глини.....	142
Слика 4-14 Генерисани порни притисци за CU циклични триаксијални опит на Newfield глини	143
Слика 4-15 Мрежа коначних елемената и гранични услови за модел тракастог темеља	145
Слика 4-16 Промена степена преконсолидације са порастом дубине у зависности од вредности параметра <i>POP</i>	147
Слика 4-17 Референтне криве зависности оптерећења и слегања тракастог темеља у недренираним условима за HASP и HASP-E модел за различите иницијалне услове	148
Слика 4-18 Тачност и ефикасност алгоритама за нумеричку интеграцију HASP и HASP-E модела за $POP = 100 \text{ kPa}$ и различит број инкремената	150
Слика 4-19 Тачност и ефикасност модификованог Euler-овог поступка са аутоматском субинкрементацијом за $POP = 100 \text{ kPa}$ у зависности од величине <i>STOL</i>	151
Слика 4-20 Тачност и ефикасност алгоритама за нумеричку интеграцију HASP и HASP-E модела у зависности од иницијалних услова за различит број инкремената	152
Слика 4-21 Гранична носивост тракастог темеља за $POP = 100 \text{ kPa}$ коју предвиђају HASP и HASP-E модел за различите алгоритме и величине инкремената	153

Слика 4-22 Гранична носивост тракастог темеља за $POP = 100 \text{ kPa}$ за модификовани Euler-ов поступак са аутоматском субинкрементацијом у зависности од величине $STOL$ за различите величине инкремента.....	154
Слика 4-23 Утицај иницијалних услова на срачунату граничну носивости за различите методе нумеричке интеграције и величине инкремената.....	155
Слика 5-1 Напонско-деформацијске криве за CD цикличне триаксијалне опите на Fujinomorі глини за различите конститутивне моделе и иницијални сет материјалних параметара	161
Слика 5-2 Запреминске деформације за CD цикличне триаксијалне опите на Fujinomorі глини за различите конститутивне моделе и иницијални сет материјалних параметара	164
Слика 5-3 Напонско-деформацијске криве за CD цикличне триаксијалне опите на Fujinomorі глини за HASP-E модел и различите вредности параметра λ	165
Слика 5-4 Запреминске деформације за CD цикличне триаксијалне опите на Fujinomorі глини за HASP-E модел и различите вредности параметра λ	166
Слика 5-5 Напонско-деформацијске криве за CD цикличне триаксијалне опите на Fujinomorі глини за HASP-E модел и различите вредности параметра κ	167
Слика 5-6 Запреминске деформације за CD цикличне триаксијалне опите на Fujinomorі глини за HASP-E модел и различите вредности параметра κ	168
Слика 5-7 Напонско-деформацијске криве за CD цикличне триаксијалне опите на Fujinomorі глини за HASP-E модел и различите вредности параметра G_{0ref}	169
Слика 5-8 Запреминске деформације за CD цикличне триаксијалне опите на Fujinomorі глини за HASP-E модел и различите вредности параметра G_{0ref}	170
Слика 5-9 Напонско-деформацијске криве за CD цикличне триаксијалне опите на Fujinomorі глини за HASP-E модел и различите вредности параметра $\gamma_{0,7}$	171
Слика 5-10 Запреминске деформације за CD цикличне триаксијалне опите на Fujinomorі глини за HASP-E модел и различите вредности параметра $\gamma_{0,7}$	172
Слика 5-11 Напонско-деформацијске криве за CD цикличне триаксијалне опите на Fujinomorі глини за HASP-E модел и усвојени сет материјалних параметара	174
Слика 5-12 Запреминске деформације за CD цикличне триаксијалне опите на Fujinomorі глини за HASP-E модел и усвојени сет материјалних параметара	175
Слика 5-13 Напонско-деформацијске криве за CU цикличне триаксијалне опите на Newfield глини за различите конститутивне моделе и иницијални сет материјалних параметара	178

Слика 5-14 Додатни порни притисци за CU цикличне триаксијалне опите на Newfield глини за различите конститутивне моделе и иницијални сет материјалних параметара	179
Слика 5-15 Напонско-деформацијске криве за CU цикличне триаксијалне опите на Newfield глини за HASP-E модел и различите вредности параметра λ	180
Слика 5-16 Додатни порни притисци за CU цикличне триаксијалне опите на Newfield глини за HASP-E модел и различите вредности параметра λ	181
Слика 5-17 Напонско-деформацијске криве за CU цикличне триаксијалне опите на Newfield глини за HASP-E модел и различите вредности параметра κ	182
Слика 5-18 Додатни порни притисци за CU цикличне триаксијалне опите на Newfield глини за HASP-E модел и различите вредности параметра κ	182
Слика 5-19 Напонско-деформацијске криве за CU цикличне триаксијалне опите на Newfield глини за HASP-E модел и различите вредности параметра G_{0ref}	183
Слика 5-20 Додатни порни притисци за CU цикличне триаксијалне опите на Newfield глини за HASP-E модел и различите вредности параметра G_{0ref}	184
Слика 5-21 Напонско-деформацијске криве за CU цикличне триаксијалне опите на Newfield глини за HASP-E модел и различите вредности параметра $\gamma_{0,7}$	184
Слика 5-22 Додатни порни притисци за CU цикличне триаксијалне опите на Newfield глини за HASP-E модел и различите вредности параметра $\gamma_{0,7}$	185
Слика 5-23 Напонско-деформацијске криве за CU цикличне триаксијалне опите на Newfield глини за HASP-E модел и усвојени сет материјалних параметара	186
Слика 5-24 Додатни порни притисци за CU цикличне триаксијалне опите на Newfield глини за HASP-E модел и усвојени сет материјалних параметара	187
Слика 5-25 Попречни пресек кроз насип са положајем мерних инструмената – адаптирано из (Chai, et al., 2013)	188
Слика 5-26 Мрежа коначних елемената и гранични услови за Saga Highway Embankment	189
Слика 5-27 Промена степена преко консолидације са порастом дубине у зависности од вредности параметра POP за Saga Highway Embankment	191
Слика 5-28 Слегање тла у тачки S_0 кроз време за различите методе нумеричке интеграције конститутивних релација	193
Слика 5-29 Додатни порни притисци у тачки P_1 кроз време за различите методе нумеричке интеграције конститутивних релација	194
Слика 5-30 Слегање тачака S_0 и S_1 кроз време за Saga Highway Embankment	194

Слика 5-31 Путање ефективних напона за HASP-E и MCC модел испод средине насипа за тачке на површини терена и на дубини од 2 m	195
Слика 5-32 Промена додатних порних притисака у тачкама <i>P1</i> , <i>P2</i> и <i>P3</i> кроз време за Saga Highway Embankment	196
Слика 5-33 Хоризонтална померања тла испод ножице насипа након завршетка изградње насипа	197
Слика 5-34 Хоризонтална померања тла испод ножице насипа две године након завршетка изградње насипа.....	197
Слика 5-35 Попречни пресек кроз насип са положајем мерних инструмената – адаптирано из (Huang, et al., 2006b).....	198
Слика 5-36 Мрежа коначних елемената и гранични услови за Teven Road Trial Embankment	199
Слика 5-37 Промена степена преконсолидације са порастом дубине у зависности од вредности параметра <i>POP</i> за Teven Road Trial Embankment.....	201
Слика 5-38 Слегање тла у тачки <i>A</i> кроз време за различите методе нумеричке интеграције конститутивних релација	203
Слика 5-39 Додатни порни притисци у тачки <i>PC2 – 1</i> кроз време за различите методе нумеричке интеграције конститутивних релација	203
Слика 5-40 Слегање тачака <i>A</i> и <i>B</i> кроз време за Teven Road Trial Embankment.....	204
Слика 5-41 Путање ефективних напона за HASP-E и MCC модел испод средине насипа за тачке на површини терена и на дубини од 2 m	205
Слика 5-42 Промена додатних порних притисака у тачкама <i>PC2 – 1</i> , <i>PC2 – 2</i> и <i>PC2 – 3</i> кроз време за Teven Road Trial Embankment.....	206
Слика 5-43 Слегање површине терена за Teven Road Trial Embankment.....	208

Попис табела

Табела 4-1 Иницијални услови за симулацију CU триаксијалних опита на Cardiff глини	126
Табела 4-2 Параметри HASP, HASP-E и OC_CLAY модела за симулацију CU триаксијалних опита на Cardiff глини	126
Табела 4-3 Девијаторски напон при лому у kPa за CU триаксијалне опите на Cardiff глини	131
Табела 4-4 Максимална вредност генерисаних порних притисака у kPa за CU триаксијалне опите на Cardiff глини	131
Табела 4-5 Иницијални услови за симулацију CD триаксијалних опита на Fujinomorі глини	132
Табела 4-6 Параметри HASP, HASP-E и OC_CLAY модела за симулацију CD триаксијалних опита на Fujinomorі глини	132
Табела 4-7 Вршни напонски однос $\eta f = qp'$ за CD триаксијални опит на Fujinomorі глини	138
Табела 4-8 Запреминска деформација у % при девијаторској деформацији од 20 % за CD триаксијални опит на Fujinomorі глини	139
Табела 4-9 Иницијални услови за симулацију CD цикличног триаксијалног опита на Fujinomorі глини	139
Табела 4-10 Параметри HASP-E и OC_CLAY модела за симулацију CD цикличног триаксијалног опита на Fujinomorі глини	139
Табела 4-11 Иницијални услови за симулацију CU цикличног триаксијалног опита на Newfield глини	142
Табела 4-12 Параметри HASP-E и OC_CLAY модела за симулацију CU цикличног триаксијалног опита на Newfield глини	143
Табела 4-13 Иницијални услови за одређивање носивости тракастог темеља на Newfield глини	146
Табела 4-14 Носивост тракастог темеља у недренираним условима и потребан утрошак рачунарског времена за “тачно” решење (RKDP поступак са аутоматском субинкрементацијом, $STOL = 10 - 6$ и $nsteps = 1000$)	149
Табела 5-1 Иницијални услови за симулацију три CD циклична триаксијална опита на Fujinomorі глини	157
Табела 5-2 Параметри HASP, HASP-E, OC_CLAY и MCC модела за симулацију три CD циклична триаксијална опита на Fujinomorі глини	158

Табела 5-3 Усвојени сет параметара HASP-E модела за симулацију три CD циклична триаксијална опита на Fujinomori глини	173
Табела 5-4 Иницијални услови за симулацију два CU циклична триаксијална опита на Newfield глини	175
Табела 5-5 Параметри HASP, HASP-E, OC_CLAY и MCC модела за симулацију два CU циклична триаксијална опита на Newfield глини.....	176
Табела 5-6 Усвојени сет параметара HASP-E модела за симулацију два CU циклична триаксијална опита на Newfield глини	185
Табела 5-7 Параметри конститутивних модела за нумеричку симулацију понашања глине услед изградње насипа – Saga Highway Embankment.....	190
Табела 5-8 Иницијални услови за нумеричку симулацију понашања глине услед изградње насипа – Saga Highway Embankment.....	190
Табела 5-9 Остали параметри нумеричког модела потребни за нумеричку симулацију понашања глине услед изградње насипа – Saga Highway Embankment.....	191
Табела 5-10 Параметри конститутивних модела за нумеричку симулацију понашања глине услед изградње насипа – Teven Road Trial Embankment.....	200
Табела 5-11 Иницијални услови за нумеричку симулацију понашања глине услед изградње насипа – Teven Road Trial Embankment.....	201
Табела 5-12 Остали параметри нумеричког модела потребни за нумеричку симулацију понашања глине услед изградње насипа – Teven Road Trial Embankment.....	202

Попис симбола и скраћеница

- \mathbf{a} вектор извода функције течења по напонима
- a величина која описује облик површи течења у OC_CLAY моделу
- $\mathbf{a}_p, \mathbf{a}_q, \mathbf{a}_{J3}$ помоћни вектори за рачунање вектора \mathbf{a} у експлицитним алгоритмима за HASP-E модел
- A величина потребна за рачунање еласто-пластичне матрице
- \mathbf{b} вектор извода функције пластичног потенцијала по напонима
- b величина која описује облик површи течења у OC_CLAY моделу
- B величина потребна за рачунање еласто-пластичне матрице
- \mathbf{B} матрица која садржи изводе интерполационих функција
- C_C индекс стишљивости
- C_k константа у релацији за промену водопропустљивости тла са променом коефицијента порозности
- C_R индекс рекомпресије
- C_1, C_2 константе за одређивање граница интервала водећег параметра
- CD консолидовани дренирани (триаксијални опит)
- CSL линија критичног стања
- $CSTOL$ толеранција за проверу услова да ли је достигнуто критично стање у методи водећег параметра
- CU консолидовани недренирани (триаксијални опит)
- d дилатанција
- $\det \mathbf{M}$ детерминанта матрице \mathbf{M}
- \mathbf{D}^e еластична конститутивна матрица
- \mathbf{D}^{ep} еласто-пластична конститутивна матрица
- e коефицијент порозности
- e_0 иницијални коефицијент порозности
- $elReload$ индикатор у OC_CLAY моделу да ли су деформације при поновном оптерећењу унутар граничне површи еластичне или еласто-пластичне

E	модул еластичности
E_{oed}^{ref}	референтни едометарски модул
E_{r_n}	релативна грешка при одређивању напона и параметра ојачања у експлицитним алгоритмима
E_{ur}	модул еластичности при растерећењу
E_{ur}^{ref}	референтни модул еластичности при растерећењу
EIPN	Euler-ова интеграција деформација дуж познате путање напона
E_Q	релативна грешка нумеричког решења
f	водећа функција
F	функција течења
$FTOL$	толеранција за проверу да ли је задовољен услов течења
G	модул смицања
G_0	модул смицања за веома мале деформације
G_0^{ref}	референтни модул смицања за веома мале деформације
G_t	тангентни модул смицања
G_{ur}	модул смицања при растерећењу
G_t^{ref}	референтни тангентни модул смицања
G_{ur}^{ref}	референтни модул смицања при растерећењу
h	параметар OC_CLAY модела
H	помоћна функција у имплицитном алгоритму (Borja & Lee, 1990)
$HTOL$	толеранција за налажење нуле функције H
HASP	H Ardening S tate P arameter конститутивни модел
HASP-E	H Ardening S tate P arameter E xtended модел
HS	Hardening Soil конститутивни модел
HSS	Hardening Soil Small конститутивни модел
I_1, I_2, I_3	инваријанте тензора напона
$I_{\varepsilon_1}, I_{\varepsilon_2}, I_{\varepsilon_3}$	инваријанте тензора деформација

J	Јакобијан матрица
$I'_{\varepsilon 1}, I'_{\varepsilon 2}, I'_{\varepsilon 3}$	инваријанте девијаторског дела тензора деформација
J_{1d}, J_{2d}, J_{3d}	инваријанте девијаторског дела тензора напона
J	друга инваријанта девијаторског дела напона
k	вектор параметара стања у функцији течења
k_h	коефицијент водопропустљивости тла у хоризонталном правцу
k_{max}	максималан број итерација у алгоритму за тражење нуле функције
k_v	коефицијент водопропустљивости тла у вертикалном правцу
K	запремински модул
K	матрица крутости система коначних елемената
K_E	матрица крутости једног коначног елемента
K_0^{NC}	коефицијент притиска тла у стању мировања за нормално консолидовано тло
K_0^{OC}	коефицијент притиска тла у стању мировања за преконсолидовано тло
K_s	запремински модул чврстих честица
K_u	запремински модул за тоталне напоне
K_{ur}	запремински модул при растерећењу
K_w	запремински модул воде
m	вектор параметара стања у функцији пластичног потенцијала
M	нагиб линије критичног стања у $p' - q$ равни
M_c	нагиб линије критичног стања у $p' - q$ равни за триаксијалну компресију
M_e	нагиб линије критичног стања у $p' - q$ равни за триаксијалну екстензију
M_J	нагиб линије критичног стања у $p' - J$ равни за триаксијалну компресију
MC	Mohr-Coulomb-ов конститутивни модел
MCC	Modified Cam Clay конститутивни модел
MKE	метод коначних елемената
MNR	модификовани Newton-Raphson-ов поступак
N	специфична запремина нормално консолидованог тла при $p' = 1 \text{ kPa}$

N_b	број цигли који се користи у “Brick” моделу
N_t	број затегнутих цигли у “Brick” моделу
OCR	overconsolidation ratio – степен преконсолидације
p'	средњи ефективни напон
\bar{p}'	средњи ефективни напон на граничној површи
p_{active}	активни порни притисци
p_{excess}	додатни порни притисци
p'_i	текућа вредност средњег ефективног напона
p'_{max}	претходна максимална вредност средњег ефективног напона
p'_0	параметар ојачања површи течења
\bar{p}'_0	параметар ојачања граничне површи
\hat{p}'_0	параметар ојачања одређен у експлицитном алгоритму из решење вишег реда
\tilde{p}'_0	међувредност параметра ојачања експлицитном RKDP алгоритму
p'_{ref}	референтни средњи ефективни напон
p_{steady}	устаљени порни притисци
p_w	порни притисци
P	функција пластичног потенцијала
POP	pre-overburden pressure – разлика напона преконсолидације и текуће вредности вертикалног ефективног напона
q	девијаторски напон
\bar{q}	девијаторски напон на граничној површи
q_{cyc}	циклични девијаторски напон
R	изотропни степен преконсолидације
RKDP	Runge-Kutta-Dormand-Prince
\mathbf{s}	девијаторски део тензора напона
s^b	дужина канапа који одговара b -тој цигли
S_{eff}	ефективни степен засићења

SMP	spatially mobilized plane
SSO	Small Strain Overlay модел
STOL	толеранција за локалну грешку у експлицитним алгоритмима
t	дебљина коначног елемента у раванским проблемима
T	псеудо време у експлицитним алгоритмима са субинкрементацијом
TOL	вредност толеранције за решавање неке нелинеарне једначине
UDSM	User-defined Soil Model
URL	линија бубрења
X, Y, Z	константе које описују облик површи течења у девијаторској равни
α_{Biot}	Biot-ов коефицијент порних притисака
β	помоћна величина у имплицитном алгоритму (Vogja & Lee, 1990)
γ	смичуће деформације
γ_{hist}	смичућа деформација која узима у обзир претходну историју оптерећивања
$\gamma_{0,7}$	смичуће деформације при којој је секантни модул смицања редукован на 70% иницијалне вредности
Γ	специфична запремина тла при $p' = 1 \text{ kPa}$ на линији критичног стања
$\Delta \mathbf{f}$	вектор прираштаја сила у чворовима мреже коначних елемената
$\Delta \mathbf{f}_E$	вектор прираштаја сила у чворовима једног коначног елемента
ΔT_n	псеудо време које одговара једном субинкременту
$\Delta \mathbf{t}$	површинске силе које делују на један коначни елемент
$\Delta \mathbf{u}$	вектор прираштаја померања у чворовима мреже коначних елемената
$\Delta \mathbf{u}_E$	вектор прираштаја померања у чворовима једног коначног елемента
$\Delta \boldsymbol{\varepsilon}$	прираштај вектора деформација
$\Delta \boldsymbol{\varepsilon}_v$	прираштај запреминских деформација
$\Delta \boldsymbol{\rho}$	запреминске силе које делују на један коначни елемент
$\Delta \boldsymbol{\sigma}$	прираштај вектора напона
$\Delta \omega_G^b$	процент деградације смичуће крутости који одговара једној цигли

ε	тензор деформација
ε^e	еластичне деформација
ε^p	пластичне деформација
ε^s	сферни део тензора деформација
ε'	девијаторски део тензора деформација
ε	дилатације
ε_q	дисторзијске деформације
ε_q^p	пластичне дисторзијске деформације
ε_q^e	еластичне дисторзијске деформације
ε_{Γ}	запреминска деформација на линији критичног стања при $p' = 1 \text{ kPa}$
ε_v	запреминске деформације
ε_v^p	пластичне запреминске деформације
ε_v^e	еластичне запреминске деформације
ζ	ознака за водећи параметар
η	напонски однос
θ	Lode-ов угао
κ	нагиб линије бубрења (растерећења)
κ^*	модификовани нагиб линије бубрења (растерећења) у OC_CLAY моделу
$\check{\kappa}$	нагиб линије бубрења (растерећења) у HASP-E моделу
λ	нагиб линије критичног стања и линије изотропне консолидације
λ^*	модификовани нагиб линије критичног стања и линије изотропне консолидације у OC_CLAY моделу
Λ	пластични мултипликатор
μ	Poisson-ово коефицијент
μ_u	Poisson-ово коефицијент за тоталне напоне
μ_{ur}	Poisson-ово коефицијент при растерећењу
ν	специфична запремина

ξ	максимална вредност дилатанције при лому у дренираним условима
ρ	релативна величина следећег субинкремента у односу на величину претходног субинкремента
σ	тензор напона
σ^s	сферни део тензора напона
$\hat{\sigma}$	тензор напона одређен у експлицитном алгоритму из решења вишег реда
$\tilde{\sigma}$	међувредност тензора напона у експлицитном RKDP алгоритму
σ	нормални напон
σ'_c	напон преконсолидације
σ'_{vi}	текући вертикални ефективни напон
$\bar{\sigma}$	средњи нормални напон
τ	смичући напон
ϕ'_c	угао смичуће отпорности тла при триаксијалној компресији у дренираним условима
ϕ'_{cv}	угао смичуће отпорности тла при константној запремини у дренираним условима
ϕ'_E	угао смичуће отпорности тла при триаксијалној екстензији у дренираним условима
Ψ	параметар стања за тренутну напонску тачку
Ψ^*	параметар стања за тренутну напонску тачку у OC_CLAY моделу
$\bar{\Psi}$	параметар стања за имагинарну напонску тачку
$\bar{\Psi}^*$	параметар стања за имагинарну напонску тачку у OC_CLAY моделу
ω	коефицијент ојачања
ω_G	укупна редукција смичуће крутости у “Brick” моделу

Поглавље 1 Увод

1.1 Опште

Савремено грађевинско инжењерство се не може замислити без геотехничког инжењерства. Све што човек направи, а да то не лети или не плови, ослања се на тло или стену, а то се посебно односи на грађевинске објекте као што су зграде, бране, путеви, тунели, канали, пристаништа, аеродроми, мостови и друго (Maksimović, 2008). Геотехничко инжењерство представља основу (или темељ) за практично све грађевинске објекте и има виталну улогу у осигуравању безбедности, стабилности и трајности грађевинских конструкција.

Тло у геотехничком инжењерству представља материјал на коме се, у коме се и од кога се граде грађевинске конструкције. У односу на друге материјале који се користе у грађевинарству, као што су челик и бетон, чији је процес производње строго контролисан са унапред познатим механичким карактеристикама, тло је материјал веома хетерогеног састава чије су механичке карактеристике за сваку локацију јединствене. Описивање механичког понашања тако разнородног материјала представља веома изазован задатак за инжењере геотехнике.

Механичко понашање тла и стене у разним напонско-деформацијским условима се описује преко конститутивних модела. Конститутивни модели представљају сет математичких релација које повезују напоне којима је изложен било који материјал и деформације које ти напони изазивају. Конститутивни модели омогућавају истраживачима и инжењерима да врше предикцију механичког понашања темељних конструкција, насипа, потпорних конструкција, испитају стабилност косина, да срачунају деформације у тлу, да процене носивост тла, као и да оптимизују геотехничке конструкције да буду одрживе, стабилне и безбедне. Механичко понашање тла је веома комплексно и још увек није формулисан конститутивни модел који може да опише све аспекте понашања тла.

Описивање механичког понашања преконсолидованих глина и формулисање конститутивног модела који може да репродукује све карактеристике понашања овог типа тла представља велики изазов у геотехничком инжењерству. Преконсолидоване глине су биле изложене већим напрезањима у прошлости него што је текуће оптерећење. Као последица тога, ове глине су збијеније и имају већу смичућу чврстоћу од нормално консолидованих глина. Механичко понашање преконсолидованих глина је под великим утицајем претходне историје напрезања. Карактерише их крто-пластични лом, односно смичући напон након достизања максималне вредности (вршна чврстоћа) опада при даљем деформисању до константне величине (омекшање). Развој тачних и поузданих конститутивних модела за описивање механичког понашања преконсолидованих глина је актуелна тема научних истраживања у области геотехнике.

Комплексна геометрија, разноврсни гранични услови и оптерећења ограничавају употребу аналитичких метода на веома једноставне проблеме. Да би конститутивни

модел могао да се користи за решавање практичних проблема и пројектовање геотехничких конструкција, неопходна је примена нумеричких метода као што су метод коначних елемената и метод коначних разлика. Примена нумеричких метода уз коришћење моћних модерних рачунара омогућава да се ефикасно анализирају веома комплексни геотехнички проблеми са великом тачношћу. Нумеричке методе дозвољавају геотехничким инжењерима да симулирају разноврсне сценарије и да тестирају различита иновативна пројектна решења како би се максимално оптимизовао процес пројектовања. Такође, мора се водити рачуна о ограничењима нумеричких метода. Избор адекватне нумеричке методе, тачност нумеричког поступка, квалитет улазних података као што су параметри тла, вештине и искуство геотехничког инжењера су фактори који утичу на поузданост и тачност нумеричког решења.

За описивање понашања преконсолидоване глине формулисан је HASP модел (Jocković, 2017). HASP модел је заснован на концепту критичног стања тла, концепту граничне површи и користи параметар стања као битан показатељ понашања тла. HASP модел превазилази многе недостатке MCC модела приликом описивања механичког понашања преконсолидованих глина уз задржавање једноставности MCC модела и истог броја материјалних параметара. HASP модел може да опише понашање преконсолидоване глине при монотоним оптерећењима у условима триаксијалног опита, али његова примена у анализи граничних проблема за троосна стања напона као и при описивању понашања глина услед цикличног оптерећења је ограничена.

Како би се унапредило предвиђање понашања преконсолидованих глина у анализи геотехничких конструкција, потребно је развијати напредне конститутивне моделе. Унапређењем HASP модела тако што би се прошириле релације модела на генерални напонски простор и узимањем у обзир промене модула смицања са променом смичућих деформација, значајно би се побољшала предвиђања понашања преконсолидованих глина у решавању граничних проблема и при цикличном оптерећењу. Истовремено са унапређењем конститутивног модела, потребно је развијати ефикасне, тачне и робустне алгоритме за имплементацију конститутивног модела у рачунарске програме засноване на методи коначних елемената. На тај начин би се значајно повећале могућности и примењивост HASP модела у решавању практичних геотехничких проблема.

1.2 Задаци и циљеви истраживања

Циљеви истраживања су:

- Проширивање релација HASP модела на генерални напонски простор, укључивањем треће напонске инваријанте (Lode-ов угао), како би се на адекватан начин дефинисао облик површи течења у девијаторској равни и значајно побољшало предвиђање модела у решавању раванских проблема.
- Проширивање релација HASP модела увођењем промене модула смицања са променом смичућих деформација, како би се омогућило да конститутивни модел може да опише понашање преконсолидованих глина при цикличном оптерећењу.

- Развој ефикасних, тачних и робустних алгоритама за нумеричку интеграцију конститутивних релација и имплементација у рачунарски програм заснован на методи коначних елемената.
- Верификација поступка имплементације конститутивног модела.
- Валидација конститутивног модела кроз поређење са експерименталним резултатима, као и кроз анализу граничних проблема.

Задаци истраживања проистичу из циљева истраживања:

- Преглед и анализа постојећих конститутивних модела за описивање понашања преконсолидованих глина који узимају у обзир адекватан услов лома и који могу да опишу понашање при цикличном оптерећењу.
- Избор релација за проширивање HASP модела.
- Формулација проширеног HASP модела.
- Развој алгоритама за нумеричку интеграцију основног и проширеног HASP модела.
- Имплементација основног и проширеног HASP модела у рачунарски програм PLAXIS
- Верификација поступка имплементације основног и проширеног HASP модела поређењем са резултатима других расположивих конститутивних модела за преконсолидоване глине у PLAXIS-у и са резултатима Euler-ове интеграције деформација дуж познате путање напона.
- Анализа ефикасности, тачности и робустности алгоритама за нумеричку интеграцију конститутивних релација.
- Валидација проширеног HASP модела кроз поређење са резултатима цикличних триаксијалних опита
- Валидација проширеног HASP модела кроз анализу граничних проблема и анализу са резултатима из литературе.
- Извођење закључака и препорука за даља истраживања.

1.3 Структура дисертације

Садржај докторске дисертације је изложен у шест поглавља. Након уводног разматрања које је признано у Поглављу 1 следи:

Поглавље 2. На почетку другог поглавља дат је преглед основних једначина механике континуума које се користе за формулисање конститутивних модела за тло. Након тога приказане су напонско-деформацијске релације које описују еластично и еласто-пластично понашање тла, као и приказ концепта критичног стања тла. Следи преглед понашања преконсолидованих глина и одговарајућих конститутивних модела. На крају овог поглавља дат је кратак опис методе коначних елемената као нумеричке методе која се најчешће користи за решавање геотехничких проблема, алгоритама за имплементацију конститутивних модела и рачунарског програма PLAXIS који је заснован на овој нумеричкој методи.

Поглавље 3. На почетку трећег поглавља дат је детаљан приказ напонско-деформацијских релација и параметара основне формулације HASP модела. У наставку

је приказана проширена и унапређена формулација HASP модела под називом HASP-E модел. Описане су релације на основу којих је извршено проширење основне формулације HASP модела на генерални напонски простор, релације преко којих је узета у обзир крутост тла у подручју веома малих и малих деформација, као и начин на који се може узети у обзир хистерезисно понашање тла. Критеријуми на основу којих је извршено унапређење HASP модела су: релације проширеног конститутивног модела треба да буду релативно једноставне за имплементацију у рачунарски програм заснован на МКЕ, да основни принципи на којима је развијен HASP модел буду задржани, да сви параметри модела буду јасно дефинисани и да се могу одредити из конвенционалних лабораторијских опита. На крају поглавља дат је опис ОС_CLAY модела (Bentley, 2019) који је развијен из основне формулације HASP модела са освртом на разлике у односу на HASP и HASP-E моделе.

Поглавље 4. У првој половини овог поглавља приказани су алгоритми за нумеричку интеграцију конститутивних релација HASP и HASP-E модела. Имплементација HASP модела је извршена методом водећег параметра, имплицитним алгоритмом (Borja & Lee, 1990), модификованим Euler-овим поступком са аутоматском субинкрементацијом и контролом грешке и Runge-Kutta-Dormand-Prince поступком са аутоматском субинкрементацијом и контролом грешке. Имплементација HASP-E модела извршена је методом водећег параметра, модификованим Euler-овим поступком са аутоматском субинкрементацијом и контролом грешке и Runge-Kutta-Dormand-Prince поступком са аутоматском субинкрементацијом и контролом грешке. У другој половини поглавља приказана је верификација поступка имплементације конститутивних модела и анализа ефикасности, тачности и робустности алгоритама за нумеричку интеграцију конститутивних релација. Верификација поступка имплементације је извршена поређењем резултата симулације триаксијалних опита у PLAXIS-у 2D са резултатима за Euler-ову интеграцију деформација дуж познате путање напона (EIPN) и са резултатима предвиђања ОС_CLAY модела. Анализа ефикасности, тачности и робустности алгоритама за интеграцију конститутивних релација је извршена на примеру једног граничног проблема у програму PLAXIS 2D.

Поглавље 5: У овом поглављу извршена је валидација HASP-E модела поређењем публикованих резултата лабораторијских опита или теренских мерења са резултатима нумеричког модела. У првој половини поглавља, приказани су резултати нумеричких симулација цикличних дренираних триаксијалних опита на Fujiomogi глини и цикличних недренираних триаксијалних опита на Newfield глини. Такође, извршена је параметарска анализа са циљем да се утврди како и колико промене материјалних параметара HASP-E модела утичу на предвиђање понашања глине. У другој половини поглавља приказане су могућности HASP-E модела да предвиди понашање слоја преконсолидоване глине услед изградње насипа. Резултати предвиђања HASP-E модела су упоређена са мерењима и са предвиђањима MCC модела.

Поглавље 6: У завршном поглављу је дат резиме истраживања са закључцима и наведене су препоруке за даља истраживања.

Поглавље 2 Теоријске основе и примењене методе

2.1 Увод

Предвиђање понашања тла као средине на и у којој се граде различите грађевинске конструкције представља веома битан део геотехничког инжењерства. Са аспекта механичког понашања, тло спада у врло комплексне материјале. Због тога, развој напредних конститутивних модела који могу да опишу различите аспекте понашања тла је веома значајан задатак за истраживаче.

Описивање понашања тла се заснива на основним принципима механике континуума и отпорности материјала. Конститутивни модел представља скуп математичких једначина које описују напонско-деформацијско понашање неког материјала. Што је понашање материјала комплексније, то су и релације конститутивног модела сложене, са великим бројем параметара који су потребни за његово адекватно описивање. Тло је хетероген и анизотропан грануларан материјал са порама које могу бити потпуно или делимично испуњене водом и ваздухом, што га чини комплексним материјалом са становишта напонско-деформацијског понашања, а његово конститутивно моделирање веома захтевним задатком.

У оквиру овог поглавља дат је преглед основних једначина механике континуума које се користе за формулацију конститутивних модела за тло, релација које описују еластично и еласто-пластично понашање тла, као и приказ концепта критичног стања тла. Након тога, дат је опис понашања и преглед конститутивних модела за преконсолидоване глине. На крају поглавља описане су нумеричке методе за решавање геотехничких проблема са кратким приказом методе коначних елемената и алгоритмима за имплементацију конститутивних модела, као и кратак опис рачунарског програма PLAXIS.

Напомена: У дисертацији ће се користити Voigt-ова нотација за представљање симетричних тензора. Тензори другог реда, као што су тензори напона и деформација, представљаће се у облику вектора. Тензори четвртог реда, као што је тензор еластичне крутости материјала, представљаће се у облику матрица.

2.2 Анализа напона

Стање напона у једној тачки је потпуно одређено ако су познати вектори напона у Descartes-овом координатном систему у три међусобно управне равни које пролазе кроз ту тачку. Скалари $\sigma_{xx}, \tau_{xy}, \dots, \tau_{zy}$ се називају компонентални напони и чине их три нормална напона (ознака σ) и шест смичућих напона (ознака τ). Скуп компоненталних напона се назива тензором напона σ и може се представити у матричном облику:

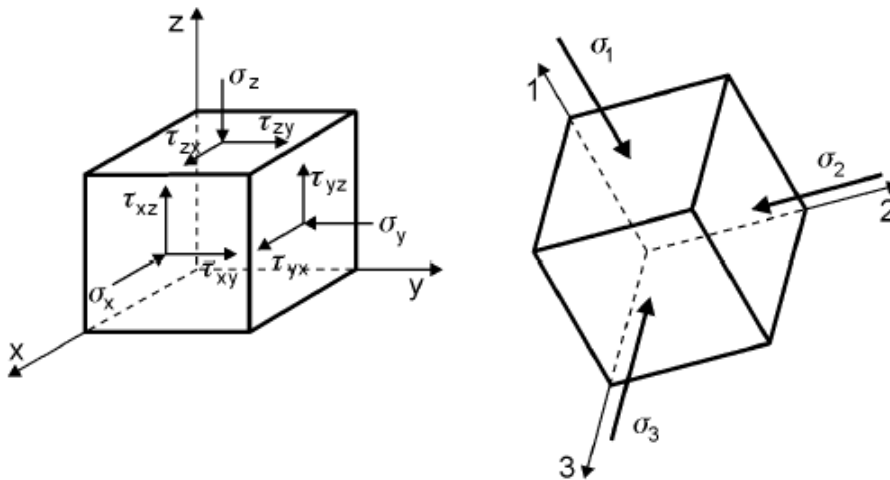
$$\sigma = \begin{bmatrix} \sigma_{xx} & \tau_{xy} & \tau_{xz} \\ \tau_{yx} & \sigma_{yy} & \tau_{yz} \\ \tau_{zx} & \tau_{zy} & \sigma_{zz} \end{bmatrix} \quad 2-1$$

Тензор σ представља тензор другог реда. Уколико се за координатне осе усвоје ознаке x_1, x_2, x_3 може се представити и у следећем облику:

$$\sigma = \begin{bmatrix} \sigma_{11} & \tau_{12} & \tau_{13} \\ \tau_{21} & \sigma_{22} & \tau_{23} \\ \tau_{31} & \tau_{32} & \sigma_{33} \end{bmatrix} \quad 2-2$$

Ако се компонентални нормални напони означе као $\sigma_x, \sigma_y, \sigma_z$, тензор напона се може представити и у следећем облику (Слика 2-1):

$$\sigma = \begin{bmatrix} \sigma_x & \tau_{xy} & \tau_{xz} \\ \tau_{yx} & \sigma_y & \tau_{yz} \\ \tau_{zx} & \tau_{zy} & \sigma_z \end{bmatrix} \quad 2-3$$



Слика 2-1 Стање напона у тачки - преузето из (Maksimović, 2008)

У механици тла усвојена је конвенција да су нормални напони притиска позитивни. Из става о конјугованости сличућих напона важи $\tau_{xy} = \tau_{yx}, \tau_{yz} = \tau_{zy}, \tau_{zx} = \tau_{xz}$, тако да је тензор напона симетричан и стање напона у тачки одређено је са шест независних компонента напона. Тензор напона се онда може написати и у облику вектора:

$$\sigma = (\sigma_x \quad \sigma_y \quad \sigma_z \quad \tau_{xy} \quad \tau_{yz} \quad \tau_{zx})^T \quad 2-4$$

У посматраној тачки постоје равни у којима нормални напони достижу екстремне вредности и сличући напони су једнаки нули. Екстремне вредности главних напона називају се и главни напони. Може се показати да постоје три такве главне равни које су међусобно управне и у којима су напони смицања једнаки нули. Такве равни се називају равни главних напона. Главни напони се обележавају са σ_1, σ_2 и σ_3 , тако да важи $\sigma_1 \geq \sigma_2 \geq \sigma_3$. Тензор напона тада има облик:

$$\sigma = \begin{bmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & \sigma_3 \end{bmatrix} \quad 2-5$$

Главни напони представљају својствене вредности тензора напона σ и могу се добити решавањем карактеристичне једначине:

$$\sigma_K^3 - I_1 \sigma_K^2 + I_2 \sigma_K - I_3 = 0 \quad 2-6$$

где су I_1, I_2 и I_3 прва, друга и трећа напонска инваријанта:

$$I_1 = \sigma_x + \sigma_y + \sigma_z \quad 2-7$$

$$I_2 = \sigma_x \sigma_y + \sigma_y \sigma_z + \sigma_z \sigma_x - \tau_{xy}^2 - \tau_{yz}^2 - \tau_{zx}^2 \quad 2-8$$

$$I_3 = \det \boldsymbol{\sigma} = \sigma_x \sigma_y \sigma_z + 2\tau_{xy} \tau_{yz} \tau_{zx} - \sigma_x \tau_{yz}^2 - \sigma_y \tau_{zx}^2 - \sigma_z \tau_{xy}^2 \quad 2-9$$

Тензор напона се може разложити на сферни $\boldsymbol{\sigma}^{(s)}$ и девијаторски део \mathbf{s} :

$$\boldsymbol{\sigma} = \boldsymbol{\sigma}^{(s)} + \mathbf{s} \quad 2-10$$

$$\boldsymbol{\sigma}^{(s)} = \begin{bmatrix} \bar{\sigma} & 0 & 0 \\ 0 & \bar{\sigma} & 0 \\ 0 & 0 & \bar{\sigma} \end{bmatrix} \quad 2-11$$

$$\mathbf{s} = \begin{bmatrix} \sigma_x - \bar{\sigma} & \tau_{xy} & \tau_{xz} \\ \tau_{yx} & \sigma_y - \bar{\sigma} & \tau_{yz} \\ \tau_{zx} & \tau_{zy} & \sigma_z - \bar{\sigma} \end{bmatrix} \quad 2-12$$

где је $\bar{\sigma}$ средњи нормални напон:

$$\bar{\sigma} = \frac{1}{3}(\sigma_x + \sigma_y + \sigma_z) = \frac{1}{3}I_1 \quad 2-13$$

Услед дејства сферног дела тензора напона мења се само запремина елемента тла без промене облика, док се услед дејства девијаторског дела тензора напона мења само облик елемента тла без промене запремине. Девијаторски део тензора напона се може написати у облику вектора:

$$\mathbf{s} = (\sigma_x - \bar{\sigma} \quad \sigma_y - \bar{\sigma} \quad \sigma_z - \bar{\sigma} \quad \tau_{xy} \quad \tau_{yz} \quad \tau_{xz})^T \\ = (s_{11} \quad s_{22} \quad s_{33} \quad s_{12} \quad s_{23} \quad s_{31})^T \quad 2-14$$

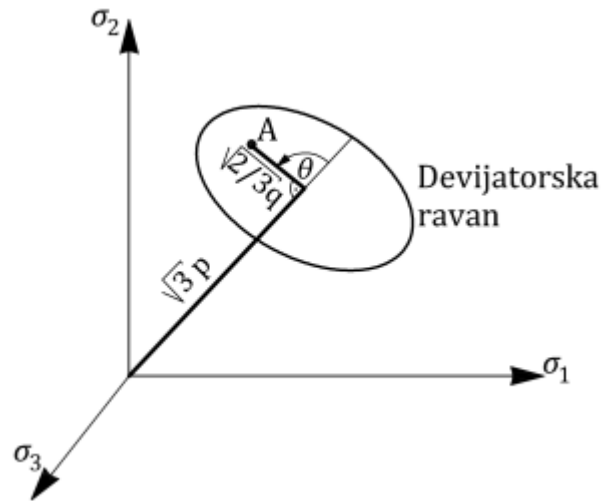
Као за тензор напона, напонске инваријанте за девијаторски део тензора напона су:

$$J_{1d} = s_{11} + s_{22} + s_{33} \quad 2-15$$

$$J_{2d} = s_{11}s_{22} + s_{22}s_{33} + s_{33}s_{11} - s_{12}^2 - s_{23}^2 - s_{31}^2 \quad 2-16$$

$$J_{3d} = \det \mathbf{s} = s_{11}s_{22}s_{33} + 2s_{12}s_{23}s_{31} - s_{11}s_{23}^2 - s_{22}s_{31}^2 - s_{33}s_{12}^2 \quad 2-17$$

Поред претходно наведених напонских инваријанти, у механици тла је погодније користити следеће напонске инваријанте: средњи напон p , девијаторски напон q и Lode-ов угао θ . Значење ових напонских инваријанти дато је на **Слика 2-2**.



Слика 2-2 Напонске инваријанте у простору главних напона – преузето из (Јосковић, 2017)

У простору главних напона, средњи напон p представља растојање дуж главне дијагонале (хидростатичка оса на којој важи $\sigma_1 = \sigma_2 = \sigma_3$) од координатног почетка до тренутне девијаторске равни. Девијаторска раван је било која раван која је управна на главну дијагонулу у простору главних напона. Девијаторски напон q представља меру удаљености тренутне напонске тачке и главне дијагонале у девијаторској равни. Лодеов угао θ дефинише оријентацију тренутне напонске тачке у девијаторској равни. Лодеов угао може бити у границама од $\theta = -30^\circ$ (што одговара условима триаксијалне компресије) до $\theta = +30^\circ$ (што одговара условима триаксијалне екстензије). Претходно описане напонске инваријанте могу се одредити преко израза:

$$p = \frac{1}{3}(\sigma_x + \sigma_y + \sigma_z) = \frac{1}{3}I_1 = \bar{\sigma} \quad 2-18$$

$$q = \sqrt{3J_{2d}} = \sqrt{\frac{1}{2}[(\sigma_x - \sigma_y)^2 + (\sigma_y - \sigma_z)^2 + (\sigma_z - \sigma_x)^2 + 6(\tau_{xy}^2 + \tau_{yz}^2 + \tau_{zx}^2)]} \quad 2-19$$

$$\theta = -\frac{1}{3} \arcsin\left(\frac{27J_{3d}}{2q^3}\right) \quad 2-20$$

где се J_{3d} може одредити преко израза:

$$J_{3d} = (\sigma_x - p)(\sigma_y - p)(\sigma_z - p) + 2\tau_{xy}\tau_{yz}\tau_{zx} - (\sigma_x - p)\tau_{yz}^2 - (\sigma_y - p)\tau_{zx}^2 - (\sigma_z - p)\tau_{xy}^2 \quad 2-21$$

Главни напони се могу изразити преко инваријанти p , q и θ преко израза:

$$\sigma_1 = p + \frac{2}{3}q \cdot \sin\left(\theta + \frac{2}{3}\pi\right) \quad 2-22$$

$$\sigma_2 = p + \frac{2}{3}q \cdot \sin(\theta) \quad 2-23$$

$$\sigma_1 = p + \frac{2}{3}q \cdot \sin\left(\theta - \frac{2}{3}\pi\right) \quad 2-24$$

2.2.1 Принцип ефективних напона

У складу са Terzaghi-јевим принципом ефективних напона, напони у тлу се могу поделити на ефективне напоне σ' и порне притиске σ_w :

$$\sigma = \sigma' + \sigma_w \quad 2-25$$

Порни притисак потиче од воде под притиском у порама тла и има исти интензитет p_w у свим правцима. Вода у порама не може да прими смичуће напоне, тако да се вектор порних притисака може написати у облику:

$$\sigma_w = (p_w \quad p_w \quad p_w \quad 0 \quad 0 \quad 0)^T \quad 2-26$$

Као последица тога да вода не може да прими смичуће напоне, ефективни смичући напони су једнаки тоталним смичућим напонима. Вектор ефективних напона може се онда написати у следећем облику:

$$\sigma' = (\sigma'_x \quad \sigma'_y \quad \sigma'_z \quad \tau_{xy} \quad \tau_{yz} \quad \tau_{zx})^T \quad 2-27$$

Промене запремине, облика и смичуће чврстоће зависе искључиво од ефективних напона због чега се велики број конститутивних модела заснива на принципу ефективних напона и релације тих модела су изражене преко ефективних напона. У наставку текста, све релације су дате за ефективне напоне и под термином напон подразумева се ефективни напон.

2.3 Анализа деформација

Стање деформација тла у некој тачки се може описати преко компонената деформација и потпуно је одређено ако су познате компоненте деформација за три некомпланарна правца кроз ту тачку. Скуп компонената деформација представља тензор деформација који се може представити у матричном облику:

$$\boldsymbol{\varepsilon} = \begin{bmatrix} \varepsilon_{xx} & \varepsilon_{xy} & \varepsilon_{xz} \\ \varepsilon_{yx} & \varepsilon_{yy} & \varepsilon_{yz} \\ \varepsilon_{zx} & \varepsilon_{zy} & \varepsilon_{zz} \end{bmatrix} \quad 2-28$$

Уколико се за координатне осе усвоје ознаке x_1, x_2, x_3 , тензор деформација се може представити и у следећем облику:

$$\boldsymbol{\varepsilon} = \begin{bmatrix} \varepsilon_{11} & \varepsilon_{12} & \varepsilon_{13} \\ \varepsilon_{21} & \varepsilon_{22} & \varepsilon_{23} \\ \varepsilon_{31} & \varepsilon_{32} & \varepsilon_{33} \end{bmatrix} \quad 2-29$$

Деформације представљају изводе компоненталних померања u_x, u_y и u_z :

$$\varepsilon_{ij} = \frac{1}{2} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \quad 2-30$$

Деформације могу бити дилатације и клизања. Дилатације се одређују према:

$$\varepsilon_{xx} = \frac{\partial u_x}{\partial x} \quad 2-31$$

$$\varepsilon_{yy} = \frac{\partial u_y}{\partial y} \quad 2-32$$

$$\varepsilon_{zz} = \frac{\partial u_z}{\partial z} \quad 2-33$$

Према теорији малих деформација, само сума комплементарних смичућих деформација (клизања) ε_{ij} и ε_{ji} узрокује смичуће напоне. Због тога, уместо ε_{xy} , ε_{xz} , ε_{yx} , ε_{yz} , ε_{zx} и ε_{zy} користе се γ_{xy} , γ_{yz} и γ_{zx} као деформације клизања:

$$\gamma_{xy} = \varepsilon_{xy} + \varepsilon_{yx} = \frac{\partial u_x}{\partial y} + \frac{\partial u_y}{\partial x} \quad 2-34$$

$$\gamma_{yz} = \varepsilon_{yz} + \varepsilon_{zy} = \frac{\partial u_y}{\partial z} + \frac{\partial u_z}{\partial y} \quad 2-35$$

$$\gamma_{zx} = \varepsilon_{zx} + \varepsilon_{xz} = \frac{\partial u_z}{\partial x} + \frac{\partial u_x}{\partial z} \quad 2-36$$

Из претходног следи да је за описивање стања деформација потребно познавати шест независних компоненти. Тензор деформација се онда може написати у векторском облику:

$$\boldsymbol{\varepsilon} = (\varepsilon_x \quad \varepsilon_y \quad \varepsilon_z \quad \gamma_{xy} \quad \gamma_{yz} \quad \gamma_{zx})^T \quad 2-37$$

У претходном изразу, за дилатације су усвојене ознаке са само једним индексом: ε_x , ε_y и ε_z уместо : ε_{xx} , ε_{yy} и ε_{zz} . Тензор деформација се може написати и у облику:

$$\boldsymbol{\varepsilon} = \begin{bmatrix} \varepsilon_x & \frac{1}{2}\gamma_{xy} & \frac{1}{2}\gamma_{xz} \\ \frac{1}{2}\gamma_{xy} & \varepsilon_y & \frac{1}{2}\gamma_{yz} \\ \frac{1}{2}\gamma_{xz} & \frac{1}{2}\gamma_{yz} & \varepsilon_z \end{bmatrix} \quad 2-38$$

Инваријанте тензора деформација су:

$$I_{\varepsilon 1} = \varepsilon_x + \varepsilon_y + \varepsilon_z \quad 2-39$$

$$I_{\varepsilon 2} = \varepsilon_x \varepsilon_y + \varepsilon_y \varepsilon_z + \varepsilon_z \varepsilon_x - \frac{1}{4} (\gamma_{xy}^2 + \gamma_{yz}^2 + \gamma_{zx}^2) \quad 2-40$$

$$I_{\varepsilon 3} = \det \boldsymbol{\varepsilon} = \varepsilon_x \varepsilon_y \varepsilon_z + \frac{1}{4} \gamma_{xy} \gamma_{yz} \gamma_{zx} - \frac{1}{4} (\varepsilon_x \gamma_{yz}^2 + \varepsilon_y \gamma_{zx}^2 + \varepsilon_z \gamma_{xy}^2) \quad 2-41$$

Као и код напона, главне деформације се могу одредити решавањем карактеристичне једначине:

$$\varepsilon_K^3 - I_{\varepsilon 1} \varepsilon_K^2 + I_{\varepsilon 2} \varepsilon_K - I_{\varepsilon 3} = 0 \quad 2-42$$

Решавањем једначине 2-42, добијају се главне деформације ε_1 , ε_2 и ε_3 и важи $\varepsilon_1 \geq \varepsilon_2 \geq \varepsilon_3$. Тензор деформација може да се растави на сферни $\boldsymbol{\varepsilon}^{(s)}$ и девијаторски део $\boldsymbol{\varepsilon}'$:

$$\boldsymbol{\varepsilon} = \boldsymbol{\varepsilon}^{(s)} + \boldsymbol{\varepsilon}' \quad 2-43$$

$$\boldsymbol{\varepsilon}^{(s)} = \begin{bmatrix} \frac{1}{3} I_{\varepsilon 1} & 0 & 0 \\ 0 & \frac{1}{3} I_{\varepsilon 1} & 0 \\ 0 & 0 & \frac{1}{3} I_{\varepsilon 1} \end{bmatrix} \quad 2-44$$

$$\boldsymbol{\varepsilon}' = \begin{bmatrix} \varepsilon_x - \frac{1}{3} I_{\varepsilon 1} & \frac{1}{2} \gamma_{xy} & \frac{1}{2} \gamma_{xz} \\ \frac{1}{2} \gamma_{xy} & \varepsilon_y - \frac{1}{3} I_{\varepsilon 1} & \frac{1}{2} \gamma_{yz} \\ \frac{1}{2} \gamma_{xz} & \frac{1}{2} \gamma_{yz} & \varepsilon_z - \frac{1}{3} I_{\varepsilon 1} \end{bmatrix} \quad 2-45$$

Девијаторски део тензора деформација се може написати у облику вектора:

$$\boldsymbol{\varepsilon}' = \left(\varepsilon_x - \frac{1}{3} I_{\varepsilon 1} \quad \varepsilon_y - \frac{1}{3} I_{\varepsilon 1} \quad \varepsilon_z - \frac{1}{3} I_{\varepsilon 1} \quad \frac{1}{2} \gamma_{xy} \quad \frac{1}{2} \gamma_{yz} \quad \frac{1}{2} \gamma_{xz} \right)^T \quad 2-46$$

$$= (\varepsilon'_{11} \quad \varepsilon'_{22} \quad \varepsilon'_{33} \quad \varepsilon'_{12} \quad \varepsilon'_{23} \quad \varepsilon'_{31})^T$$

Инваријанте девијаторског дела тензора деформација су:

$$I'_{\varepsilon 1} = \varepsilon'_{11} + \varepsilon'_{22} + \varepsilon'_{33} \quad 2-47$$

$$I'_{\varepsilon 2} = \varepsilon'_{11} \varepsilon'_{22} + \varepsilon'_{22} \varepsilon'_{33} + \varepsilon'_{33} \varepsilon'_{11} - \varepsilon'_{12}{}^2 - \varepsilon'_{23}{}^2 - \varepsilon'_{31}{}^2 \quad 2-48$$

$$I'_{\varepsilon 3} = \det \boldsymbol{\varepsilon}' = \varepsilon'_{11} \varepsilon'_{22} \varepsilon'_{33} + 2 \varepsilon'_{12} \varepsilon'_{23} \varepsilon'_{31} - \varepsilon'_{11} \varepsilon'_{23}{}^2 - \varepsilon'_{22} \varepsilon'_{31}{}^2 - \varepsilon'_{33} \varepsilon'_{12}{}^2 \quad 2-49$$

Прва инваријанта тензора деформација $I_{\varepsilon 1}$ (2-39) представља меру промене запремине елемента тла, односно запреминску деформацију (волуметријска деформација):

$$\varepsilon_v = \varepsilon_x + \varepsilon_y + \varepsilon_z = I_{\varepsilon 1} \quad 2-50$$

У механици тла се сматра да је запреминска деформација позитивна ако се запремина елемента тла смањује, а негативна ако се запремина елемента тла повећава.

Деформација промене облика (или дисторзијска деформација) се може одредити преко израза из друге инваријанте девијаторског дела тензора деформација:

$$\varepsilon_q = 2 \sqrt{\frac{1}{3} I'_{\varepsilon_2}} \quad 2-51$$

или

$$\varepsilon_q = \sqrt{\frac{2}{9} [(\varepsilon_x - \varepsilon_y)^2 + (\varepsilon_y - \varepsilon_z)^2 + (\varepsilon_z - \varepsilon_x)^2] + \frac{1}{3} (\gamma_{xy}^2 + \gamma_{yz}^2 + \gamma_{zx}^2)} \quad 2-52$$

Величина ε_q се може срачунати и преко следећег израза:

$$\varepsilon_q = \sqrt{\frac{2}{3} \left[\left(\varepsilon_x - \frac{\varepsilon_v}{3} \right)^2 + \left(\varepsilon_y - \frac{\varepsilon_v}{3} \right)^2 + \left(\varepsilon_z - \frac{\varepsilon_v}{3} \right)^2 + \frac{1}{2} (\gamma_{xy}^2 + \gamma_{yz}^2 + \gamma_{zx}^2) \right]} \quad 2-53$$

Поред претходно поменуте две деформацијске инваријанте, може се дефинисати и инваријанта смичуће деформације γ преко следећег израза:

$$\gamma = \frac{3}{2} \varepsilon_q \quad 2-54$$

У триаксијалним условима, смичућа деформација γ се може изразити као:

$$\gamma = \varepsilon_a - \varepsilon_l \quad 2-55$$

где је ε_a аксијална деформација, а ε_l бочна (латерална) деформација.

Кад је тло у питању, промена запреминске деформације $\Delta\varepsilon_v$ се може изразити преко промене коефицијента порозности e помоћу следећег израза:

$$\Delta\varepsilon_v = \frac{\Delta e}{1 + e} \quad 2-56$$

Уколико се уведе нова величина под називом специфична запремина ν , која се дефинише као запремина елемента тла који садржи јединицу запремине чврстих честица:

$$\nu = 1 + e \quad 2-57$$

Промена запреминске деформације се онда може представити преко израза:

$$\Delta\varepsilon_v = \frac{\Delta\nu}{\nu} \quad 2-58$$

2.4 Конститутивне релације за тло

Спољашње силе и утицаји у тлу изазивају унутрашње силе (напоне) услед којих се тло деформише, односно долази до промене запремине и облика (деформације) елемената тла. Претходно дати изрази које описују стање напона и деформација су општи и важе за било који материјал. Понашање два различита материјала оптерећена истим оптерећењима биће различито. Релације које описују понашање неког материјала услед додатног оптерећења, односно дају везу између величина које описују стање напона и величина које описују стање деформација, називају се конститутивне релације.

Везе између напона и деформација за тло су изразито нелинеарне и пластичне деформације се јављају брзо са порастом деформација. Имајући у виду трофазну структуру тла (чврсту, течну и гасовиту), тешко је формулисати конститутивни модел који може да опише све аспекте понашања тла, а да има релативно мали број улазних параметара (Potts & Zdravković, 1999). Тло се понаша као еластичан материјал само при веома малим деформацијама (мањим од 10^{-5}), док се са порастом деформација јављају и еластичне и пластичне деформације, односно понашање тла је еласто-пластично. У наставку овог поглавља дат је ближи приказ основних математичких релација потребних за описивање еласто-пластичног понашања тла.

2.4.1 Еластично понашање тла

Еластично понашање тла може бити линеарно еластично или нелинеарно еластично, изотропно или анизотропно. Најједноставнији конститутивни модел који описује изотропно линеарно еластично понашање је Нооке-ов закон. Веза између напона и деформација се може приказати у следећем облику:

$$\boldsymbol{\sigma}' = \mathbf{D}'^e \boldsymbol{\varepsilon} \quad 2-59$$

У претходном изразу, $\boldsymbol{\sigma}'$ и $\boldsymbol{\varepsilon}$ су дати изразима (2-27) и (2-37), а \mathbf{D}'^e представља еластичну конститутивну матрицу за ефективне напоне:

$$\mathbf{D}'^e = \begin{bmatrix} K' + \frac{4}{3}G & K' - \frac{2}{3}G & K' - \frac{2}{3}G & 0 & 0 & 0 \\ K' - \frac{2}{3}G & K' + \frac{4}{3}G & K' - \frac{2}{3}G & 0 & 0 & 0 \\ K' - \frac{2}{3}G & K' - \frac{2}{3}G & K' + \frac{4}{3}G & 0 & 0 & 0 \\ 0 & 0 & 0 & G & 0 & 0 \\ 0 & 0 & 0 & 0 & G & 0 \\ 0 & 0 & 0 & 0 & 0 & G \end{bmatrix} \quad 2-60$$

За формирање еластичне конститутивне матрице за ефективне напоне потребна су само два параметра: запремински модул за ефективне напоне K' и модул смицања G . Ова два параметра деформабилности се могу изразити преко Young-овог модула E' и Poisson-овог коефицијента μ' (оба за ефективне напоне) преко израза:

$$K' = \frac{E'}{3(1 - 2\mu')} \quad 2-61$$

$$G = \frac{E'}{2(1 + \mu')} \quad 2-62$$

Параметри K' и G су адекватнији за описивање понашања тла од параметара E' и μ' , зато што они представљају меру крутости тла на промену запремине и облика елемента тла. Као што је приказано у поглављу 2.2, сферни део тензора напона који се може представити средњим ефективним напонам p' узрокује само запреминске деформације ε_v , док девијаторски део тензора напона који се може представити девијаторским напонам q узрокује само промене облика које се описују девијаторском деформацијом ε_q . Тако да се израз (2-59) може написати и у следећем облику:

$$\begin{bmatrix} p' \\ q \end{bmatrix} = \begin{bmatrix} K' & 0 \\ 0 & 3G \end{bmatrix} \begin{bmatrix} \varepsilon_v \\ \varepsilon_q \end{bmatrix} \quad 2-63$$

Претходни израз не одговара реалном понашању тла, јер је могуће да се са променом смичућих деформација у тлу јаве и запреминске деформације. Веза између девијаторског дела тензора напона \mathbf{s} и девијаторског дела тензора деформација $\boldsymbol{\varepsilon}'$ је:

$$\mathbf{s} = 2G\boldsymbol{\varepsilon}' \quad 2-64$$

Релације (2-59)-(2-64) дају везу између ефективних напона и деформација. Да би се добиле релације које повезују тоталне напоне и деформације, довољно је заменити у поменутом изразима запремински модул за ефективне напоне K' са запреминским модулом за тоталне напоне K_u . Модул смицања за ефективне и тоталне напоне је исти, пошто порна вода не може да прими напоне смицања. У поглављу 2.7.4 детаљније је приказано како се могу срачунати тотални напони, односно запремински модул за тоталне напоне.

Иако једноставан за употребу, изотропни линеарно еластични конститутивни модел не може да адекватно опише реално понашање тла. Нелинеарно еластични конститутивни модели могу да узму у обзир зависност материјалних параметара од нивоа напона и/или деформација, али и даље нису довољно добри за описивање комплексног понашања тла. У литератури (Potts & Zdravković, 1999), (Lazović & Vukićević, 1997) и (Castellon & Ledesma, 2022a) дат је преглед еластичних конститутивних модела за тло.

2.4.2 Еласто-пластично понашање тла

Уколико су деформације мале, онда важи адитивна декомпозиција деформација. Тако да се укупне деформације могу представити као збир еластичних $\boldsymbol{\varepsilon}^e$ и пластичних деформација $\boldsymbol{\varepsilon}^p$:

$$\boldsymbol{\varepsilon} = \boldsymbol{\varepsilon}^e + \boldsymbol{\varepsilon}^p \quad 2-65$$

У еласто-пластичним конститутивним моделима важи да се главни правци достигнутих напона и главни правци инкременталних деформација поклапају. Пластичне деформације се јављају када се достигне одређен ниво напона. Пошто је напонско стање у тлу увек вишеосно, услов за појаву пластичних деформација се изражава као функција компоненталних напона. Та функција у простору главних

напона представља површ која се назива функција течења, површ течења или још површ попуштања и може се представити у следећем облику:

$$F(\boldsymbol{\sigma}, \mathbf{k}) = 0 \quad 2-66$$

У претходном изразу, \mathbf{k} представља вектор параметара стања који контролишу величину површи течења (параметри ојачања). Уколико је \mathbf{k} константно, онда је понашање тла идеално пластично, а ако се мења са пластичним деформацијама онда долази до ојачања или омекшања. Ако је напонско стање унутар површи течења ($F(\boldsymbol{\sigma}, \mathbf{k}) < 0$) јављају се само еластичне деформације. Ако је напонско стање на површи течења ($F(\boldsymbol{\sigma}, \mathbf{k}) = 0$), јављају се и пластичне деформације. Напонско стање изван површи течења ($F(\boldsymbol{\sigma}, \mathbf{k}) > 0$) није могуће, тако да површ течења у ствари представља област могућих напонских стања. Приликом пластичног течења мора бити задовољен израз (2-66) као и услов да напонска тачка увек остаје на површи течења (услов конзистенције пластичног течења):

$$dF = \frac{\partial F(\boldsymbol{\sigma}, \mathbf{k})}{\partial \sigma_i} d\sigma_i + \frac{\partial F(\boldsymbol{\sigma}, \mathbf{k})}{\partial k_i} dk_i = 0 \quad 2-67$$

Величина и правац пластичних деформација је одређен законом течења:

$$d\varepsilon_i^p = \Lambda \frac{\partial P(\boldsymbol{\sigma}, \mathbf{m})}{\partial \sigma_i} \quad 2-68$$

У претходном изразу, $d\varepsilon_i^p$ представља компоненте прираштаја пластичних деформација, $P(\boldsymbol{\sigma}, \mathbf{m})$ је функција пластичног потенцијала а Λ скаларни мултипликатор. Вектор прираштаја пластичних деформација је увек управан на површ пластичног потенцијала, док скаларни мултипликатор дефинише величину пластичних деформација. Функција пластичног потенцијала се може приказати у следећем облику:

$$P(\boldsymbol{\sigma}, \mathbf{m}) = 0 \quad 2-69$$

У претходном изразу \mathbf{m} представља вектор параметара стања. Уколико се усвоји да се површ течења и функција пластичног потенцијала поклапају ($F(\boldsymbol{\sigma}, \mathbf{k}) = P(\boldsymbol{\sigma}, \mathbf{m})$), вектор прираштаја пластичних деформација је управан и на површ течења и такав закон течења се назива асоцијативним законом течења. У супротном, закон течења је неасоцијативан.

Избор асоцијативног или неасоцијативног закона течења у конститутивном моделу знатно утиче на предвиђање понашања тла, као и на ресурсе потребне у нумеричкој анализи (као што је метода коначних елемената) применом рачунара. Закон течења веома утиче на описивање ефекта дилатанције. Дилатанција представља особину тла да мења запремину током смицања и дефинише се као однос прираштаја пластичних запреминских и прираштаја пластичних смичућих деформација.

Промена параметара стања \mathbf{k} , који описују величину површи течења, са променом пластичних деформација је дефинисана законом ојачања, односно законом омекшања. Приликом ојачања тла, са порастом пластичних деформација расте и смичућа чврстоћа тла. Закон ојачања омогућава да се одреди вредност скаларног мултипликатора у изразу (2-68). Ојачање материјала је изотропно ако се површ течења шири у свим

правцима уз задржавање почетног облика. Уколико се површ течења транслира или ротира уз задржавање првобитног облика, онда се такво ојачање назива кинематичким ојачањем.

У еласто-пластичним конститутивним моделима, све релације су уобичајено дате у инкременталном облику, односно прираштаји напона су повезани са прираштајима деформација. Укупни напони након једног инкремента се одређују као збир напона на почетку инкремента и прираштаја напона на том инкременту. Веза између прираштаја напона $\Delta\sigma$ и прираштаја деформација $\Delta\varepsilon$ може се написати у следећем облику:

$$\Delta\sigma = \mathbf{D}^{ep} \Delta\varepsilon \quad 2-70$$

где је \mathbf{D}^{ep} еласто-пластична конститутивна матрица. Код конститутивних модела који користе асоцијативан закон течења, \mathbf{D}^{ep} је симетрична, док код модела који користе неасоцијативан закон течења \mathbf{D}^{ep} је несиметрична. Еласто-пластична матрица се може одредити из следећег израза:

$$\mathbf{D}^{ep} = \mathbf{D}^e - \frac{\mathbf{D}_e \mathbf{b} \mathbf{a}^T \mathbf{D}_e}{\mathbf{a}^T \mathbf{D}^e \mathbf{b} + A} \quad 2-71$$

$$\mathbf{a} = \frac{\partial F(\sigma, \mathbf{k})}{\partial \sigma_i} \quad 2-72$$

$$\mathbf{b} = \frac{\partial P(\sigma, \mathbf{m})}{\partial \sigma_i} \quad 2-73$$

$$A = - \frac{1}{\Lambda} \frac{\partial F(\sigma, \mathbf{k})}{\partial k_i} \frac{\partial \mathbf{k}}{\partial \varepsilon_p} \Delta \varepsilon_p \quad 2-74$$

Величина A у претходном изразу је константна уколико постоји линеарна веза између параметара стања \mathbf{k} и пластичних деформација ε_p .

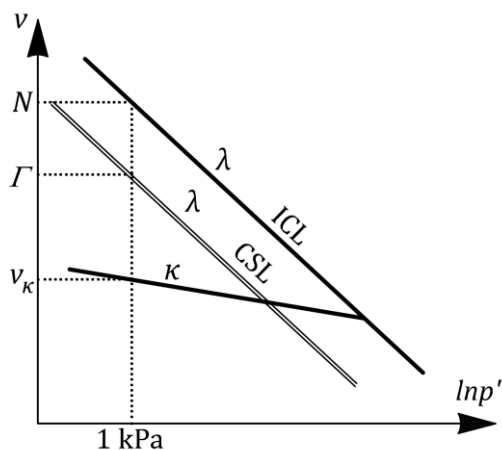
2.4.3 Концепт критичног стања тла

Критично стање тла представља модел еласто-пластичног понашања тла и представља стање идеалне пластичности. Уколико је тло довољно дуго изложено смицању, доћи ће у стање у коме се пластичне деформације смицања дешавају без промене ефективних напона и без промене запремине (критично стање). У критичном стању важи:

$$\frac{\partial p'}{\partial \varepsilon_q} = \frac{\partial q}{\partial \varepsilon_q} = \frac{\partial v}{\partial \varepsilon_q} = 0 \quad 2-75$$

Концепт критичног стања тла су развили Roscoe и сарадници (Roscoe, et al., 1958), (Schofield & Wroth, 1968) на Универзитету у Cambridge-у средином прошлог века. Иако је овај модел понашања тла развијен да опише понашања нормално консолидованих и благо преконсолидованих глина, широко је коришћен за формулацију великог броја конститутивних модела свих типова и понашања тла. Стање тла се приказује у тродимензионалном $p' - q - v$ простору.

Понашање у опиту изотропне компресије узорка нормално консолидоване глине приказан је на слици испод, на полулогаритамском дијаграму $\ln p' - v$. Извршена је идеализација тако да се промена запремине при изотропној компресији и приликом растерећења и поновног оптерећења приказује као права линија. Такође, занемарена је и хистерезисна петља у циклусу растерећења и поновног оптерећења.



Слика 2-3 Понашање тла при изотропној компресији – преузето из (Јосковић, 2017)

Линија изотропне компресије (Isotropic compression line - ICL) одговара нормално консолидованом тлу, док линије које описују понашање тла при растерећењу и поновном оптерећењу се називају линије бубрења и рекомпресије. Стање тла на линији бубрења (рекомпресије) одговара преконсолидованом тлу. Нагиб линије изотропне компресије дефинисан је параметром λ , а нагиб линије бубрења (рекомпресије) параметром κ . Једначина изотропне компресије је:

$$v = N - \lambda \ln p' \quad 2-76$$

Једначина линије бубрења је:

$$v = v_k - \kappa \ln p' \quad 2-77$$

Параметар N представља специфичну запремину нормално консолидованог тла при $p' = 1 \text{ kPa}$, а параметар v_k представља специфичну запремину преконсолидованог тла при $p' = 1 \text{ kPa}$.

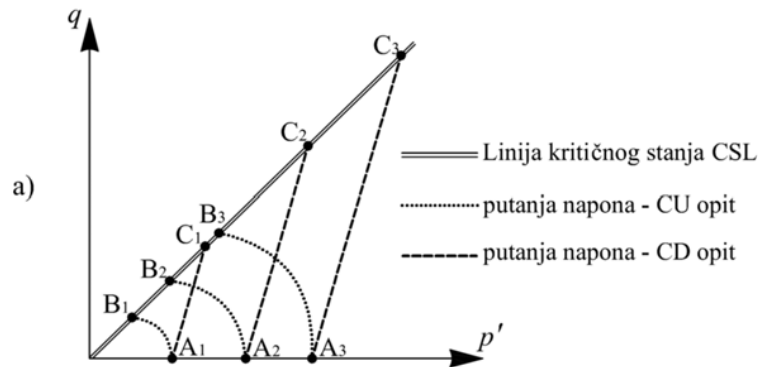
Путање напона у недренираном CU и дренираном CD опиту триаксијалне компресије на узорцима тла оптерећеним до лома, завршавају на једној линији за оба опита. Та линија представља линију критичног стања (Critical state line - CSL). При достизању критичног стања напонски однос износи:

$$\frac{q_{cs}}{p'_{cs}} = \eta_{cs} = M \quad 2-78$$

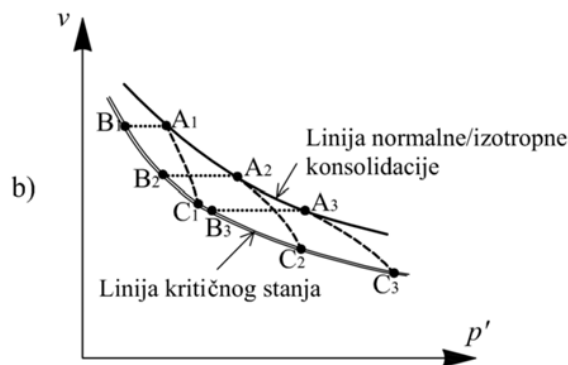
Параметар M представља нагиб линије критичног стања у $p' - q$ дијаграму (Слика 2-4). На (Слика 2-5) приказана је линија критичног стања у $p' - v$ равни. На (Слика 2-3) је приказана линија критичног стања у $\ln p' - v$ равни и њена једначина гласи:

$$v = \Gamma - \lambda \ln p' \quad 2-79$$

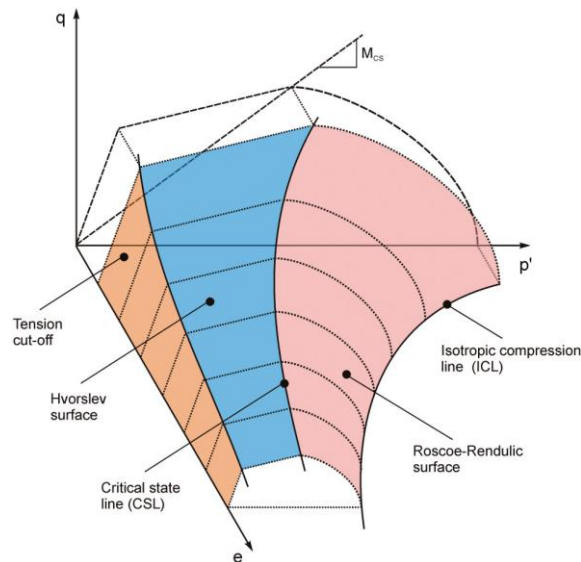
Параметар Γ представља специфичну запремину тла при $p' = 1 \text{ kPa}$ на линији критичног стања.



Слика 2-4 Путање напона у CU и CD опиту – преузето из (Vukićević & Jocković, 2019)



Слика 2-5 Линија критичног стања у $p' - v$ равни - преузето (Vukićević & Jocković, 2019)

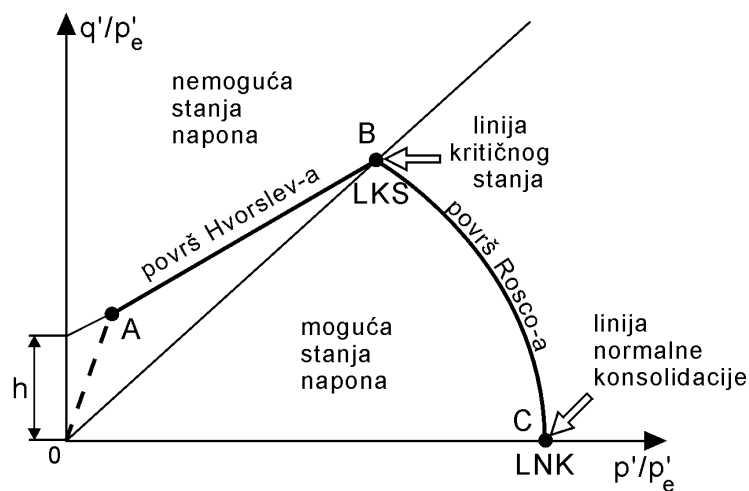


Слика 2-6 Гранична површ у $p' - q - e$ простору – преузето из (Schadlich & Schweiger, 2014)

У простору $p' - q - v$ постоји површ која одваја простор могућих стања напона од оних који нису могући. Таква површ се назива гранична површ. За напонска стања унутар граничне површи, деформације су еластичне, а пластичне деформације се јављају када путања напона дође до граничне површи. Гранична површ се састоји од неколико

површи (Слика 2-6). Површ која се налази између линије изотропне компресије и линије критичног стања се назива површ Roscoe-а. Напонске путање у дренираним и недренираним триаксијалним опитима за нормално консолидоване узорке тла се крећу по овој површи до лома.

За преконсолидоване узорке тла, у условима триаксијалне компресије, запремина се прво смањује, а затим се повећава, да би на крају достигла одређену стационарну вредност. У том случају, путања напона прелази линију критичног стања, а затим се поново враћа на њу. Због тога, гранична површ за преконсолидована тла се налази изнад линије критичног стања, и та гранична површ се назива површ Hvorslev-а. Уколико би се површ Hvorslev-а продужила до равни $q = e$, постојала би могућност да се јаве напони затезања. Да би се то избегло, ова површ је локално модификована правом OA (Слика 2-7). Пошто се пластичне деформације јављају када путања напона додирне граничну површ, онда гранична површ представља и површ течења.



Слика 2-7 Комплетна гранична површ у нормализованој равни $p' - q$ – преузето из (Maksimović, 2008)

2.5 Понашање преконсолидованих глина

Глине су нормално консолидоване ако у прошлости нису биле оптерећене већим вертикалним ефективним напонам од текуће вредности вертикалног ефективног напона. Глине су преконсолидоване ако су у прошлости биле оптерећене већим вертикалним ефективним напонам (напонам преконсолидације) него што је текући вертикални ефективни напон. Степен преконсолидације (OCR – overconsolidation ratio) се дефинише као однос напона преконсолидације σ'_c и текућег вертикалног ефективног напона σ'_{vi} :

$$OCR = \frac{\sigma'_c}{\sigma'_{vi}} \quad 2-80$$

За нормално консолидовано тло важи да је $\sigma'_c = \sigma'_{vi}$, односно $OCR = 1$, а за преконсолидовано тло $\sigma'_c > \sigma'_{vi}$, односно $OCR > 1$. Преконсолидација тла може да се изрази и преко разлике напона преконсолидације и текуће вредности вертикалног ефективног напона (POP – pre-overburden pressure):

$$POP = \sigma'_c - \sigma'_{vi}$$

2-81

Може се још дефинисати и изотропни степен преконсолидације R као однос максималног средњег ефективног напона p'_{max} који је деловао у тлу и текуће вредности средњег ефективног напона p'_i :

$$R = \frac{p'_{max}}{p'_i}$$

2-82

Само у случају да су глине нормално консолидоване важи да је $OCR = R = 1$. За преконсолидовање глине, ова два степена имају различите бројне вредности.

Преконсолидација глина може бити последица напрезања у тлу која се јављају услед исушивања површинских слојева тла од стране корења вегетације, површинске евапорације или промена нивоа подземне воде. Ерозија површинских слојева тла и топљење ледника има за последицу растерећење доњих слојева тла, односно доњи слојеви постају преконсолидовани. Секундарна консолидација (пузање) и старење глина такође могу узроковати преконсолидацију тла. Преконсолидација тла се може десити и услед извођења грађевинских радова као што су изградња потпорних конструкција и тунела.

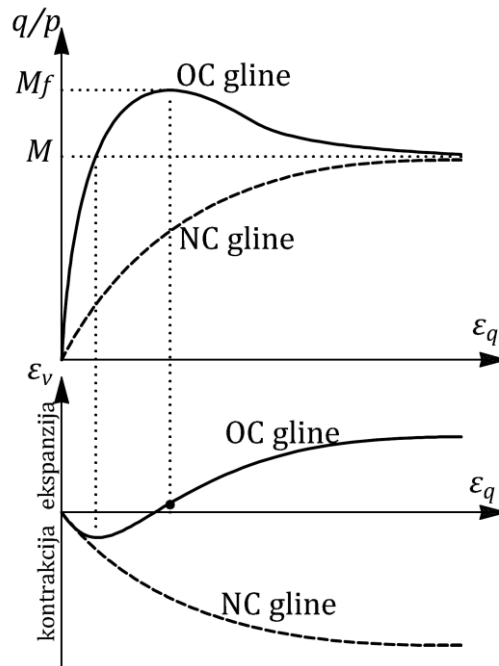
Преконсолидоване и нормално консолидоване глине имају различито напонско-деформацијско понашање. Неке од тих разлика, као што су понашање у едометарском и триаксијалном опиту, су описане у оквиру претходног поглавља. Преконсолидоване глине имају мањи коефицијент порозности, већу смичућу чврстоћу и мање су деформабилне од нормално консолидованих глина. Преконсолидоване глине имају посебно изражен ефекат дилатанције. Понашање преконсолидованих глина зависи од претходне историје оптерећења, али само до одређене границе, након које се глине понашају као нормално консолидоване (Јосковић, 2017).

Карактеристично понашање преконсолидоване глине у дренираним условима приказано је на (Слика 2-8). Смичући напон се повећава до вршне смичуће чврстоће, а затим опада до константне вредности (омекшање), што одговара крто-пластичном лому. Запремина се прво смањује, а онда се повећава пре лома.

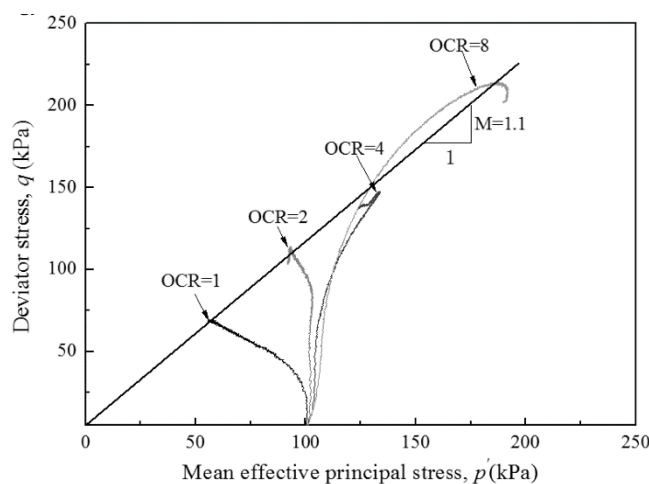
У недренираним условима, долази до генерисања додатних порних притисака услед спречене промене запремине. Позитивни порни притисци који се повећавају до лома јављају се код нормално консолидованих глина. Порни притисци код преконсолидованих глина прво расту, а затим опадају и могу чак бити негативни. Смањење порних притисака је веће што је већи степен преконсолидације. Код нормално консолидованих глина путања напона долази до граничне површи испод линије критичног стања, и пошто ће генерисани порни притисци бити позитивни, узорак глине ће отпуштати воду. Због тога се тај део граничне површи назива "мокра" страна (wet side). Код преконсолидованих глина, путања напона долази до граничне површи изнад линије критичног стања, и генерисани порни притисци могу бити негативни, односно узорак упија воду. Тај део граничне површи се назива "сува" страна (dry side).

Путање ефективних напона у недренираним условима за глине са различитим степеном преконсолидације дате су на (Слика 2-9). За нормално консолидоване глине,

путање ефективних напона се завршавају на линији критичног стања, док за преконсолидоване глине додирују површ Hvorslev-a, а затим се враћају до линије критичног стања.



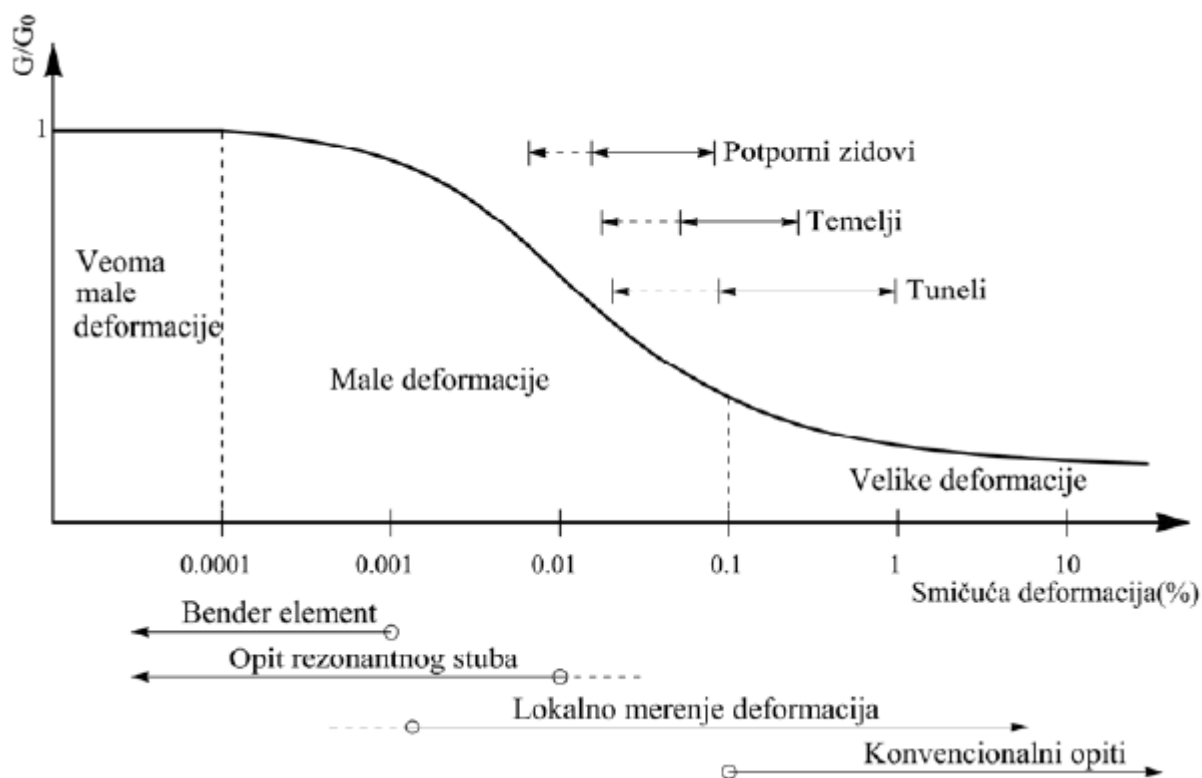
Слика 2-8 Карактеристично понашање преконсолидоване глине у дренажним условима – преузето из (Vukićević & Jocković, 2019)



Слика 2-9 Путање ефективних напона у недренажном триаксијалном опиту – преузето из (Gu, et al., 2016)

Смичућа чврстоћа испитана на узорцима глине у опиту триаксијалне компресије или у опиту триаксијалне екстензије неће бити иста, због утицаја средњег главног напона. Утицај средњег главног напона на чврстоћу глина је веома изражен у проблемима раванског стања деформација. Смичућа чврстоћа тла у опиту триаксијалне компресије и у опиту који је изведен у условима раванског стања деформација се разликује и та разлика расте што је тло збијеније (Vukićević & Jocković, 2019). Због тога, потребно је користити критеријум лома који на адекватан начин узима у обзир утицај сва три главна напона на чврстоћу тла и који предвиђа различите чврстоће тла у компресији, екстензији или раванском стању деформација.

Још један од аспеката понашања тла које треба узети у обзир приликом формулисања конститутивног модела је величина крутости тла при малим деформацијама (small strain stiffness) и како крутост тла опада са порастом смичућих деформација. Крутост тла при веома малим деформацијама ($\gamma < 10^{-4} \%$) је чак и до десет пута већа од крутости тла при великим деформацијама ($\gamma > 10^{-1} \%$) које се бележе у конвенционалним лабораторијским опитима. Понашање тла у подручју веома малих деформација је линеарно еластично и модул смицања има тада максималну (или иницијалну) вредност G_0 . Са порастом смичућих деформација, модул смицања опада, све до подручја великих деформација где има релативно малу вредност G (Слика 2-10).



Слика 2-10 Промена модула смицања са порастом смичућих деформација (S – крива) – преузето из (Vukićević & Jocković, 2019)

На величину иницијалне вредности модула смицања G_0 и облик криве редукције модула смицања (S – крива) утичу фактори као што су величина смичућих деформација, величина средњег ефективног напона, коефицијент порозности, индекс пластичности, степен преконсолидације, дијагенеза тла, претходна историја оптерећивања, брзина наношења оптерећења и анизотропија. Због тога, да би се формирала S -крива и сви набројани фактори узели у обзир, потребна су адекватна и детаљна лабораторијска и теренска испитивања.

За потребе рационалног пројектовања геотехничких конструкција и адекватно предвиђање померања у тлу, конститутивни модел треба да на одговарајући начин предвиђа величину крутости тла у подручју веома малих и малих деформација. Такав конститутивни модел би онда могао да се користи и за предвиђање понашања тла при цикличним оптерећењем. Одговарајући конститутивни модел за преконсолидоване глине који би се користио за решавање различитих проблема у геотехници, би био онај модел који може да обухвати претходно описано напонско-деформацијско понашање.

2.6 Преглед конститутивних модела за глине

Једноставни конститутивни модели као што су Tresca-ин и von Mises-ов идеално пластични модели могу да се користе за описивање понашања глина у недренираним условима у зависности од тоталних напона. Понашање у дренираним условима у зависности од ефективних напона се може описати Mohr-Coulomb-овим и Drucker-Prager-овим идеално пластичним конститутивним моделима. Проширени Mohr-Coulomb-ов модел који може да узме у обзир ојачање или омекшање са порастом пластичних деформација може се наћи у (Potts & Zdravković, 1999). Ови конститутивни модели су једноставни и имају мали број параметара модела који се могу релативно једноставно одредити. Упркос томе што су једноставни за коришћење, не могу адекватно да опишу напонско-деформацијско понашање глина.

Велики број конститутивних модела за глине је формулисан у оквиру концепта критичног стања. Први модел критичног стања је био Cam Clay модел (Roscoe & Schofield, 1963) и (Schofield & Wroth, 1968). Модификацију Cam Clay модела под називом Modified Cam Clay (MCC) модел представили су (Roscoe & Burland, 1968). Иако MCC модел може на задовољавајући начин да опише понашање нормално консолидованих и благо преконсолидованих глина при монотонном оптерећењу, не даје адекватна предвиђања понашања преконсолидованих глина. MCC модел представља основу на којој су даље развијани напреднији конститутивни модели који превазилазе његове недостатке. Прикази ових модела могу се наћи у (Potts & Zdravković, 1999), (Vukićević & Jocković, 2019) и (Lade, 2005).

За мека тла, као што су благо преконсолидоване глине, глиновите прашине и тресет, развијен је Soft Soil модел као модификација MCC модела (Stolle, et al., 1997). Soft Soil Creep модел представља унапређен Soft Soil модел који може да опише пузање меких глина склоних секундарној компресији (Vermeer & Neher, 1999). Оба модела су изотропна и комбинују Mohr-Coulomb-ов услов лома са површи течења MCC модела. За меке глине су још развијени модели S-CLAY1 (Wheeler, et al., 2003), S-CLAY1S (Koskinen, et al., 2002), Creep-SCLAY1S (Sivasithamparam, et al., 2015) као надоградња MCC модела. Ови модели узимају у обзир анизотропију услед пластичних деформација (S-CLAY1), анизотропију комбиновану са деструктурирањем глина (S-CLAY1S) и анизотропију, деструктурирању и пузање (Creep-SCLAY1S).

MELANIE (Mouratidis & Magnan, 1983) модел је модификација MCC модела тако да узима у обзир анизотропију и користи неасоцијативан закон течења. Sekiguchi-Ohta модел, (Sekiguchi & Ohta, 1977) и (Iizuka & Ohta, 1987) је развијен да опише понашање нормално консолидованих глина из MCC модела и узима у обзир реолошке ефекте и напонски индуковану анизотропију. Овај модел користи SMP (Spatially Mobilized Plane) критеријум лома и асоцијативан закон течења. (Yu, 1998) је развио CASM модел у оквиру критичног стања за глине и пескове који користи концепт параметра стања. Овај модел представља модификацију MCC модела и може на задовољавајући начин да предвиди понашање преконсолидованих глина.

MIT-E3 (Massachusetts Institute of Technology) конститутивни модел тла је развијен у оквиру концепта граничне површи и развијен је из MIT-E1 модела (Kavvasas, 1982). MIT-E1 модел је развијен за нормално консолидоване глине из MCC модела и превазилази неке његове недостатке. У односу на MCC модел, MIT-E1 користи

анизотропну површ течења, кинематичко ојачање и може да симулира омекшање у недренираним условима. MIT-E3 (Whittle, 1987) може да симулира нелинеарност при малим деформацијама као и хистерезисни одговор при циклусима оптерећења и растерећења. MIT-E3 може да опише понашање преконсолидованих глина.

Конститутивни модели који користе концепт балона ("Bubble"), односно користе једну или више мањих кинематичких површи течења унутар граничне површи, као што су модели (Al-Tabba & Wood, 1989), (Wood, 1995) и 3-SKH модел (Stallebrass, 1990) могу да опишу понашање глина при цикличном оптерећењу. Ови модели, унутар граничне површи користе површ течења која ограничава еластично понашање и површ историје оптерећења унутар које тло „памти“ претходну историју оптерећења.

Такође, Hardening Soil (HS) (Schanz, et al., 1999) и Hardening Soil Small (HSS) (Benz, 2007), као модели који користе двоструко ојачање и који су веома заступљени у комерцијалним програмима за нумеричку анализу, могу се користити за анализу понашања преконсолидованих глина.

У оквиру концепта критичног стања као надоградња MCC модела, користећи концепт граничне површи и комбинованог ојачања, развијен је HASP модел (Jocković, 2017) и (Jocković & Vukićević, 2018a) за преконсолидоване глинe. Овај модел биће детаљно описан у Поглавље 3.

Brick модели (Simpson, 1992), (Lansivaara & Nordal, 1998), (Lehane & Simpson, 2000), (Clarke & Hird, 2012), (Ellison, et al., 2012) и (Jovičić, et al., 2023) су се показали да могу успешно да се користе за описивање понашања преконсолидованих глина. Brick модели су дефинисани у деформацијском простору и теоријски оквир како ови модели функционишу дат је у (Simpson, 1992). Понашање тла се може описити користећи аналогију особе који вуче цигле ("brick") помоћу неистегљивих канапа док се креће. Простор у коме се особа креће је простор деформација. Положај особа представља стање укупних деформација у тлу а кретање цигли пластичне деформације. Уколико се цигла не креће, односно канап који повезује особу и циглу није затегнут понашање је еластично. Уколико је канап затегнут, цигла се помера, генеришу се пластичне деформације и крутост тла се редукује. Модел треба да користи одређен број цигли и одговарајуће дужине канапа како би адекватно описао деградацију крутости тла, поготово у домену веома малих деформација. Brick модели могу да се комбинују са другим моделима како би се добили напреднији модели као што је Hardening Soil-Brick Model (Cudny & Truty, 2020).

Хипопластични конститутивни модели описују понашање тла ван концепта еласто-пластичних конститутивних модела. Односно, ови модели не користе површ течења, пластични потенцијал, законе ојачања и друго. Хипопластични конститутивни модели користе развојне инкременталне једначине где су прираштаји напона тензорска функција прираштаја напона. Хипопластични модели за глинe се могу наћи у радовима (Mašin, 2005), (Huang, et al., 2006a), (Weifner & Kolymbas, 2007), а за преконсолидоване глинe у раду (Wang & Wu, 2021).

Са порастом комплексности претходно поменутих конститутивних модела расте и број параметара модела, од којих многи не могу да се одреде из конвенционалних лабораторијских опита. Такође, формулација неких напредних модела може бити таква да

њихова имплементација у рачунарским програмима буде веома комплексна. Због тога је примена таквих модела за решавање практичних проблема ограничена.

2.7 Нумеричке методе за решавање геотехничких проблема

2.7.1 Опште

Напредни конститутивни модели за тло који имају могућност да на адекватан начин опишу напонско-деформацијско понашање тла, по правилу имају компликовану математичку формулацију. Примена таквих конститутивних модела за анализу граничних проблема захтева да модел буде имплементиран у неки рачунарски програм. Имплементација конститутивног модела обухвата формулацију нумеричке процедуре за интеграцију напона за задати инкремент деформација. Поступак имплементације треба да буде ефикасан, тачан и робустан. Ефикасност поступка имплементације подразумева да је потребан што мањи број рачунских операција и итерација, односно што мањи утросак рачунарског времена. Од тачности поступка имплементације зависи и тачност решења граничног проблема који се анализира. Поступак имплементације је робустан ако се може применити у општем случају у односу на услове оптерећивања и величину инкремента оптерећења.

Рачунарски програм у који је имплементиран један конститутивни модел, користи неку од нумеричких метода као што су метод коначних елемената (finite element method), метод коначних разлика (finite difference method), метод граничних елемената (boundary element method), метод дискретних елемената (discrete element method), метод спектралних елемената (spectral element method), метод виртуелних елемената (virtual element method) и друге за решавање диференцијалних једначина које описују физички феномен. Гранични проблем (boundary value problem) представља систем диференцијалних једначина са одговарајућим граничним условима. Рачунарски програми за нумеричку анализу геотехничких проблема највише користе метод коначних елемената (МКЕ).

2.7.2 Метод коначних елемената

МКЕ подразумева физичку дискретизацију разматраног континуума (систем са бесконачно много степени слободе) у систем са коначним бројем степени слободе. Континуум се дели на коначан број елемената са коначним димензијама (коначни елементи) који су међусобно повезани у чворовима и чине мрежу коначних елемената. Основне непознате величине у чворовима мреже коначних елемената могу бити померања, напони или њихова комбинација. Најчешће се као непознате величине у чворовима коначних елемената узимају померања. Веза између непознатних померања у чворовима и померања у било којој тачки коначног елемента дата је интерполационим функцијама. Избор интерполационих функција представља једну од основних апроксимација у овој методи. Предност МКЕ је у томе што је веома прилагодљива нумеричка метода због чега је најзаступљенија у рачунарским програмима. Може се користити за решавање разних физичких проблема без ограничења у погледу геометрија, оптерећења, граничних услова и материјала. На тачност решења се

може лако утицати повећањем броја и комбиновањем различитих типова коначних елемената.

Основна једначина у МКЕ на нивоу једног елемента се добија користећи одговарајући варијациони принцип и гласи:

$$\mathbf{K}_E \Delta \mathbf{u}_E = \Delta \mathbf{f}_E \quad 2-83$$

Где је \mathbf{K}_E матрица крутости елемента, $\Delta \mathbf{u}_E$ је вектор прираштаја померања у чворовима елемента, а $\Delta \mathbf{f}_E$ је вектор прираштаја сила у чворовима елемента. Комбиновањем (assembling) једначина за све елементе, добија се једначина за цео систем коначних елемената:

$$\mathbf{K} \Delta \mathbf{u} = \Delta \mathbf{f} \quad 2-84$$

У претходном изразу \mathbf{K} је глобална матрица крутости, $\Delta \mathbf{u}$ је вектор прираштаја померања у свим чворовима, а $\Delta \mathbf{f}$ је вектор прираштаја сила у свим чворовима мреже коначних елемената. Уз одговарајуће граничне услове, односно познате силе и померања у чворовима мреже, једначина (2-84) се може решити тако да се одреде непозната померања у чворовима. Из познатих померања у чворовима, даље се могу срачунати све непознате величине у коначном елементу као што су напони и деформације.

Матрица крутости коначног елемента, \mathbf{K}_E , се може одредити из следећег изрази:

$$\mathbf{K}_E = \int_{Vol} \mathbf{B}^T \mathbf{D} \mathbf{B} dVol \quad 2-85$$

У претходном изразу \mathbf{D} је конститутивна матрица, \mathbf{B} је матрица која садржи изводе интерполационих функција, а интеграција се врши по читавој запремини коначног елемента. У случају да се користе изопараметарски елементи, израз за матрицу крутости једног елемента је:

$$\mathbf{K}_E = \int_{-1}^1 \int_{-1}^1 t \mathbf{B}^T \mathbf{D} \mathbf{B} |J| dS dT \quad 2-86$$

Где су S и T природне координате коначног елемента, t је дебљина коначног елемента у раванским проблемима, а $|J|$ је детерминанта Јакобијан матрице која представља везу између природних координата једног елемента и глобалних координата. Вектор прираштаја чворних сила $\Delta \mathbf{r}_E$ се може одредити из следећег изрази:

$$\Delta \mathbf{f}_E = \int_{Vol} \mathbf{N}^T \Delta \mathbf{p} dVol + \int_{Srf} \mathbf{N}^T \Delta \mathbf{t} dSrf \quad 2-87$$

У претходном изразу \mathbf{N} је матрица која садржи интерполационе функције, $\Delta \mathbf{p}$ су запреминске силе које делују на коначни елемент, $\Delta \mathbf{t}$ су површинске силе које делују по контурама коначног елемента, а интеграција се врши по запремини и делу контура коначног елемента где делују површинске силе. Више о томе како се изводе једначине (2-83) па до (2-87) може се наћи у (Potts & Zdravković, 1999).

Интеграл дати изразима (2-85), (2-86) и (2-87) се могу директно срачунати само за коначне елементе који имају једноставну геометрију. Због тога, матрица крутости једног елемента и вектор чворних сила се рачунају методама нумеричке интеграције. Користећи неко од правила за нумеричку интеграцију, интеграл се замењује пондерисаном сумом вредности подинтегралне функције у унапред изабраним тачкама (integration points). У МКЕ се најчешће користи Gauss-ов поступак интеграције и тачке које се користе за интеграцију се називају Gauss-ове тачке. Број тачака које се користе за интеграцију зависи од типа и облика коначног елемента и утиче на тачност нумеричког поступка. Пошто се матрица крутости елемента и вектор чворних сила рачунају у овим тачкама, онда се напони и деформације такође рачунају у овим тачкама (stress points). Алгоритмима за нумеричку интеграцију конститутивних релација, који се користе за имплементацију конститутивног модела у рачунарски програм заснован на МКЕ, рачунају се напони у тачкама нумеричке интеграције. Ти напони се даље користе за добијање матрице крутости елемента и проверу да ли су унутрашње силе у равнотежи са спољашњим силама у нелинеарној анализи.

Пошто је напонско-деформацијско понашање тла веома нелинеарно (материјална нелинеарност), основне једначине МКЕ (2-83) и (2-84) су дате у инкременталној форми. Конститутивна матрица \mathbf{D} у изразима за одређивање матрице крутости елемента није константна, већ се мења у функцији напона и деформација. Односно, за одређивање матрице крутости елемента, треба да се користи еласто-пластична конститутивна матрица \mathbf{D}^{ep} .

Материјално нелинеарни проблеми се у МКЕ решавају применом неког инкрементално-итеративног поступка. Оптерећење се наноси у инкрементима, и у оквиру сваког инкремента се примењује неки од итеративних поступака, тако да на крају сваког инкремента буде задовољен услов равнотеже спољашњих и унутрашњих сила:

$${}^{t+\Delta t}\mathbf{q} = {}^{t+\Delta t}\mathbf{f} \quad 2-88$$

Где ${}^{t+\Delta t}\mathbf{q}$ и ${}^{t+\Delta t}\mathbf{f}$ представљају векторе унутрашњих, односно спољашњих чворних сила. Са Δt је означен временски интервал, односно инкремент оптерећења. У статичкој нелинеарној анализи, временски интервал показује у ствари различите конфигурације тела током деловања оптерећења. Време t одговара почетку инкремента, а време $t + \Delta t$ одговара крају инкремента. Вектор унутрашњих сила се може одредити из израза:

$${}^{t+\Delta t}\mathbf{q} = \int_{Vol} \mathbf{B}^T {}^{t+\Delta t}\boldsymbol{\sigma} dVol \quad 2-89$$

У претходном изразу ${}^{t+\Delta t}\boldsymbol{\sigma}$ представља вектор напона на крају инкремента. Једначина (2-84) се за један инкремент оптерећења може написати у облику:

$$\mathbf{K}_t^i \Delta \mathbf{u}^i = \Delta \mathbf{f}^i \quad 2-90$$

Са \mathbf{K}_t^i је означена тангентна матрица крутости, $\Delta \mathbf{u}^i$ представља прираштај померања, $\Delta \mathbf{f}^i$ нанети инкремент спољашњег оптерећења. Индекс i означава број итерације. Решавањем претходне једначине, добија се прираштај померања $\Delta \mathbf{u}^i$. Прираштај деформација у елементу се онда може срачунати као:

$$\Delta \boldsymbol{\varepsilon}^i = \mathbf{B} \Delta \mathbf{u}^i \quad 2-91$$

Прираштај напона се из конститутивних релација може добити преко:

$$\Delta \boldsymbol{\sigma}^i = \int_0^{\Delta \boldsymbol{\varepsilon}^i} \mathbf{D}^{ep} d\boldsymbol{\varepsilon} \quad 2-92$$

Вектор напона на се онда може срачунати из:

$$\boldsymbol{\sigma}^i = {}^t \boldsymbol{\sigma} + \Delta \boldsymbol{\sigma}^i \quad 2-93$$

Са ${}^t \boldsymbol{\sigma}$ је означен вектор напона на почетку инкремента

Вектор унутрашњих сила је онда:

$$\mathbf{q}^i = \int_{Vol} \mathbf{B}^T \boldsymbol{\sigma}^i dVol \quad 2-94$$

Ако је \mathbf{q}^i у равнотежи са вектором спољашњих сила (израз 2-89), односно разлика између унутрашњих и спољашњих сила (residual) мања од неке унапред дефинисане величине:

$$\mathbf{r}^i = \mathbf{q}^i - {}^{t+\Delta t} \mathbf{f} < tol \quad 2-95$$

Онда је вектор напона на крају инкремента:

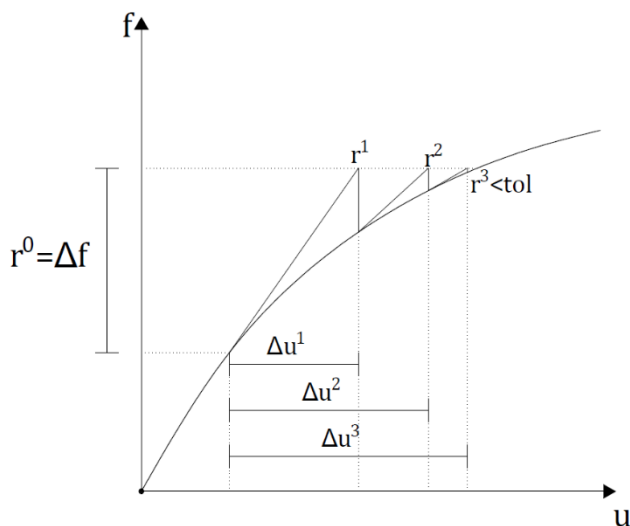
$${}^{t+\Delta t} \boldsymbol{\sigma} = \boldsymbol{\sigma}^i \quad 2-96$$

Ако није задовољен услов (2-95), остатак \mathbf{r}^i се наноси као нови инкремент и поступак се понавља (нова итерације) све док није задовољен тај услов (**Слика 2-11**). Инкрементална померања се рачунају као збир прираштаја померања из свих итерација у оквиру једног инкремента.

Уколико је тангентна матрица крутости \mathbf{K}_t^i иста у свим итерацијама у оквиру једног инкремента, онда се такав поступак назива модификовани Newton-Raphson-ов поступак (MNR). У MNR \mathbf{K}_t^i се рачуна на основу стања напона и деформација на почетку инкремента. Уколико се \mathbf{K}_t^i рачуна у свакој итерацији, онда је то пун Newton-Raphson-ов поступак. За одређивање матрице крутости елемената може да се поред еласто-пластичне конститутивне матрице \mathbf{D}^{ep} користи и еластична матрица \mathbf{D}^e . Избор конститутивне матрице за рачунање матрице крутости елемената не утиче на тачност решења, већ на брзину прорачуна.

У претходно описаном инкрементално-итеративном поступку за нелинеарну анализу у МКЕ, важан корак представља одређивање интеграла у изразу (2-92). Сем за веома једноставне конститутивне моделе (нпр. линеарно еластични конститутивни модел), тај интеграл нема аналитичко решење и потребно је коришћење неке нумеричке методе за његово рачунање. Нумеричке методе за рачунање интеграла (2-92) најчешће захтевају примену итеративних поступака. Због тога, у нелинеарној анализи у МКЕ за

решавање граничних проблема, уколико се користи еласто-пластични конститутивни модел, потребно је спровести две итеративне процедуре. Прва процедура у оквиру MNR поступка решава нелинеарне релације између оптерећења и померања користећи задате граничне услове (глобални ниво). Друга процедура у оквиру алгоритама за нумеричку интеграцију конститутивних релација решава нелинеарне релације између напона и деформација у тачкама нумеричке интеграције на нивоу елемента, користећи прираштај деформација из срачунатих чворних померања (локални ниво).



Слика 2-11 Newton-Raphson-ов итеративни поступак

2.7.3 Алгоритми за нумеричку интеграцију конститутивних релација

Као што је претходно приказано, да би могао да се спроведе MNR поступак у нелинеарној МКЕ, потребно је из познатог прираштаја деформација $\Delta\varepsilon$ у једном инкременту срачунати колики је прираштај напона $\Delta\sigma$. Израз (2-92) се за један инкремент може написати у облику:

$$\Delta\sigma = \int_{\varepsilon}^{\varepsilon+\Delta\varepsilon} \mathbf{D}^{ep} d\varepsilon \quad 2-97$$

За рачунање овог интеграла користе се нумеричке методе – методе за интеграцију напона (или алгоритми за нумеричку интеграцију конститутивних релација). Методе за интеграцију напона се генерално могу поделити на експлицитне и имплицитне методе.

2.7.3.1 Експлицитне методе интеграције

У експлицитним методама интеграције до прираштаја напона на крају инкремента се долази користећи стање напона и деформација на почетку инкремента:

$$\Delta\sigma = \mathbf{D}^{ep}({}^t\sigma, {}^t\mathbf{k})\Delta\varepsilon \quad 2-98$$

где су ${}^t\sigma$ и ${}^t\mathbf{k}$ напони и параметри стања на почетку инкремента (у конфигурацији t). Напони на крају инкремента су онда:

$${}^{t+\Delta t}\boldsymbol{\sigma} = {}^t\boldsymbol{\sigma} + \Delta\boldsymbol{\sigma}$$

2-99

Величина инкремента деформација $\Delta\boldsymbol{\varepsilon}$ утиче на тачност прорачуна, поготово када је понашање тла веома нелинеарно. Због тога се код експлицитних метода, инкремент деформација дели на већи број (n) мањих инкремената (субинкременти):

$$\Delta\boldsymbol{\varepsilon}_n = \frac{\Delta\boldsymbol{\varepsilon}}{n} \quad 2-100$$

Тако да се прираштаји напона и напони након сваког субинкремента могу одредити преко:

$$\Delta\boldsymbol{\sigma}_n = \mathbf{D}^{ep}(\boldsymbol{\sigma}_n, \mathbf{k}_n)\Delta\boldsymbol{\varepsilon}_n \quad 2-101$$

$$\boldsymbol{\sigma}_{n+1} = \boldsymbol{\sigma}_n + \Delta\boldsymbol{\sigma}_n \quad 2-102$$

Ознаке n и $n + 1$ се односе на претходно и ново стање. Што је већи број субинкремената, поступак одређивања напона је тачнији, али прорачун траје дуже. Најчешће се усваја да прираштаји деформација у оквиру једног субинкремента варирају на исти начин као и прираштаји деформација за цео инкремент. Ова претпоставка не мора увек бити тачна, што за последицу може имати нагомилавање грешке. Грешке у експлицитним методама се акумулирају и са сваким субинкрементом. Ако субинкременти нису довољно мали, срачунато стање напона и деформација може знатно одступати од тачног решења. Такође, на крају инкремента, уколико инкремент садржи еластичне и пластичне деформације, може се десити да напони и параметри стања не буду на површи течења, што није у складу са условом конзистенције пластичног течења. Ови недостаци се могу превазићи формулисањем адекватних алгоритама за одређивање величине субинкремента, за корекцију одступања стања напона од површи течења и контролу величине грешке у процесу интеграције. Помоћу тих алгоритама, експлицитне методе интеграције могу бити веома тачне, ефикасне и робустне и могу се на једноставан начин применити и за веома компликоване конститутивне моделе.

Допринос развоју експлицитних метода дали су (Nayak & Zienkiewicz, 1972), (Owen & Hinton, 1980) и (Wissman & Hauck, 1983). Експлицитна интеграција која користи модификован Euler-ов поступак и Runge-Kutta-Dormand-Price поступак вишег реда за нумеричку интеграцију са алгоритмом за аутоматску субинкрементацију и контролу грешке може се наћи у (Sloan, 1987) и (Sloan, et al., 2001). Упоредна анализа три алгоритама за експлицитну интеграцију унапређеног МСС модел на примеру неколико типичних геотехничких проблема може се наћи у (Gens & Potts, 1988). Дискусија о неким аспектима имплементације генерализованог МСС модела применом алгоритама за експлицитну интеграцију дата је у (Sheng, et al., 2000). Тачност и ефикасност експлицитне методе интеграције са аутоматском субинкрементацијом и контролом грешке на примеру два комплексна конститутивна модела формулисана у оквиру концепта критичног стања за структуриране глине и пескове је анализирана у (Zhao, et al., 2005). Имплементација UNSW (University of New South Wales model) конститутивног модела за пескове и глине применом експлицитног алгоритама са субинкрементацијом дата је у (E-Kan & Taiebat, 2013). Анализа величине грешке у експлицитном алгоритму са аутоматском субинкрементацијом и контролом грешке на примеру интеграција МСС модела спроведена је у (Lloret-Cabot, et al., 2016). Поступак за смањење величине

грешке у експлицитном алгоритму за интеграцију релација МСС модела дат је у (De la Moleña, et al., 2017). Анализа тачности, ефикасности и робустности експлицитног алгоритма са аутоматском субинкрементацијом, контролом грешке и корекцијом одступања од површи течења за имплементацију MIT-E3 модела спроведена је у (Dong, 2023). Нумеричка интеграција применом експлицитних алгоритама GCM (Glasgow Coupled Model) конститутивног модела, који је развијен у оквиру концепта критичног стања за незасићене материјале спроведена је у (Lloret-Cabot, et al., 2021). Поступак за корекцију одступања напона од површи течења у експлицитним методама на примеру S-CLAY1 модела приказан је у (Wang, et al., 2022). У (Hong, et al., 2012) дат је алгоритам за експлицитну методу интеграције термо-механичког модела за глине, а у (Solowski & Gallipoli, 2010a) и (Solowski & Gallipoli, 2010b) дат је алгоритам за експлицитну методу интеграције Barcelona Basic модела са анализом ефикасности и тачности алгоритма.

2.7.3.2 Имплицитне методе интеграције

У имплицитним методама се до прираштаја напона, прираштаја пластичних деформација и прираштаја параметара стања долази преко напона на крају инкремента који такође зависе од ових прираштаја. У већини имплицитних метода, први корак је еластично предвиђање, када су пластичне деформације у оквиру инкремента једнаке нули. Уколико се напонска тачка налази изван површи течења за еластично предвиђање, онда се врши корекција напонског стања, односно враћање на површ течења користећи неки итеративни поступак.

Један од првих имплицитних алгоритама дао је (Wilkins, 1963) за Mises-ов модел. Тај алгоритам познат је и под називом радијално повратно пресликавање (radial return). За идеално еласто-пластичне моделе, (Simo & Taylor, 1985) су предложили имплицитну методу интеграције (closest point algorithm), а (Ortiz & Popov, 1985) методу генерализованог трапезног правила која се може применити за неасоцијативну пластичност и било који закон ојачања. (Ortiz & Simo, 1986) и (Simo & Taylor, 1986) су објавили општи алгоритам повратног пресликавања који може да се примени код сложених еласто-пластичних модела са нелинеарним еластичним понашањем и комплексним површима течења, законима течења и ојачања. (Potts & Zdravković, 1999) наводе да алгоритам који су развили (Ortiz & Simo, 1986) не би требало да се користи за интеграцију релација конститутивних модела као што је МСС модел, због тога што се градијенти функције течења и пластичног потенцијала одређују за физички немогуће стање напона. Као последица тога, примена тог алгоритма може дати погрешан резултат у анализи граничних проблема.

Имплицитну интеграцију МСС модела и сличних модела, дали су (Borja & Lee, 1990) и (Borja, 1991). Овај алгоритам нема недостатке алгоритма (Ortiz & Simo, 1986). До данас је развијен велики број имплицитних алгоритама. С обзиром на велику разноликост и комплексност конститутивних модела, није могуће направити општи алгоритам који би важио за све моделе. За сваки појединачни конститутивни модел потребно је извести посебан имплицитни алгоритам. Што су конститутивни модели комплекснији, задатак интеграције конститутивних релација имплицитном методом постаје захтевнији. Експлицитни алгоритми су једноставнији за извођење, али тачност поступка интеграције зависи од величине инкремента, док тачност имплицитних алгоритама, иако могу бити знатно компликовани за извођење, генерално не зависи од

величине инкремента. Због тога субинкрементација углавном није потребна у имплицитним алгоритмима.

Метода водећег параметра (Governing Parameter Method – GPM) је имплицитна метода која представља генерализацију методе радијалног повратног пресликавања. У овој методи, све непознате величине треба да се изразе у функцији једног параметра (водећи параметар). Проблем интеграције напона се онда своди на решавање једне нелинеарне једначине са једном непознатом. Ову методу развио је (Кojić, 1996) за већи број еласто-пластичних модела за метале, а за тло (Кojić & Vukićević, 1992) и (Кojić, et al., 1994). Ову методу за имплементацију конститутивних модела за тло даље су развијали: (Кojić & Vukićević, 1998), (Кojić & Bathe, 2003), (Vukićević & Rakić, 2010), (Vukićević, 2010) и (Vukićević & Jocković, 2018).

Поред претходно набројаних имплицитних алгоритама, у литератури се могу наћи: (Rouainia & Wood, 2001), (Borja, et al., 2003), (Petalas & Dafalias, 2019), (Geng, et al., 2021), (Ledesma, et al., 2021), (Wei, et al., 2021), (Zhou, et al., 2022), (Zhao, et al., 2023) и (Chen, et al., 2024).

Такође, бројни радови пореде тачност, ефикасност и робустност експлицитних и имплицитних метода: (Potts & Ganendra, 1992), (Yamaguchi, 1993), (Potts & Ganendra, 1994), (Solowski, et al., 2012), (Hu & Liu, 2014), (Carow & Rackwitz, 2021) и (Lloret-Cabot & Sheng, 2022).

2.7.4 PLAXIS – рачунарски програм заснован на МКЕ за анализу геотехничких проблема

PLAXIS је рачунарски програм заснован на методи коначних елемената, који је развијен за анализу разних граничних проблема у геотехничком инжењерству. Развој PLAXIS-а започео је 1987. године на Техничком универзитету у Делфту (Delft University of Technology) на иницијативу холандског министарства за јавне послове и водопривреду. Идеја је била да се направи рачунарски програм заснован на МКЕ који је једноставан за коришћење за анализу насипа уз реке фундираних на меком тлу у Холандији. Од тада креће даљи развој PLAXIS-а, који се данас може користити за веома широк спектар геотехничких проблема. PLAXIS 2D за Windows оперативне системе се појавио 1998. године, а PLAXIS 3D 2010. године.

PLAXIS 2D се може користити за раванске проблеме као што су напонско-деформацијска анализа тунела, насипа, брана, ископа, тракастих темеља и друго. У оквиру PLAXIS 2D могуће је срачунати деформације, напоне и порне притиске у тлу, извршити прорачун струјања воде и топлоте кроз тло, као и прорачун статичких утицаја у елементима конструкција које су у интеракцији са тлом. У PLAXIS-у 3D могуће је анализирати било коју геотехничку конструкцију. У оквиру ове дисертације анализирани су само дводимензионални проблеми у PLAXIS-у 2D.

У оквиру PLAXIS-а имплементиран је велики број напредних конститутивних модела за тло и стене како би на адекватан начин могло да се моделира понашање тла и стене у разним геотехничким проблемима. Такође, програм има могућност да рачуна порне притиске у водом засићеном и делимично засићеном тлу, као и да моделује везу између тла и конструкције која је у интеракцији са тлом. Напредни конститутивни модели који су имплементирани у програму, омогућавају описивање еласто-пластичног понашања

тла са ојачањем или омекшањем, зависност крутости тла са величином деформација у тлу, пузање меког тла, ликвифакције песковитог тла током динамичке побуде, анизотропног понашања тла и другог. Имплементиран је и конститутивни модел за бетон који се користи за елементе конструкције.

PLAXIS омогућава корисницима да имплементирају било који конститутивни модел (UDSM – User-defined Soil Model). Рачунарски код потребан за имплементацију треба да буде написан у FORTRAN-у и компајлиран у DLL формат (Dynamic Link Library). UDSM треба да врати PLAXIS-у напоне и параметре стања на крају инкремента уколико су познати напони, параметри стања и деформације на почетку инкремента, као и прираштаји деформација и времена. Структура UDSM дата је у Прилогу 1.

Стања напона и деформација која се могу анализирати у 2D верзији су раванско стање деформације и стање осне симетрије. Раванско стање деформација се користи за геометрије и оптерећења са приближно константним попречним пресеком дуж осе (z оса) управне на раван попречног пресека. Померања и деформације у правцу те осе су једнаки нули, док су нормални напони у том правцу различити од нуле. Тунели, насипи, бране, тракасти темељи и потпорни зидови су примери геотехничких конструкција које се анализирају применом раванског стања деформација. Стање осне симетрије се користи за кружне конструкције код којих су радијални попречни пресеци и оптерећења приближно константни и напони и деформације су исти у било ком радијалном правцу. Оса симетрије је у оса а x оса је радијална оса. Стање осне симетрије се користи за анализу геотехничких проблема као што су темељи кружне основе, аксијално оптерећени шипови и пропација вибрација услед побуде у једној тачки (темељи машина).

PLAXIS 2D користи троугаоне коначне елементе са 6 чворова и троугаоне коначне елементе са 15 чворова. Елементи са 15 чворова омогућавају интерполацију четвртог реда за померање и садрже 12 тачака за нумеричку интеграцију напона. Елементи са 15 чворова омогућавају велику тачност прорачуна и представљају подразумевани тип елемената који се користи у програму. Елементи са шест чворова омогућавају интерполацију другог реда за померања и садрже три тачке за нумеричку интеграцију напона. Елементи са 6 чворова су мање тачни од елемената са 15 чворова, па је потребно да модели имају већи број таквих елемената како би давали резултате задовољавајуће тачности. Поред ових елемената, PLAXIS 2D користи линијске елементе за моделовање конструкција које су у интеракцији са тлом, геомреже и анкере, као и интерфејс елементе за симулацију интеракције тла и конструкције.

PLAXIS нуди неколико типова анализа као што су: Plastic (еласто-пластична напонско-деформацијска анализа), Consolidation (напонско-деформацијска анализа која узима у обзир развој и дисипацију додатних порних притисака у водом засићеном ситнозрном тлу), Safety (напонско-деформацијска анализа у којој је могуће рачунање глобалног фактора сигурности), Dynamic (напонско-деформацијска анализа која узима у обзир пропацију таласа и вибрација у тлу), Dynamic with consolidation (напонско-деформацијска анализа која узима у обзир генерисање и дисипацију порних притисака током краткотрајних динамичких побуда у тлу) и Fully coupled flow-deformation (спрегнута напонско-деформацијска и филтрациона анализа). Пре спровођења прорачуна, програм генерише почетне напоне и порне притиске у тлу.

PLAXIS генерално спроводи напонско-деформацијску анализу на основу параметара за ефективне напоне, односно рачуна ефективне напоне. Сви параметри конститутивних модела који се користе су параметри за ефективне напона. У зависности од водопропустљивости тла, услед наношења оптерећења могу се генерисати додатни порни притисци. Због тога, PLAXIS 2D омогућава спровођење дрениране анализе (дугорочна анализа) и недрениране анализе (краткорочна анализа). У случају дрениране анализе, нема генерисања додатних порних притисака, док у случају недрениране анализе се генеришу додатни порни притисци.

Дренирана анализа се примењује за сува тла или водом засићена тла велике водопропусности (крупнозрна тла). Може се применити и за тла мале водопропусности када је брзина наношења оптерећења мала, тако да се генерисани додатни порни притисци могу занемарити.

Недренирана анализа се примењује за водом засићена тла када је спречено струјање воде као што су ситнозрна тла мале водопропусности или када је брзина наношења оптерећења велика. У оквиру PLAXIS програма могуће је спровести три типа недрениране анализе. Тип А (Undrained A) је недренирана анализа ефективним напонима преко параметара отпорности и деформабилности за ефективне напоне. У типу А, програм даје предвиђање колики ће бити додатни порни притисци. Чврстоћа тла у недренираним условима у типу А је резултат анализе у моделу и није улазни параметар. Тип В (Undrained B) је недренирана анализа ефективним напонима преко параметара деформабилности за ефективне напоне и параметара отпорности за тоталне напоне. Као и код типа А, у типу В се дају предвиђања величине порних притисака. Недренирана чврстоћа тла је улазни параметар у типу В. Тип С (Undrained C) је недренирана анализа тоталним напонима са параметрима отпорности и деформабилности за тоталне напоне. У овом типу се не могу одредити колики су генерисани порни притисци.

Кориснички конститутивни модели могу да користе само Тип А недренирану анализу. Сви улазни параметри конститутивног модела који описују отпорност и деформабилност су параметри за ефективне напона. PLAXIS у типу А недрениране анализе аутоматски додаје запремински модул воде K_w запреминском модулу K' у конститутивној матрици (2-60). У PLAXIS-у принцип ефективних напона је дат у облику:

$$\sigma = \sigma' + m p_{active} \quad 2-103$$

У претходном изразу, σ и σ' су вектори тоталних и ефективних напона, m је вектор:

$$m^T = (1 \quad 1 \quad 1 \quad 0 \quad 0 \quad 0)^T \quad 2-104$$

а p_{active} су активни порни притисци:

$$p_{active} = \alpha_{Biot} S_{eff} p_w \quad 2-105$$

где су α_{Biot} Biot-ов коефицијент порних притисака, S_{eff} је ефективни степен засићења и p_w су порни притисци који се могу представити као:

$$p_w = p_{steady} + p_{excess} \quad 2-106$$

У претходном изразу p_{steady} су устаљени порни притисци, односно притисци који се током времена не мењају и представљају улазни параметар у прорачуну, а p_{excess} су додатни порни притисци који настају као резултат наношења оптерећења на тло. За засићена тла важи $S_{eff} = 1$ и $\alpha_{Biot} = 1$ за нестишљива зрна, док за стишљива зрна је $\alpha_{Biot} < 1$. PLAXIS прираштаје додатних порних притисака рачуна преко израза:

$$\Delta p_{excess} = \frac{\alpha_{Biot} \Delta \varepsilon_v}{\frac{n}{K_w} + \frac{\alpha_{Biot} - n}{K_s}} \quad 2-107$$

У претходном изразу, n је коефицијент порозности а K_s запремински модул чврстих честица тла. За нестишљиве честице тла, K_s има веома велику вредност и $\alpha_{Biot} = 1$, па израз 2-107 се своди на:

$$\Delta p_{excess} = \frac{K_w}{n} \Delta \varepsilon_v \quad 2-108$$

За линеарно еластичан материјал, прираштај тоталног и ефективног средњег напона су:

$$\Delta p = K_u \Delta \varepsilon_v \quad 2-109$$

$$\Delta p' = K' \Delta \varepsilon_v \quad 2-110$$

Применом принципа ефективних напона добија се:

$$K_u = K' + \frac{K_w}{n} \quad 2-111$$

Запремински модул за тоталне напоне K_u и запремински модул за ефективне напоне K' се могу изразити преко модула смицања G (који је исти и за тоталне и за ефективне напоне) и Poisson-ових коефицијента за тоталне и ефективне напоне, μ_u и μ' :

$$K_u = \frac{2G(1 + \mu_u)}{3(1 - 2\mu_u)} \quad 2-112$$

$$K' = \frac{2G(1 + \mu')}{3(1 - 2\mu')} \quad 2-113$$

Комбинујући изразе (2-111), (2-112) и (2-113), добија се израз за запремински модул воде:

$$\frac{K_w}{n} = \frac{3(\mu_u - \mu')}{(1 - 2\mu_u)(1 + \mu')} K' \quad 2-114$$

За Poisson-ов коефицијент за тоталне напоне, PLAXIS узима вредност $\mu_u = 0.495$ уместо вредност 0.5 како би се избегли нумерички проблеми. Програм рачуна модул стишљивости воде према изразу (2-114) који даље користи за прорачун деформација у тлу (додавањем K_w у конститутивну матрицу) и за прорачун додатних порних

притисика. Програм нуди и као алтернативну верзију да корисник сам зада вредности K_w и α_{Biot} .

На претходно описан начин, недренирана анализа се може спровести користећи параметре отпорности и деформабилности за ефективне напоне. Чврстоћа тла у недренираним условима је онда резултат прорачуна заснованих на дренираним параметрима и може знатно одступати од стварне вредности, због чега је потребна велика опрезност у коришћењу добијених резултата.

Више о рачунарском програму PLAXIS може се наћи у приручницима (Bentley Systems, Incorporated, 2023).

Поглавље 3 Проширени HASP конститутивни модел за преконсолидоване глине – HASP-E модел

3.1 Увод

HASP (HArdening State Parameter) конститутивни модел (Jocković & Vukićević, 2017) је модел критичног стања тла и представља надоградњу MCC модела. HASP модел користи концепт граничне површи и комбиновани закон ојачања који је модификован тако да укључује параметар стања тла (state parameter) као битан показатељ понашања тла. У поређењу са MCC моделом омогућава реалистичније описивање механичког понашања преконсолидованих глина користећи исте параметре модела. Увођење параметра стања у напонско-деформацијске релације конститутивног модела, омогућено је предвиђање понашање преконсолидованих глина као што су ојачање, достизање вршне чврстоће и омекшање у дренираним условима, као и развој негативних порних притисака у недренираним условима.

HASP модел се заснива на следећим принципима (Jocković, 2017):

- Тло је изотропно.
- Од самог почетка оптерећивања, истовремено са развојем еластичних деформација развијају се и пластичне деформације.
- Нормално консолидоване до благо преконсолидоване глине смањују запремину током смицања у дренираним условима и прираштаји порног притиска су позитивни у недренираним условима, док се код јако преконсолидованих глина повећава запремина (дилатанција) у дренираним условима и прираштаји порног притиска су негативни у недренираним условима.
- У дренираним условима преконсолидоване глине показују омекшање након достизања вршне чврстоће.
- У недренираним условима нема омекшања.
- Параметар ојачања у моделу зависи од прираштаја пластичних запреминских деформација и прираштаја пластичних смичућих деформација.
- За опис комплетне конститутивне везе користе се само параметри MCC модела који се могу одредити из конвенционалних лабораторијских опита.

У даљем тексту термин „основна формулација HASP модела“, или HASP односи се на формулацију HASP модела описану у (Jocković & Vukićević, 2017) и (Jocković, 2017). Примена основне формулације HASP модела на предвиђање понашања слоја преконсолидоване глине услед изградње насипа може се наћи у (Jocković & Vukićević, 2018a). Значај увођења параметра стања тла као битног индикатора понашања тла у основну формулацију HASP модела приказан је у (Jocković, et al., 2021). Могућност HASP модела да адекватно предвиди понашање засићене глине услед монотонно растућег оптерећења у дренираним и недренираним условима је приказана у (Jocković & Vukićević, 2018b).

На бази HASP модела развојни тим PLAXIS-a је формулисао модел под именом OC_CLAY (Bentley, 2019). Релације HASP модела су формулисане у триаксијалној ($p' - q$) равни

док су релације OC_CLAY модела проширене на глобални напонски простор укључивањем треће напонске инваријанте (Lode-овог угла). У оквиру OC_CLAY модела, релације HASP модела су додатно проширене да обухвате деградацију модула смицања са порастом смичућих деформација. Такође, постоји разлика у дефинисању параметара деформабилности OC_CLAY модела у односу на HASP модел.

Као наставак развоја HASP модела, у оквиру ове докторске дисертације извршено је проширење основне формулације модела на генерални напонски простор као и обухватање подручја малих деформација са карактеристикама хистерезисног понашања. За овако проширену формулацију HASP модела у наставку дисертације биће коришћен назив HASP-E модел (HASP-Extended)

У потпоглављу 3.2 дат је детаљан приказ напонско-деформацијских релација и параметара основне формулације HASP модела.

У потпоглаву 3.3 дат је опис релација преко којих је извршено проширивање основне формулације HASP модела на генерални напонски простор, релација преко којих је узета у обзир крутост тла у подручју веома малих и малих деформација и начина на који се може узети у обзир хистерезисно понашање тла. На крају је дат приказ напонско-деформацијских релација и параметара HASP-E модела.

У потпоглављу 3.4 је дат опис OC_CLAY модела (Bentley, 2019) за преконсолидоване глине. OC_CLAY модел је у дисертацији послужио као упоредни модел за верификацију HASP-E модела из три разлога: проширење релација HASP модела је извршено по угледу на OC_CLAY модел, OC_CLAY модел је, као и HASP-E модел, као основни модел користио HASP, и имплементиран је као напредни модел у програмском пакету PLAXIS.

3.2 Напонско-деформацијске релације HASP модела

HASP модел користи површ течења MCC модела као граничну површ. Гранична површ (површ нормалне консолидације) има све особине као површ течења MCC модела: запремина се смањује и површ се шири за напонске односе испод линије критичног стања, док за напонски однос изнад линије критичног стања запремина расте и површ се скупља. Величину граничне површи дефинише максимални средњи ефективни напон \bar{p}'_0 (Слика 3-1 а). Тачка $A(p', q)$ представља тренутно напонско стање и налази се на унутрашњој површи течења (у даљем тексту површ течења) чија је величина дефисана преко средњег ефективног напона p'_0 (Слика 3-1 а). Површ течења се шири док се не достигне вршна чврстоћа за напонски однос $\eta = M_f$, након чега се скупља док се не достигне критично стање. Код HASP модела нема чисто еластичних деформација и напонска тачка A се налази увек на површи течења. Тачки A одговара имагинарна тачка $\bar{A}(\bar{p}', \bar{q})$ на граничној површи тако да према радијалном мапирању важи:

$$\eta = \frac{q}{p'} = \frac{\bar{q}}{\bar{p}'} \quad 3-1$$

Површ течења и гранична површ у девијаторској равни имају облик круга. Једначина површи течења је:

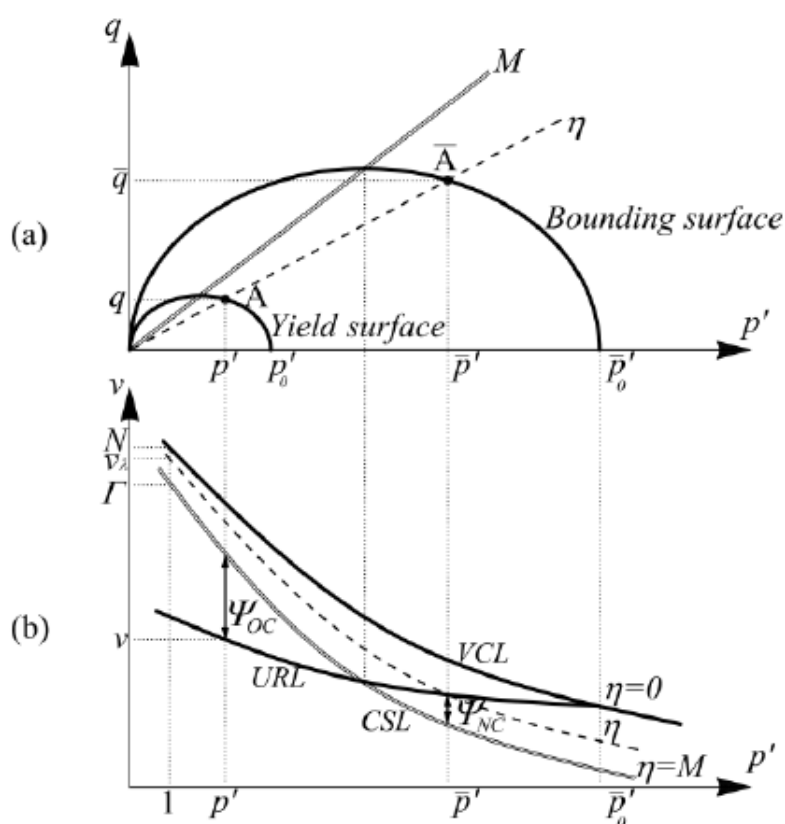
$$F(p', q, p'_0) = \frac{q^2}{M^2} + p'(p' - p'_0) = 0 \quad 3-2$$

Једначина граничне површи је:

$$\bar{F}(\bar{p}', \bar{q}, \bar{p}'_0) = \frac{\bar{q}^2}{M^2} + \bar{p}'(\bar{p}' - \bar{p}'_0) = 0 \quad 3-3$$

HASP модел користи асоцијативни закон течења. Једначина функције пластичног потенцијала је:

$$P(p', q, p'_0) = \frac{q^2}{M^2} + p'(p' - p'_0) = 0 \quad 3-4$$



Слика 3-1 (a) Концепт граничне површи; (b) Параметар стања – преузето из (Jocković & Vukićević, 2017)

HASP модел користи комбиновани закон ојачања за површ течења који зависи од пластичне запреминске деформације и пластичне смичуће деформације. Разлог за то је што закон ојачања који зависи само од пластичне запреминске деформације, што је случај код MCC модела, не омогућава адекватно описивање дилатанције и ојачања за преконсолидована тла. Употреба комбиновог закона ојачања омогућава да површ течења наставља да се шири и за вредности напонског односа $M < \eta < M_f$. Закон ојачања за HASP модел гласи:

$$dp'_0 = \frac{v}{\lambda - \kappa} p'_0 (d\varepsilon_v^p + \xi d\varepsilon_q^p) \quad 3-5$$

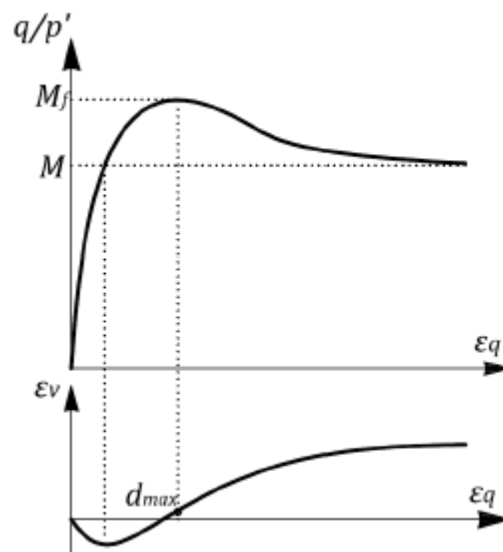
Параметар ξ представља максималну вредност дилатанције при лому у дренираним условима (Јосковић, 2017) (Слика 3-2):

$$\xi = -d_{max} \quad 3-6$$

где је дилатанција однос пластичне запреминске и пластичне смичуће деформације:

$$d = \frac{d\varepsilon_v^p}{d\varepsilon_q^p} \quad 3-7$$

Комбиновани закон ојачања значајно утиче на путање напона. Омогућава да путања ефективних напона пређе преко линије критичног стања и достигање вршне чврстоће у дренираним условима. У недренираним условима, комбиновани закон ојачања је неопходан за описивање путања ефективног напона облика латичног слова „S“.



Слика 3-2 Максимални градијент промене запремине – преузето из (Јосковић, 2017)

Закон ојачања дат у (3-5) може се написати у следећем облику:

$$dp'_0 = \frac{\nu}{\lambda - \kappa} p'_0 d\varepsilon_v^p \omega \quad 3-8$$

где је ω коефицијент ојачања:

$$\omega = \left(1 + \frac{\xi}{d}\right) R \quad 3-9$$

R је тренутни степен изотропне преконсолидације:

$$R = \frac{\bar{p}'}{p'} = \frac{\bar{q}}{q} = \frac{\bar{p}'_0}{p_0} \quad 3-10$$

Изрази за пластичне деформације у HASP моделу су:

$$d\varepsilon_v^p = \frac{\lambda - \kappa}{\nu} \frac{1}{p'} \frac{1}{\omega} \left(\frac{M^2 - \eta^2}{M^2 + \eta^2} dp' + \frac{2\eta}{M^2 + \eta^2} dq \right) \quad 3-11$$

$$d\varepsilon_q^p = \frac{\lambda - \kappa}{\nu} \frac{1}{p'} \frac{1}{\omega} \left(\frac{2\eta}{M^2 + \eta^2} dp' + \frac{4\eta^2}{(M^2 + \eta^2)(M^2 - \eta^2)} dq \right) \quad 3-12$$

Коефицијент ојачања ω у изразима (3-11) и (3-12) представља и коефицијент редукције пластичних деформација. Као последица тога, пластичне деформације преконсолидоване глине у почетној фази оптерећивања могу бити значајно редуковане за разлику од МСС модела који у истој фази предвиђа искључиво еластичне деформације. Из претходног следи да се у HASP моделу у преконсолидованом тлу јављају и пластичне деформације од самог почетка оптерећивања. Пластичне деформације су у почетку веома мале, али се током процеса деформисања смањује степен преконсолидације тла, коефицијент ојачања ω се смањује (3-9), што даје као резултат све веће пластичне деформације. За реализацију претходно описаних особина HASP модела, неопходна је адекватна формулација коефицијента ојачања. У HASP моделу је усвојен следећи израз за одређивање коефицијента ојачања:

$$\omega = \left(1 + \frac{\bar{\Psi} - \Psi}{\bar{\Psi}} \right) R \quad 3-13$$

У изразу (3-13), Ψ представља параметар стања за тренутну напонску тачку A , а $\bar{\Psi}$ представља параметар стања за имагинарну напонску тачку \bar{A} (Слика 3-1b). Параметар стања Ψ се дефинише као разлика између тренутне вредности специфичне запремине и специфичне запремине на референтној линији стања (линији критичног стања) за исту вредност средњег ефективног напона:

$$\Psi = \nu + \lambda \ln p' - \Gamma \quad 3-14$$

Иницијална вредност параметра стања је већа од нуле ($\Psi > 0$) за нормално консолидоване и благо преконсолидоване глине ($OCR \leq 2$), а за јако преконсолидоване глине је мања од нуле ($\Psi < 0$). За напонску тачку на линији критичног стања $\Psi = 0$. Израз за параметар стања за имагинарну напонску тачку $\bar{\Psi}$ је:

$$\bar{\Psi} = (\lambda - \kappa) \ln \left(\frac{2M^2}{M^2 + \eta^2} \right) \quad 3-15$$

За напонске тачке испод линије критичног стања $\bar{\Psi} > 0$, а за напонске тачке изнад линије критичног стања $\bar{\Psi} < 0$. Степен изотропне преконсолидације се може изразити преко параметара стања:

$$R = \exp \left(\frac{\bar{\Psi} - \Psi}{\lambda - \kappa} \right) \quad 3-16$$

Детаљно извођење израза (3-13), (3-14), (3-15) и (3-16) може се наћи у (Јосковић, 2017). Уколико се изрази (3-11) и (3-12) упореде са истим изразима за МСС модел, уочава се да је главна разлика између HASP модела и МСС модела у коефицијенту ојачања ω . За

нормално консолидоване глине коефицијент ојачања има вредност $\omega = 1$ јер је $\Psi = \bar{\Psi}$. У том случају HASP модел предвиђа исто понашање тла као и MCC модел.

Изрази за еластичне деформације HASP модела су:

$$d\varepsilon_v^e = \kappa \frac{dp'}{vp'} = \frac{1}{K} dp' \quad 3-17$$

$$d\varepsilon_q^e = \frac{1}{3G} dq \quad 3-18$$

где је K запремински модул:

$$K = \frac{vp'}{\kappa} \quad 3-19$$

а G модул смицања:

$$G = \frac{3(1 - 2\mu)}{2(1 + \mu)} K \quad 3-20$$

Изрази за еластичне деформације (3-17) и (3-18) у HASP моделу су исте као у MCC моделу. Комплетне конститутивне релације HASP модела су:

$$\begin{Bmatrix} d\varepsilon_v \\ d\varepsilon_q \end{Bmatrix} = \begin{Bmatrix} \frac{1}{K} + \frac{\lambda - \kappa M^2 - \eta^2}{vp' \omega M^2 + \eta^2} & \frac{\lambda - \kappa}{vp' \omega} \frac{2\eta}{M^2 + \eta^2} \\ \frac{\lambda - \kappa}{vp' \omega} \frac{2\eta}{M^2 + \eta^2} & \frac{1}{3G} + \frac{\lambda - \kappa}{vp' \omega} \frac{4\eta^2}{(M^2 + \eta^2)(M^2 - \eta^2)} \end{Bmatrix} \begin{Bmatrix} dp' \\ dq \end{Bmatrix} \quad 3-21$$

3.2.1 Параметри HASP модела

Да би се комплетно описале све конститутивне релације HASP модела, потребни су следећи параметри: консолидациони параметри λ и κ , параметар који описује дренарану смичућу чврстоћу тла M , параметри еластичности μ или G , коефицијент порозности као параметар који описује иницијално стање e_0 и параметри који описују степен преконсолидације тла Γ , OCR или POP . HASP модел користи исте параметре као и MCC модел. Сви претходно набројани параметри могу се одредити из конвенционалних лабораторијских опита.

Консолидациони параметри λ и κ представљају нагибе линије нормалне консолидације и линије бубрења у условима изотропне компресије у $\ln p' - v$ дијаграму (Слика 2-3). Да би модел понашања тла био једноставнији, нагиб линије бубрења је исти као нагиб линије поновног оптерећивања, чиме је занемарена хистерезисна петља која је јавља приликом растерећивања и поновног оптерећивања. Ови параметри се могу одредити из опита изотропне компресије или едометарског опита.

Нагиб линије нормалне консолидације у едометарским условима је исти као нагиб линије нормалне консолидације и линије критичног стања у условима изотропне компресије. Параметар λ се може одредити уколико је познат индекс стишљивости C_c :

$$\lambda = \frac{C_c}{\ln 10} \quad 3-22$$

Параметар κ се може одредити уколико је познат индекс рекомпресије C_R :

$$\kappa = \frac{C_R}{\ln 10} \quad 3-23$$

Параметар дренаране смичуће чврстоће тла M представља нагиб линије критичног стања у $p' - q$ дијаграму (Слика 3-3). Параметар M се може одредити из резултата стандардних опита триаксијалне компресије или опита директног смицања на основу угла смичуће отпорности тла при константној запремини у дренараним условима ϕ'_{cv} . За компресију:

$$M_c = \frac{6 \sin \phi'_{cv}}{3 - \sin \phi'_{cv}} \quad 3-24$$

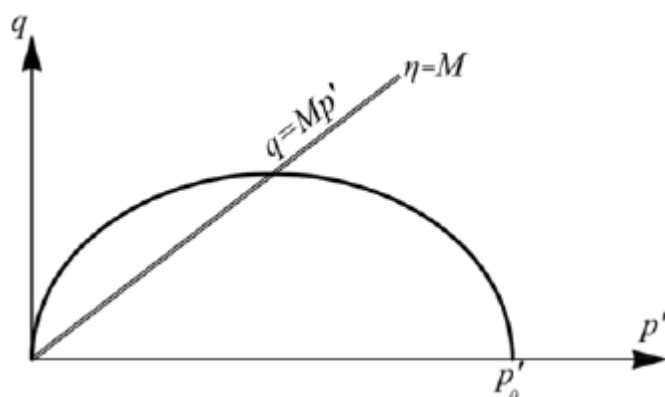
За екстензију:

$$M_e = \frac{6 \sin \phi'_{cv}}{3 + \sin \phi'_{cv}} \quad 3-25$$

За раванско стање деформација може се узети за M :

$$M = \sqrt{3} \sin \phi'_{cv} \quad 3-26$$

Код HASP модела, као и код MCC модела, параметар M је константан током целокупног оптерећивања тла. Углови смичуће отпорности тла при триаксијалној компресији и триаксијалној екстензији могу бити различити ($\phi'_C \neq \phi'_E$), и у том случају M_c и M_e се рачунају са одговарајућим угловима, M_c са ϕ'_C и M_e са ϕ'_E .



Слика 3-3 Линија критичног стања у $p' - q$ дијаграму – преузето из (Vukićević & Jocković, 2019)

Параметри који описују еластично понашање тла су модул смицања G или Poisson-ов коефицијент μ . Да би се потпуно описало еластично понашање тла потребан је и запремински модул K који се у HASP моделу одређује према изразу (3-19). Уколико је задат као улазни параметар Poisson-ов коефицијент, онда се модул смицања у HASP моделу одређује преко израза (3-20). Уколико је задат као улазни параметар модул смицања, онда су K и G међусобно независни. Као последица тога и ако модул смицања има константну вредност током оптерећивања, вредност Poisson-овог коефицијента у моделу може бити већа од 0.5 или мања од 0 што је физички немогуће. У том случају, потребно је вршити корекцију модула како би вредност Poisson-овог коефицијента била у границама $\mu \in (0,0.5)$. У основној формулацији HASP модела, усвојено је да Poisson-ов коефицијент улазни параметар и да има константну вредност током оптерећивања тла. Због тога, модул смицања у HASP моделу није константан.

Иницијална (или почетна) вредност коефицијента порозности e_0 је неопходан параметар за одређивање почетне вредности параметра стања Ψ у изразу (3-14) из којег се даље одређује степен преконсолидације и понашање глине у HASP моделу. Такође, из e_0 се рачуна промена коефицијента порозности e током оптерећивања глине. Из тренутне вредности коефицијента порозности e рачунају се еластични модули према изразима (3-19) и (3-20), као и параметар стања према изразу (3-14). Због тога је e_0 есенцијални параметар у HASP моделу за описивање понашања преконсолидоване глине.

Да би се у потпуности описало понашање преконсолидоване глине у HASP моделу, неопходан параметар је Γ који представља специфичну запремину тла при $p' = 1 \text{ kPa}$ на линији критичног стања. Помоћу параметра Γ се рачуна параметар стања у изразу (3-14). Параметар Γ се може одредити из параметра N који представља специфичну запремину нормално консолидованог тла при $p' = 1 \text{ kPa}$ у условима изотропне компресије:

$$\Gamma = N - (\lambda - \kappa) \ln 2 \quad 3-27$$

Уколико су као улазни параметри задати e_0 и Γ , онда је у потпуности дефинисано почетно стање преконсолидоване глине. Са променом коефицијента порозности мења се и стање преконсолидоване глине, па се преко израза (3-13), (3-14), (3-15) и (3-16) може срачунати коефицијент ојачања ω . Параметар Γ се не мења током оптерећивања глине. Из претходног се може закључити да се преконсолидација глине у HASP моделу може задати преко параметара e_0 и Γ .

Преко консолидација глине се може задати и преко параметара OCR или POP (видети потпоглавље 2.7.4). У том случају потребно је дефинисати релацију између параметра Γ и та два параметра како би могле да се користе релације HASP модела. У наставку је дат поступак извођења тих релација. Поступак извођења је преузет из OC_CLAY модела (Bentley, 2019) са одговарајућим изменама.

Иницијално напонско стање пре почетка оптерећивања глине је дефинисано преко ефективног вертикалног напона σ'_{vi} . Бочни ефективни напони се могу одредити као:

$$\sigma'_{hi} = K_0^{OC} \sigma'_{vi} \quad 3-28$$

где је K_0^{OC} коефицијент притиска тла у стању мировања за преконсолидовано тло. Иницијалне напонске инваријанте су:

$$p'_i = \frac{\sigma'_{vi} + 2\sigma'_{hi}}{3} = \sigma'_{vi} \frac{1 + 2K_0^{OC}}{3} \quad 3-29$$

$$q_i = |\sigma'_{vi} - \sigma'_{hi}| = \sigma'_{vi} |1 - K_0^{OC}| \quad 3-30$$

Напони који су деловали у тлу пре преконсолидације тла, односно када је тло било последњи пут нормално консолидовано се могу одредити преко параметара OCR или POP :

$$\sigma'_{vp} = OCR \cdot \sigma'_{vi} \quad 3-31$$

или

$$\sigma'_{vp} = \sigma'_{vi} + POP \quad 3-32$$

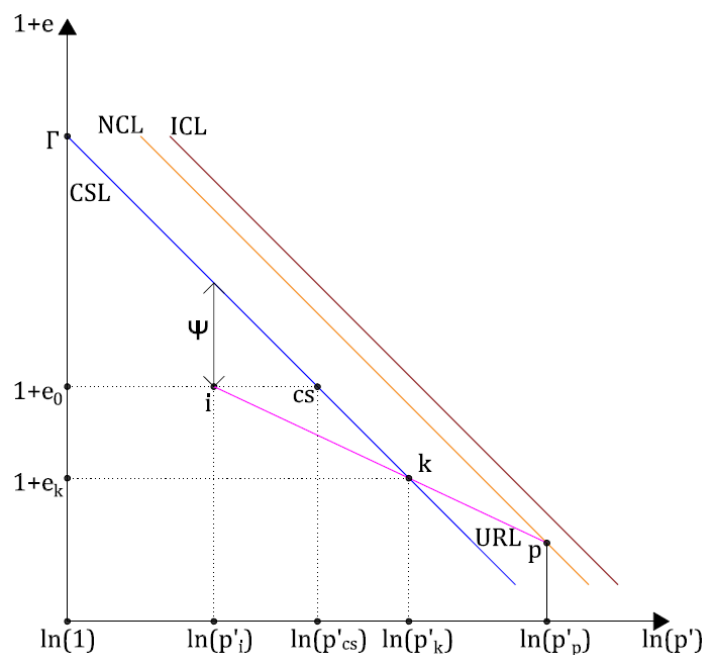
$$\sigma'_{hp} = K_0^{NC} \sigma'_{vp} \quad 3-33$$

где је K_0^{NC} коефицијент притиска тла у стању мировања за нормално консолидовано тло. Напонске инваријанте за ово претходно напонско стање су:

$$p'_p = \frac{\sigma'_{vp} + 2\sigma'_{hp}}{3} = \sigma'_{vp} \frac{1 + 2K_0^{NC}}{3} \quad 3-34$$

$$q_p = |\sigma'_{vp} - \sigma'_{hp}| = \sigma'_{vp} |1 - K_0^{NC}| \quad 3-35$$

$$\eta_p = \frac{q_p}{p'_p} = 3 \frac{|1 - K_0^{NC}|}{1 + 2K_0^{NC}} \quad 3-36$$



Слика 3-4 Одређивање параметра Γ - адаптирано из (Bentley, 2019)

Ако је e_0 иницијална вредност параметра стања, онда тој вредности одговара средњи ефективни напон p'_{cs} на линији критичног стања у $\ln p' - 1 + e_0$ дијаграму (Слика 3-4) тако да важи:

$$1 + e_0 = \Gamma - \lambda \ln p'_{cs} \quad 3-37$$

односно

$$\Gamma = 1 + e_0 + \lambda \ln p'_{cs} \quad 3-38$$

Из израза (3-38) потребно је одредити p'_{cs} да би се добио параметар Γ . Ако се из тачке $(\ln p'_i, 1 + e_0)$ повуче линија рекомпресије (Слика 3-4), пресећи ће линију критичног стања у тачки $(\ln p'_k, 1 + e_k)$ тако да важи:

$$1 + e_k = 1 + e_0 - \kappa \ln \frac{p'_k}{p'_i} \quad 3-39$$

Пошто се тачка $(\ln p'_k, 1 + e_k)$ налази на линији критичног стања, важи и (Слика 3-4):

$$1 + e_k = 1 + e_0 - \lambda \ln \frac{p'_k}{p'_{cs}} \quad 3-40$$

Ако се изрази (3-39) и (3-40) изједначе, добија се:

$$\kappa \ln \frac{p'_k}{p'_i} = \lambda \ln \frac{p'_k}{p'_{cs}} \quad 3-41$$

Сређивање претходног израза, добија се:

$$p'_{cs} = p'_i{}^{\frac{\kappa}{\lambda}} \cdot p'_k{}^{\frac{\lambda-\kappa}{\lambda}} \quad 3-42$$

Сада је потребно одредити средњи ефективни напон p'_k . Као што је раније написано, овај напон представља пресек линије критичног стања и линије рекомпресије и може се одредити из величине граничне површи $\bar{p}'_{0p} = 2p'_k$ (Слика 3-1), односно:

$$p'_k = \frac{\bar{p}'_{0p}}{2} \quad 3-43$$

Једначина граничне површи је:

$$\frac{q_p^2}{M^2} + p'_p(p'_p - p'_{0p}) = 0 \quad 3-44$$

Претходни израз може да се напише у следећем облику:

$$\frac{M^2}{\eta_p^2 + M^2} = \frac{p'_p}{\bar{p}'_{0p}} \quad 3-45$$

Комбинујући изразе (3-43) и (3-44), добија се:

$$p'_k = \left(\frac{\eta_p^2 + M^2}{2M^2} \right) \cdot p'_p \quad 3-46$$

У изразу (3-46), параметар M има вредност која одговара условима триаксијалне компресије $M = M_c$. Када је тло последњи пут било нормално консолидовано, K_0^{NC} је било мање од 1.0. У том случају вертикални ефективни напон је био већи од оба бочна ефективна напона ($\sigma'_{vp} > \sigma'_{hp}$), што одговара условима триаксијалне компресије. Ако се у израз (3-42) убаци израз (3-46), добија се:

$$p'_{cs} = \left[\left(\frac{\eta_p^2 + M^2}{2M^2} \right) \cdot p'_p \right]^{\frac{\lambda-\kappa}{\lambda}} \cdot p'_i{}^{\frac{\kappa}{\lambda}} \quad 3-47$$

Коначно, када се у израз (3-38) убаци израз (3-47) добија се израз за одређивање параметра Γ :

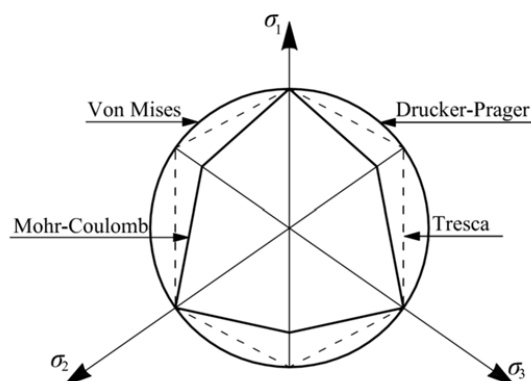
$$\Gamma = 1 + e_0 + \lambda \ln \left\{ \left[\left(\frac{\eta_p^2 + M^2}{2M^2} \right) \cdot p'_p \right]^{\frac{\lambda-\kappa}{\lambda}} \cdot p'_i{}^{\frac{\kappa}{\lambda}} \right\} \quad 3-48$$

Из претходних израза следи да је потребно дефинисати коефицијенте бочних притисака тла за нормално консолидовано и преконсолидовано тло K_0^{NC} и K_0^{OC} како би се одредили сви параметри HASP модела.

3.3 Проширење HASP модела - HASP-E модел

3.3.1 Проширивање релација HASP модела на генерални напонски простор узимањем у обзир утицај треће напонске инваријанте (Lode-ов угао)

Релације HASP модела описане у потпоглављу 3.2 су формулисане за стање напона у триаксијалном опиту. У триаксијалном опиту, средњи главни напон је једнак једном од друга два главна напона, $\sigma_1 > \sigma_2 = \sigma_3$ за триаксијалну компресију, односно $\sigma_1 = \sigma_2 > \sigma_3$ за триаксијалну екстензију и његов утицај није узет у обзир приликом извођења релација основне формулације HASP модела. У нумеричкој анализи разних геотехничких проблема, стање напона је такво да су сва три главна напона различита и утицај средњег главног напона мора се на одговарајући начин узети приликом формулисања релација конститутивног модела. Утицај средњег главног напона на понашање тла се може узети у обзир тако што ће се увести одговарајуће измене у критеријум лома, односно у облик површи течења и површ пластичног потенцијала у девијаторској равни. Класични критеријуми лома као што су Tresca-ин, von Mises-ов, Mohr-Coulomb-ов (MC) и Drucker-Prager-ов (DP) занемарују утицај средњег главног напона σ_2 чији утицај може бити веома значајан за услове раванског стања деформација. Облик класичних критеријума лома у девијаторској равни приказан је на **Слика 3-5**.



Слика 3-5 Класични критеријуми лома у девијаторској равни – преузето из (Vukićević & Jocković, 2019)

Основна формулација HASP модела користи DP услов лома, који подразумева круг у девијаторској равни. DP критеријум је због свог облика једноставан за имплементацију у рачунарске програме за нумеричку анализу. Са друге стране, коришћење овог критеријума за путање напона које одступају од триаксијалне компресије или екстензије даје велика одступања у предвиђању понашања тла у односу на експерименталне резултате (Jocković, 2017). Такође, уочено је да постоји разлика између угла смичуће чврстоће у опиту триаксијалне компресије у односу на опит у условима раванског стања деформација, као и да та разлика расте са збијеношћу тла. Разлике потичу од различитог доприноса ефеката дилатанције који зависи од средњег главног напона и укупног нивоа напона. Генерално се у опиту у условима раванског стања деформација добијају већи углови смичуће отпорности у односу на услове триаксијалне компресије и те разлике могу износити и до 20 % за збијене материјале при ниском нивоу нормалних напона (Maksimović, 2008).

Утицај средњег главног напона се може узети у обзир у HASP моделу тако што би се конститутивне релације прошириле на генерални напонски простор укључивањем треће напонске инваријанте – Lode-овог угла θ . Трећа напонска инваријанта би се укључила у релације HASP модела тако што би се у изразима за површ течења, граничну површ и површ пластичног потенцијала константа M заменила са функцијом која зависи од Lode-овог угла $M(\theta)$:

$$F(p', q, \theta, p'_0) = \frac{q^2}{M_F^2(\theta)} + p'(p' - p'_0) = 0 \quad 3-49$$

$$\bar{F}(\bar{p}', \bar{q}, \theta, \bar{p}'_0) = \frac{\bar{q}^2}{M_{\bar{F}}^2(\theta)} + \bar{p}'(\bar{p}' - \bar{p}'_0) = 0 \quad 3-50$$

$$P(p', q, \theta, p'_0) = \frac{q^2}{M_P^2(\theta)} + p'(p' - p'_0) = 0 \quad 3-51$$

У зависности да ли су функције $M_F(\theta)$ и $M_P(\theta)$ међусобно једнаке, зависи да ли ће проширен HASP модел користити асоцијативан или неасоцијативан закон течења. Функције $M_F(\theta)$ и $M_P(\theta)$ треба формулисати тако да на адекватан начин буде описан критеријум лома у девијаторској равни као и да облик функције пластичног потенцијала адекватно описује развој пластичних деформација. У наставку је дат преглед релација из литературе које узимају у обзир утицај Lode-овог угла у изразима

за површ тешења и површ пластичног потенцијала, које су коришћене за унапређење МСС модела.

Mohr-Coulomb-ов хексагон у девијаторској равни адекватније описује површ течења него Drucker-Prager-ов круг. (Potts & Zdravković, 1999) и (Potts & Zdravković, 2000) су изменили МСС модел тако да површ течења у девијаторској равни буде Mohr-Coulomb-ов хексагон:

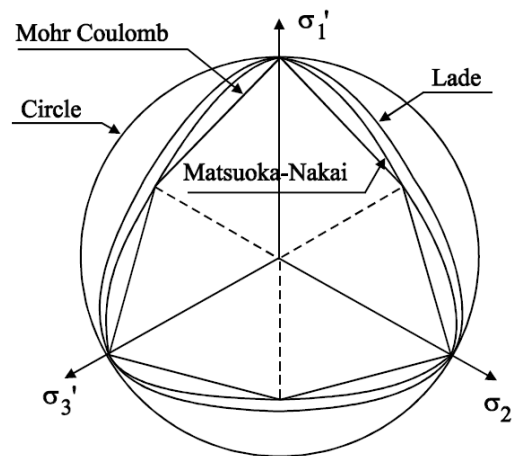
$$F(p', J, \theta, p'_0) = \left(\frac{J}{g(\theta)} \right)^2 + p'(p' - p'_0) = 0 \quad 3-52$$

У претходном изразу J и $g(\theta)$ су дати изразима:

$$J = \frac{q}{\sqrt{3}} \quad 3-53$$

$$g(\theta) = \frac{\sin \phi'_{cv}}{\cos \theta + \frac{\sin \theta \sin \phi'_{cv}}{\sqrt{3}}} \quad 3-54$$

Иако је Mohr-Coulomb-ов хексагон боље решење за критеријум лома у девијаторској равни него круг, и даље не описује понашање тла при лому у складу са експерименталним подацима. Због тога се могу користити други критеријуми лома као што су критеријуми (Matsuoka & Nakai, 1974) и (Lade & Duncan, 1975). Облик критеријума лома у девијаторској равни приказан је на **Слика 3-6**.



Слика 3-6 (Matsuoka & Nakai, 1974) и (Lade & Duncan, 1975) критеријуми лома – преузето из (Georgiadis, et al., 2004)

Ова два критеријума могу да се напишу у истом облику као (3-52). Функција $g(\theta)$ за (Matsuoka & Nakai, 1974) критеријум је дата изразом:

$$g(\theta) = \sqrt{J_{2\eta}^f} \quad 3-55$$

У претходном изразу $J_{2\eta}^f$ се рачуна за задату вредност Lode-овог угла решавањем једначине:

$$(C_{MN} - 3)J_{2\eta}^f + \frac{2}{\sqrt{27}}C_{MN} \sin 3\theta (J_{2\eta}^f)^{\frac{3}{2}} - (C_{MN} - 9) = 0 \quad 3-56$$

И C_{MN} је дато изразом:

$$C_{MN} = \frac{9 - 3M_J^2}{\frac{2\sqrt{3}}{9}M_J^3 - M_J^2 + 1} \quad 3-57$$

У претходном изразу M_J је нагиб линије критичног стања у $J - p'$ равни ($M = M_J/\sqrt{3}$ за триаксијалну компресију).

Функција $g(\theta)$ за критеријум (Lade & Duncan, 1975) је исто дата изразом (3-55), док се $J_{2\eta}^f$ рачуна за задату вредност Lode-овог угла θ и средњег ефективног напона p' решавањем једначине:

$$J_{2\eta}^f + \frac{2}{\sqrt{27}} \sin 3\theta (J_{2\eta}^f)^{\frac{3}{2}} - C_L = 0 \quad 3-58$$

И C_L је дато изразом:

$$C_L = \frac{\frac{\eta_L \left(\frac{p_a}{3p'}\right)^m}{27}}{1 + \frac{\eta_L \left(\frac{p_a}{3p'}\right)^m}{27}} \quad 3-59$$

У претходном изразу η_L и m су материјалне константе, а p_a је атмосферски притисак.

Функција $g(\theta)$ се може написати и у облику који је предложио (van Eekelen, 1980):

$$g(\theta) = X(1 + Y \cdot \sin 3\theta)^Z \quad 3-60$$

У претходном изразу X , Y и Z ($Z < 0$) су константе које треба тако изабрати да површ течења у девијаторској равни буде конвексна, као и да се што приближније описује површ течења добијена из експеримената.

(Bardet, 1990) је дао преглед неколико различитих облика критеријума лома у функцији Lode-овог угла као и услове које та зависност треба да задовољи како би површ течења у девијаторској равни била конвексна, диференцијабилна и адекватно предвиђала облик површи који одговара експерименталним подацима. Зависности у (Bardet, 1990) су дате у облику:

$$M(\theta) = M_c \cdot \rho(\theta, r_{CE}) \quad 3-61$$

У претходном изразу r_{CE} зависи од ϕ'_{cv} према изразу:

$$r_{CE} = \frac{3 - \sin \phi'_{cv}}{3 + \sin \phi'_{cv}} \quad 3-62$$

Претходни израз у ствари представља однос нагиба линије критичног стања у $p' - q$ равни у триаксијалној екстензији M_E и нагиба линије критичног стања у $p' - q$ равни у триаксијалној компресији M_C уколико су углови смичуће отпорности у компресији и екстензији исти и једнаки углу смичуће отпорности тла при константној запремини $\phi_C = \phi_E = \phi_{cv}$.

У (Bardet, 1990) је приказано неколико зависности за $M(\theta)$ као и њихова примена на МСС модел. У наставку су приказане три које се могу применити за моделе критичног стања. Једна од њих је дата у облику:

$$\rho(\theta) = \frac{2r_{EC}}{1 + r_{EC} + (r_{EC} - 1) \cdot \sin(-3\theta)} \quad 3-63$$

Друга је генерализација претходног израза:

$$\rho(\theta) = \frac{1}{2^n} \left(1 + (r_{EC})^{\frac{1}{n}} + \left(1 - (r_{EC})^{\frac{1}{n}} \right) \cdot \sin(-3\theta) \right)^n \quad 3-64$$

У претходном изразу n ($n < 0$) је константа. (Bardet, 1990) такође предлаже зависност која је заснована на (Matsuoka & Nakai, 1974) и (Lade & Duncan, 1975) условима лома (LMN критеријум):

$$\rho(\theta) = \frac{\sqrt{3}}{2} \frac{r_{CE}}{\sqrt{(r_{CE})^2 - r_{CE} + 1}} \frac{1}{\cos(\epsilon)} \quad 3-65$$

Где је

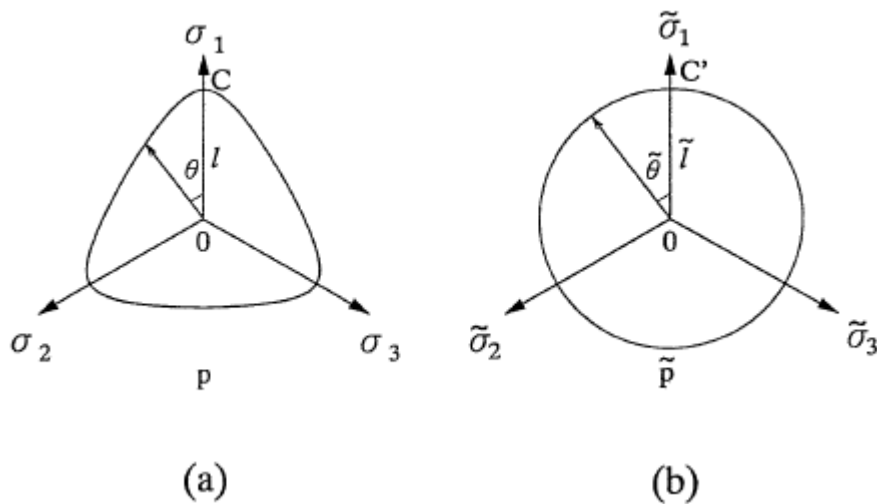
$$\epsilon = \begin{cases} \frac{1}{6} \cdot \arccos \left(-1 + \frac{27}{2} \frac{(r_{CE})^2 (1 - r_{CE})^2}{((r_{CE})^2 - r_{CE} + 1)^3} \cdot (\sin(-3\theta))^2 \right), & \theta \geq 0 \\ \frac{\pi}{3} - \frac{1}{6} \cdot \arccos \left(-1 + \frac{27}{2} \frac{(r_{CE})^2 (1 - r_{CE})^2}{((r_{CE})^2 - r_{CE} + 1)^3} \cdot (\sin(-3\theta))^2 \right), & \theta < 0 \end{cases} \quad 3-66$$

Критеријум лома који је развио (Matsuoka & Nakai, 1974), (Matsuoka, 1976), (Matsuoka, et al., 1999) и (Matsuoka & Sun, 2006) се назива још и SMP (Spatially Mobilized Plane) критеријум. Облик овог критеријума приказан је на **Слика 3-6** и у простору главних напона представља глатку конвексну криву која описује Mohr-Coulomb-ов услов лома. SMP критеријум се боље слаже са експерименталним подацима и веома је коришћен за унапређивање модела критичног стања као и других конститутивних модела. У (Matsuoka, et al., 1999) је предложен поступак трансформације главних напона тако да SMP критеријум у простору трансформисаних напона има облик купе а у девијаторској равни облик круга (**Слика 3-7**). Проширење МСС модела са SMP критеријумом може се наћи у (Matsuoka, et al., 1999) и (Matsuoka & Sun, 2006). (Matsuoka, et al., 2005) су развили једноставан конститутиван модел за глине и пескове у оквиру концепта критичног стања који користи SMP критеријум у простору трансформисаних главних напона. Овај модел је даље развијан са новим параметром ојачања независним од путање напона

како би се развио заједнички конститутивни модел за пескове и глине у (Yao, et al., 2008). (Yao, et al., 2009) су развили UH (Unified Hardening) модел за преконсолидоване глине који је заснован на MCC моделу и користи SMP критеријум у простору трансформисаних главних напона. (Yao & Wang, 2014) су дали општи поступак како се SMP, (Lade & Duncan, 1975) и други критеријуми лома могу применити на конститутивне моделе развијене у оквиру концепта критичног стања применом трансформације главних напона.

(Sheng, et al., 2000) су за проширење MCC модела на генерални напонски простор користили зависност $M(\theta)$ дату изразом:

$$M(\theta) = M_c \cdot \left(\frac{2(r_{CE})^4}{1 + (r_{CE})^4 - (1 - (r_{CE})^4) \cdot \sin(-3\theta)} \right)^{\frac{1}{4}} \quad 3-67$$



Слика 3-7 SPM критеријум у девијаторској равни: (a) простор главних напона; (b) простор трансформисаних напона – преузето из (Matsuoka, et al., 1999)

Creep-SCLAY1S модел (Sivasithamparam, et al., 2015) користи исту зависност дату (3-67). (Zdravković, et al., 2020) су за унапређење MCC модела користили зависност дату изразом (3-60). За вредност константе Z усвојили су вредност $Z = -0.1$, док су константе X и Y одређене из услова да се за услове триаксијалне компресије, односно екстензије добију одговарајући нагиби линије критичног стања у $p' - J$ равни.

Дакле, адекватним избором $M(\theta)$ у једначини површи течења и површи пластичног потенцијала може се постићи да се у конститутивни модел имплементира било који облик ових површи у девијаторској равни. Избор облика пластичног потенцијала у девијаторској равни утиче на вредност Lode-овог угла при лому при раванском стању деформације и због тога површ пластичног потенцијала треба изабрати тако да конститутивни модел предвиђа реалистично понашање тла при лому (Potts & Zdravković, 1999). Често је потребно да се усвоје различити облици за површ течења и површ пластичног потенцијала, односно да се користи неасоцијативни закон течења и (Potts & Zdravković, 1999) дају детаљну дискусију како избор функције пластичног

потенцијала утиче на предвиђање понашања тла у нумеричким симулацијама, поготово код раванског стања деформација.

(Potts & Zdravković, 1999) и (Potts & Zdravković, 2000) су користили неасоцијативан закон течења за унапређење МСС модела. Функција течења је дата изразом (3-52), док је функција пластичног потенцијала дата изразом:

$$P(p', J, p'_0) = \left(\frac{J}{g(\theta_c)} \right)^2 + p'(p' - p'_0) = 0 \quad 3-68$$

У претходном изразу функција $g(\theta_c)$ се рачуна за вредност Lode-овог угла θ_c у напонској тачки у којој су потребне вредности градијената пластичног потенцијала. Односно, променљива θ се мења са константом θ_c . У овако унапређеном МСС моделу, површ течења у девијаторској равни има облик Mohr-Coulomb-овог хексагона, док пластични потенцијал има облик круга. (Zdravković, et al., 2020) у унапређеном МСС моделу за боље предвиђање понашања преконсолидованих глина такође користе неасоцијативан закон течења, док су функција течења и површ пластичног потенцијала дати у следећем облику:

$$F(p', J, \theta, p'_0) = 0 \quad 3-69$$

3-70

$$P(p', J, p'_0) = 0$$

НАSP модел је проширен на генерални напонски простор тако што је у изразима (3-49) и (3-50) функција $M_F(\theta)$ изражена у облику:

$$M_F(\theta) = M(\theta) = X(1 + Y \cdot \sin 3\theta)^Z \quad 3-71$$

У изразу за пластични потенцијал (3-51) функција $M_P(\theta)$ је у облику:

$$M_P(\theta) = M_P(\theta_c) = M(\theta_c) = X(1 + Y \cdot \sin 3\theta_c)^Z \quad 3-72$$

НАSP-E модел користи неасоцијативан закон течења. Облик површи течења у девијаторској равни је заобљена конвексна површ, а површ пластичног потенцијала има облик круга. У изразима (3-71) и (3-72) за константу Z се усваја вредност $Z = -0.229$, што одговара истој вредности која је усвојена код OC_CLAY модела.

Вредности константи X и Y се одређује из следећих услова:

$$M(\theta = -30^\circ) = X(1 - Y)^Z = M_c \quad 3-73$$

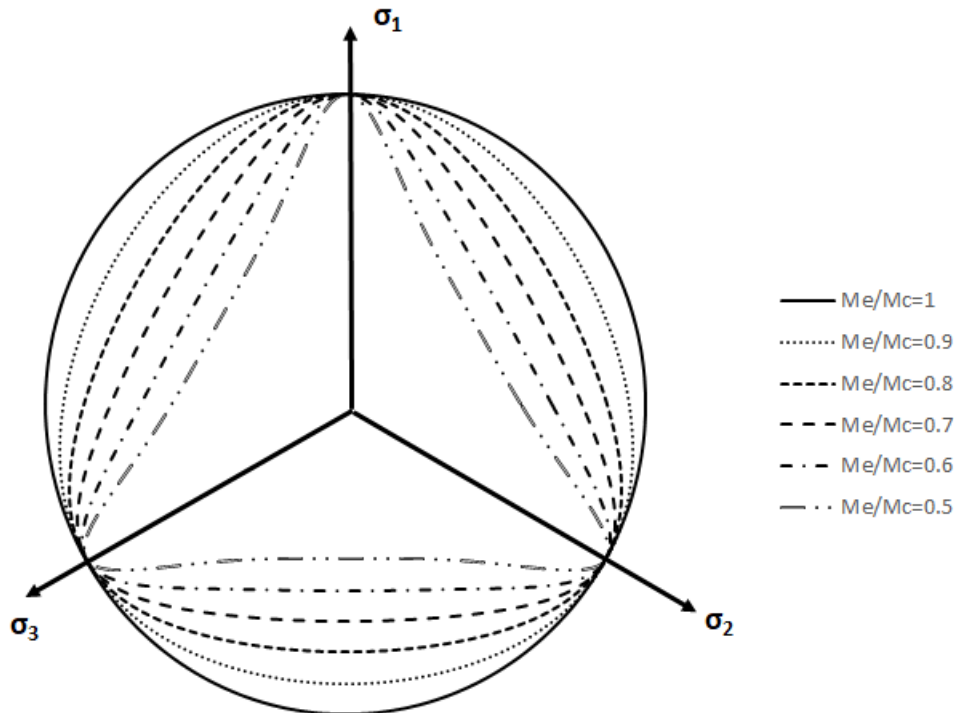
3-74

$$M(\theta = 30^\circ) = X(1 + Y)^Z = M_e$$

Решавањем система једначина (3-73) и (3-74) добијају се изрази за X и Y :

$$X = \left(\frac{M_c^{\frac{1}{Z}} + M_e^{\frac{1}{Z}}}{2} \right)^Z \quad 3-75$$

$$Y = \frac{1 - \left(\frac{M_c}{M_e}\right)^{\frac{1}{Z}}}{1 + \left(\frac{M_c}{M_e}\right)^{\frac{1}{Z}}}$$

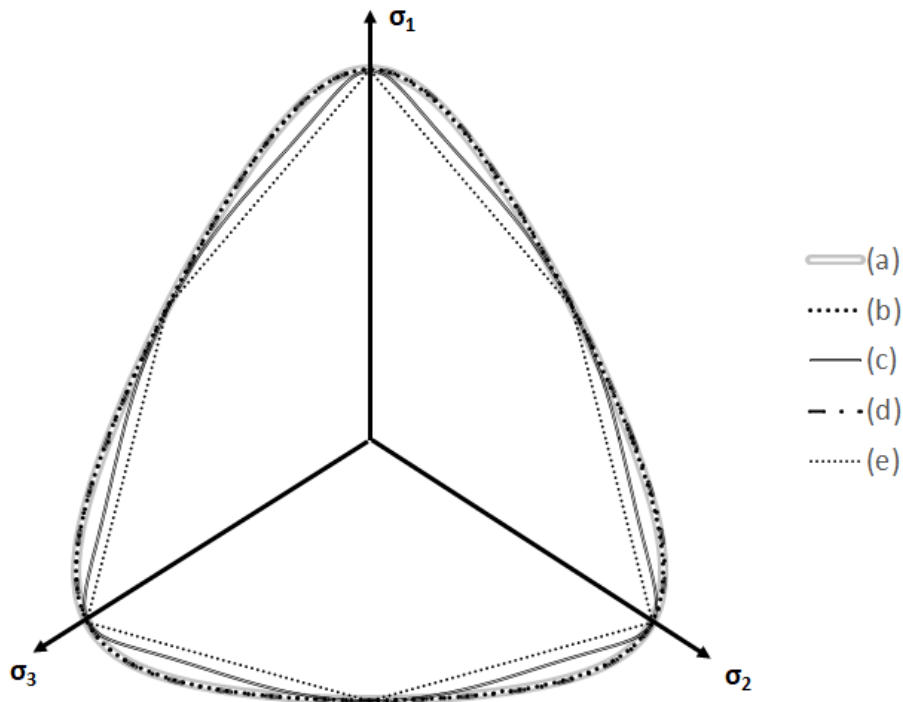


Слика 3-8 Облик критеријума лома у девијаторској равни за различите односе M_e/M_c за усвојену зависност $M(\theta)$ и $Z = -0.229$

Зависност (3-71) омогућава да се могу описати површи течења код којих углови смичуће отпорности у триаксијалној компресији и триаксијалној екстензији нису исти ($\phi'_C \neq \phi'_E$), док су зависности дате у (Bardet, 1990) и (Sheng, et al., 2000) ограничене на ситуације када је ($\phi'_C = \phi'_E$).

На **Слика 3-8** приказано је како изгледа критеријум лома у девијаторској равни за усвојену зависност $M(\theta)$ према изразу (3-71) и вредност константе $Z = -0.229$ за различите вредности односа M_e/M_c . Може се уочити да за односе M_e/M_c мање од 0.6 критеријум лома прелази у конкавни облик.

На **Слика 3-9** приказано је како изгледају критеријуми лома у девијаторској равни када је $\phi_C = \phi_E = \phi_{cv} = 30^\circ$ за пет различитих зависности $M(\theta)$. Може се уочити да све зависности описују Mohr-Coulomb-ов хексагон и конвексне су површи. Усвојена зависност се скоро поклапа са LMN критеријумом и критеријумом који је користио (Sheng, et al., 2000). За критеријум који су користили (Zdravković, et al., 2020) може се закључити да је за константу Z у изразу (3-71) изабрана таква вредност да се површ течења што је могуће више поклапа са Mohr-Coulomb-овим критеријум и притом је конвексна и диференцијабилна у свим тачкама.



Слика 3-9 Облик критеријума лоба у девијаторској равни за $\phi_C = \phi_E = \phi_{cv} = 30^\circ$ за: (a) усвојена зависност $M(\theta)$ и $Z = -0.229$, (b) LMN критеријум - израз (3-65), (c) (Zdravković, et al., 2020) са $Z = -0.1$, (d) (Sheng, et al., 2000) израз (3-67), (e) Mohr-Coulomb-ов хексагон

3.3.2 Проширивање релација HASP модела узимањем у обзир промене модула смицања са променом смичућих деформација

Као што је описано у потпоглављу 2.5, конститутивни модел треба да на одговарајући начин предвиђа величину крутости тла у подручју веома малих и малих деформација. Основна формулација HASP модела не узима у обзир промену крутости тла у зависности од величине деформација у тлу и због тога је било потребно проширити релације модела како би се на адекватан начин предвиђале деформације у тлу и понашање геотехничких конструкција. Релације модела су проширене тако што је узета у обзир деградација модула смицања са порастом смичућих деформација у складу са S-кривом приказаном на **Слика 2-10**. Проширивање модела се може извести модификацијом основних релација тако да се предвиђа адекватна величина крутости тла или да се постојеће релације модела комбинују са релацијама неког другог конститутивног модела (small strain overlay model) који адекватно предвиђа крутост тла у подручју веома малих и малих деформација. Улога тог другог конститутивног модела би била да предвиђа величину модула смицања у функцији величине смичућих деформација коју после користи основни модел за предвиђање понашања тла. У наставку је дат кратак преглед из литературе на које се све начине у конститутивним моделима узима у обзир деградација крутости тла са порастом деформација. На крају овог потпоглавља су приказане релације које су одабране за проширивање HASP модела.

У (Potts & Zdravković, 1999) дат је опис два конститутивна модела који су били међу првим формулисаним моделима за одређивање крутости тла при малим

деформацијама: модел (Jardine, et al., 1986) и модел (Puzrin & Burland, 1998). Модел (Jardine, et al., 1986) је развијен да опише понашање тла у подручју малих деформација и због тога се комбинује са еласто-пластичним конститутивним моделима када треба да опише понашање тла при великим деформацијама. Овај модел даје релације деградације модула смицања са порастом смичућих деформација и деградације запреминског модула са порастом запреминских деформација, тако да су вредности модула смицања и запреминског модула међусобно независне. Модел (Puzrin & Burland, 1998) користи концепт више кинематичких површи смештених унутар класичне површи течења у оквиру којих је понашање тла линеарно еластично или нелинеарно еластично. Овај модел је унапређење (Jardine, et al., 1986) модела. Може адекватно да предвиди промену крутости тла са променом путање напона и вредности модула смицања и запреминског модула зависе међусобно.

Детаљан преглед 54 конститутивна модела који описују понашање тла у подручју веома малих и малих деформација дали су (Castellon & Ledesma, 2022a). Такође, дали су опис свих битних елемената конститутивног модела који су потребни за описивање понашања тла у подручју веома малих и малих деформација, као и при цикличном оптерећењу, односно да ли конститутивни модел има могућност да предвиди хистерезис који се јавља у једном циклусу оптерећења и растерећења.

HSS модел (Benz, 2007) је често заступљен у пракси и комерцијалним рачунарским програмима. Овај конститутивни модел користи SSO модел за који су потребна само два параметра за описивање S-криве: иницијална вредност модула смицања при веома малим деформацијама G_0 и величина смичуће деформације $\gamma_{0.7}$ при којој је секантни модул смицања редукован на 70% G_0 . SSO модел представља такозване “overlay” моделе. Релације таквих модела се придружују релацијама оних модела који се унапређују: HSS модел је настао прикључивањем SSO модела релацијама HS модела (Schanz, et al., 1999). Крутост тла у подручју малих деформација као и понашање тла при промени путање напона или деформација не зависе од релација HS модела, већ се крутост тла рачуна према релацијама SSO модела коју после HS модел користи даље за предвиђање понашања тла. У OC_CLAY моделу је SSO модел прикључен релацијама HASP модела у циљу бољег предвиђања понашања у подручју веома малих и малих деформација као и да би се омогућило да HASP модел може да предвиђа понашање преконсолидованих глина при цикличном оптерећењу.

(Castellon, 2019) и (Castellon & Ledesma, 2022b) су развили EPHYSS (Elastoplastic Hysteretic Small Strain) модел као унапређење HSS модела. EPHYSS модел представља комбинацију HS модела и HQN (Hysteretic Quasi-Hypoelastic) модела као “overlay” модела. У односу на HSS модел, EPHYSS модел даје боља предвиђања понашања тла у подручју веома малих и малих деформација и услед цикличног оптерећења, поготово у ситуацијама када долази до делимичне промене правца оптерећивања или до промене правца оптерећивања при достигнутим великим деформацијама. Такође, EPHYSS модел узима у обзир анизотропију индуковану деформацијама (strain-induced anisotropy) и могућност усвајања различитих облика S-кривих (HSS модел користи само један облик S-криве дат изразом (3-77)). Више о предностима EPHYSS модела у односу на HSS модел може се наћи у (Castellon & Ledesma, 2022b).

(Bolton, et al., 1994) и (Osman & Bolton, 2005) су развили SDMCC (Strain Dependent Modified Cam Clay) модел. SDMCC модел је развијен да опише понашање преконсолидо-

ваних глина у подручју веома малих и малих деформација и представља модификацију МСС модела. Унутар површи течења SDMCC модела модул смицања се мења у функцији девијаторских деформација у складу са S-кривом, док се запремински модул срачунава као у МСС моделу. За напонска стања на површи течења, SDMCC се понаша као МСС модел. SDMCC може да региструје када је дошло до промене путање напона тако да овај модел може да предвиђа хистерезисно понашање преконсолидованих глина.

(Hsieh, et al., 2017) су развили SC1SS модел тако што су проширили релације S-CLAY1 модела (Wheeler, et al., 2003) узимајући у обзир утицај Lode-овог угла и крутости тла у подручју малих и веома малих деформација. У SC1SS моделу модул смицања се рачуна у зависности од величине девијаторских деформација у складу са усвојеном S кривом. Након достигнуте одређене девијаторске деформације, модул смицања се одређује као у S-CLAY1. Модул смицања у овом моделу може да монотонно расте или опада у зависности од нивоа девијаторских деформација и не може да се детектује да ли је дошло до промене путање напона тако да овај модел не може да репродукује хистерезис и не може се користити за циклична оптерећења.

Конститутивни модели који користе концепт више кинематичких површи у напонском или деформацијском простору које су смештене једне унутар других се такође користе за описивање понашања тла у подручју веома малих и малих деформација као и при цикличним оптерећима. Модел (Puzrin & Burland, 1998) је већ описан на почетку овог потпоглавља, а на истом концепту засновани су модели (Prevost, 1977) и (Simpson, et al., 1979). Сва три конститутивна модела су у стању да предвиђају хистерезисно понашање тла и могу се користити за описивање понашања тла при цикличном оптерећењу.

У потпоглављу 2.6 описани су тзв. “bubble” модели (Al-Tabba & Wood, 1989) и 3-SKN (Stallebrass, 1990) који могу да се користе за описивање промене крутости тла са променом деформација и понашања тла при цикличном оптерећењу. MIT-E3 (Whittle, 1987) и MIT-S1 (Pestana & Whittle, 1999) могу да опишу понашање тла у домену малих деформација и приликом цикличног оптерећења. MIT-S1 модел је настао генерализацијом MIT-E3 модела за глине и пескове. Модел који су развили (Schadlich & Schweiger, 2012) у оквиру концепта мултиламинатних модела може се користити за описивање цикличног понашања пескова и глина, као и за предвиђање понашања тла у подручју веома малих и малих деформација.

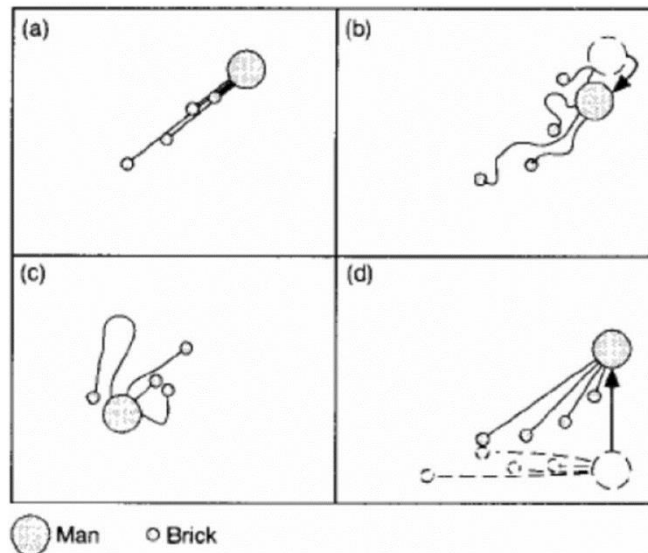
(Taborda & Zdravković, 2012) и (Taborda, et al., 2016) развили су циклични нелинеарно еластични конститутивни модел под називом ICG3S (Imperial College Generalized Small Strain Stiffness) модел. У овом моделу крутост тла зависи од нивоа деформација: модул смицања зависи од нивоа девијаторских деформација а запремински модул зависи од нивоа запреминских деформација. Као последица тога, модул смицања и запремински модул су међусобно независни, али је у моделу могуће прописати да се користи константна вредност Poisson-овог коефицијента, чиме запремински модул зависи од вредности модула смицања. ICG3S модел може да опише понашање тла при цикличном оптерећењу и као “overlay” модел може да се комбинује са другим еласто-пластичним моделима како би их побољшао. Примена ICG3S модела у комбинацији са проширеним МСС моделом може се наћи у (Zdravković, et al., 2020) и (Zhou, 2022).

У описивању понашања тла у подручју веома малих и малих деформација и понашања тла при цикличном оптерећењу веома успешно су се показали такозвани “Brick”

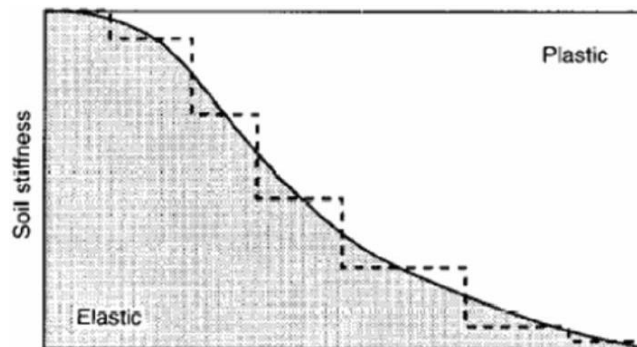
(цигла) модели. Ови модели су развијени и формулисани у простору деформација уместо у напонском простору. Могу да се користе као независни модели или да се као “overlay” модели комбинују са другим конститутивним моделима. (Simpson, 1992) је први поставио теоријске основе на којима су засновани “Brick” модели. Као што је већ описано у потпоглављу 2.6, “Brick” модели описују понашање тла у подручју веома малих и малих деформација користећи аналогију особе која вуче за собом сет цигли помоћу неистегљивих канапа. Дужине свих канапа су различите. Особа се креће у простору деформација и путања особе представља нанете деформације. Кретање цигли представља пластичне деформације. Уколико се услед кретања особе цигле не померају, односно канапи нису затегнути јер је растојање од особе до цигли мање него што је дужина канапа, деформације су еластичне и крутост тла има максималну вредност. Ако особа настави да се креће у истом правцу, затегнуће се најкраћи канап и цигла коју повезује тај канап ће почети да се креће. Померање те цигле представља пластичне деформације и крутост тла се смањује за део максималне крутости који одговара тој цигли. Како се кретање особе наставља у истом правцу, један по један канап ће постајати затегнут и постепено ће све цигле почети да се крећу у правцу кретања особе. Како се која цигла покреће тако пластичне деформације расту и крутост тла се смањује. Када су сви канапи затегнути и све цигле се крећу у правцу кретања особе, померања особе и цигли су иста и све деформације су пластичне. Уколико особа промени правац кретања за 180° (крене натраг), сви затегнути канапи постаће опуштени и цигле престају да се крећу. Понашање тла је поново еластично и крутост тла има максималну вредност. Приликом промене смера кретања за 180° и уколико су сви канапи били затегнути, растојање између особе и сваке цигле понаособ је једнако дужини канапа. Кретањем натраг по истој истој путањи особа треба да пређе пут једнак двострукој дужини канапа да би цигла кренула поново да се креће. Тако да при кретању натраг по истој путањи, особа треба да пређе два пута дужи пут да би цигле почеле да се крећу и крутост тла да се смањује. Такође, приликом промене правца кретања за било који други угао, неки канапи ће се опустити а неки ће остати и даље затегнути, тако да ће се повратити део крутости тла. Илустрација претходно описаног кретања особе и цигала приказана је на **Слика 3-10**.

“Brick” модели према претходно описаној аналогији могу да предвиде понашање тла у подручју веома малих и малих деформација као што је максимална крутост на почетку деформисања која се постепено редукује. Такође може да опише понашање тла при цикличном оперећењу као што је да тло поврати максималну крутост при промени путање напона или деформација за 180° или да поврати део крутости при промени путање напона или деформација за било који други угао.

У “Brick” моделима број, положај цигли и информација да ли је канап затегнути или не представљају параметре стања. Промена путање напона или деформација се открива уколико је канап који је на почетку инкремента био затегнут постао опуштен на крају инкремента. Положај сваке цигле на почетку инкремента је познат и ажурира се на крају инкремента у зависности од прираштаја деформација. У зависности од броја коришћених цигли и њихових положаја зависи колико ће модел бити осетљив да открије да ли је дошло до промене путање напона или деформација. Пошто се свакој цигли додељује део крутости тла која се одузима од максималне крутости уколико се та цигла креће, крива која описује промену крутости је степенаста (**Слика 3-11**):



Слика 3-10 Аналогија особе која вуче за собом сет цигли помоћу неистегљивих канана – преузето из (Simpson, 1992)



Слика 3-11 Степенаста деградација крутости тла у “Brick” моделима – преузето из (Simpson, 1992)

Што се већи број цигли користи, степенаста крива на претходној слици ће тачније описивати S-криву, али са повећањем броја цигли потребно је више информација чувати о положајима цигли.

Први “Brick” модел (Simpson, 1992) је формулисан у простору деформација који одговара раванском стању деформација. Положај једне цигле у таквом простору дефинисан је величином запреминске и смичуће деформације. Коришћено је 10 цигли да би се описала S-крива. Овај модел може да узме у обзир степен преконсолидације глине и даје добра предвиђања за путање напона у недренираним условима, док за путање у дренираним условима не даје адекватна предвиђања понашања глине и не може да репродукује критично стање.

(Lansivaara & Nordal, 1998) су развили еласто-пластични конститутивни модел за преконсолидоване глине комбиновањем “Brick” модела (Simpson, 1992) са моделом који су развили (Nordal, et al., 1990). Према резултатима симулације основних лабораторијских опита које су (Lansivaara & Nordal, 1998) објавили, овај модел може у одређеној мери да предвиди понашање тла у подручју малих и веома малих деформација, као и при цикличном оптерећењу.

(Lehane & Simpson, 2000) су проширили “Brick” модел (Simpson, 1992) за тродимензионално стање напона. Користили су 10 цигли и положај једне цигле је изражен преко шест деформацијских координата: запреминска деформација и пет смичућих деформација. Ова нова верзија “Brick” модела доноси још унапређења у односу на старију верзију и може успешно да репродукује битне аспекте понашања тврдих преконсолидованих глина у условима који владају у триаксијалном апарату и даје реалистична предвиђања слегања плитких темеља.

(Clarke, 2009) и (Clarke & Hird, 2012) су развили SRD (strain rate-dependent) BRICK модел који је заснован на “Brick” моделу (Simpson, 1992) за предвиђање вискозног понашања меких глина укључујући пузање, релаксацију напона и ефекте брзине деформисања на крутост тла. Користили су 18 цигли за апроксимацију S-криве. На основу резултата симулације едометарског и CU триаксијалног опита у којима су симулирани утицаји пузања и коришћене различите брзине доношења деформација (Clarke & Hird, 2012) оцењују да овај модел успешно предвиђа понашање тла у подручју веома малих и малих деформација.

(Ellison, et al., 2012) су додатно проширили “Brick” модел (Lehane & Simpson, 2000) и (Simpson, 1992) тако да овај модел може да предвиди критично стање тла и да узме у обзир анизотропију крутости тла. Овај модел даје много боља предвиђања понашања глине него основни “Brick” модел што су (Ellison, et al., 2012) показали поређењем резултата нумеричких симулација са резултатима разних лабораторијских опита извршених на лондонској глини.

(Jovičić, et al., 2023) су развили SA_BRICK модел који узима у обзир анизотропију и утицај структуре на механичко понашање природних тврдих глина. Могућност да SA_BRICK модел може да третира анизотропију је верификована кроз симулацију триаксијалних опита на лондонској глини.

(Cudny & Truty, 2020) су формулисали HS-Brick модел као побољшање и превазилажење неких недостака HSS модела при предвиђању цикличног понашања тла. HS-Brick користи “Brick” модел као “overlay” модел уместо (Benz, 2007) SSO модела. HS-Brick модел задржава исти облик S-криве као у HSS моделу. Израз за S-криву коју користи HSS и HS-Brick модел је:

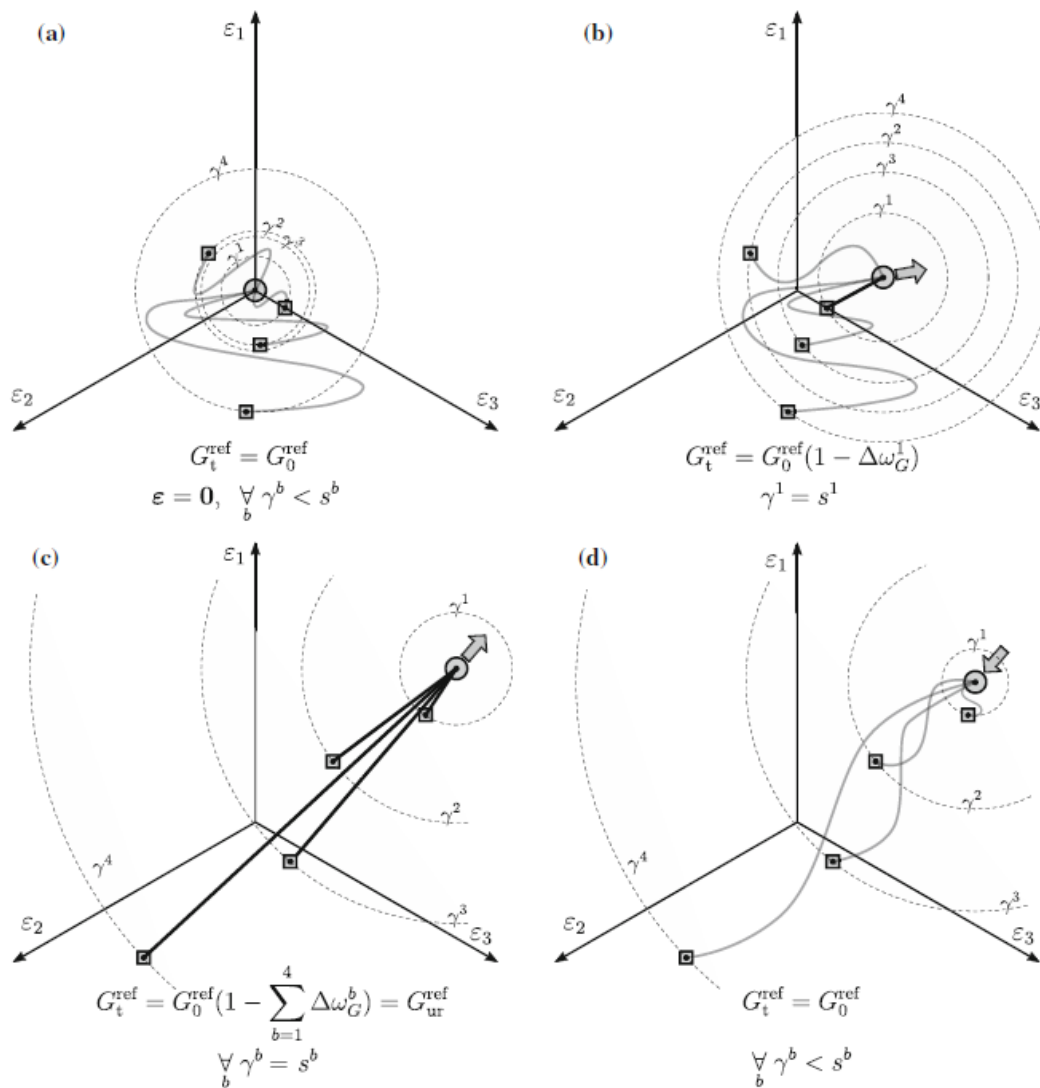
$$G_t = \frac{G_0}{\left[1 + 0.385 \cdot \left(\frac{\gamma_{hist}}{\gamma_{0.7}}\right)\right]^2} \quad 3-77$$

У претходном изразу, G_t је тангентна вредност модула смицања, G_0 је иницијална вредност модула смицања при веома малим деформацијама, γ_{hist} је смичућа деформација која узима у обзир претходну историју оптерећивања и $\gamma_{0.7}$ је смичућа деформација при којој је секантни модул смицања редукован на 70% G_0 .

“Brick” компонента HS-Brick модела је формулисана у шестодимензионом простору деформација ($\varepsilon_x, \varepsilon_y, \varepsilon_z, \gamma_{xy}, \gamma_{yz}, \gamma_{zx}$). Растојање између особе и b -те цигле се мери преко смичуће деформације γ^b . На тај начин, се процес кретања особе и цигли пројектује на девијаторску раван у простору деформација. На **Слика 3-12** је приказана илустрација кретања особе и цигала у девијаторској равни у простору деформација у HS-Brick

моделу. S-крива у HS-Brick моделу је степенаста за разлику од HS модела који користи континуалну криву (Слика 3-13).

Као што се може видети на Слика 3-13 вредност референтног тангентног модула смицања G_t^{ref} се креће у границама од референтног максималног модула смицања за веома мале деформације G_0^{ref} и референтног модула смицања за велике деформације G_{ur}^{ref} . Сва три модула смицања су дата за референтни средњи ефективни напон од $p'_{ref} = 100 \text{ kPa}$. У зависности од величине средњег ефективног напона одређује се вредност тангентног модула смицања G_t тако да овај модул онда зависи и од деформација и нивоа напона.



Слика 3-12 Илустрација особе која вуче цигле у HS-Brick моделу: (a) почетни положај цигли, (b) почетак деградације крутости када је један канап затегнут, (c) максимална деградација крутости јер су сви канапи затегнути, (d) промена правца кретања, сви канапи су опуштени и крутост поново има максималну вредност – преузето из (Cidny & Truty, 2020)

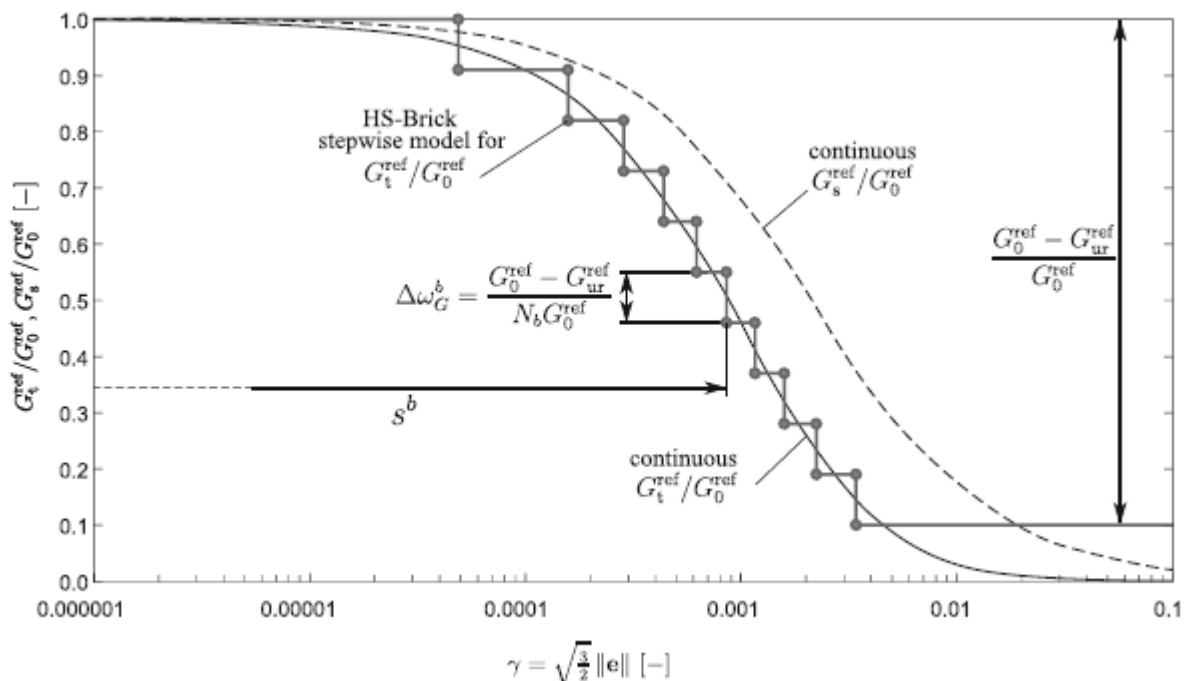
На почетку оптерећивања, када су деформације у посматраној тачки тла једнаке нули (особа је у координатном почетку), цигле могу заузимати неки унапред задат положај

(ван координатног почетка) који може бити последица претходне историје оптерећивања или неких других процеса који су се догодили у прошлости. Или се и цигле могу налазити у координатном почетку на почетку прорачуна. Уколико је N_b број цигли који се користи, b број посматране цигле ($b = 1, \dots, N_b$) онда $\boldsymbol{\varepsilon}_n^b$ представља вектор који садржи координате (деформације) положаја b -те цигле на почетку инкремента (или субинкремента) оптерећења. На крају инкремента (или субинкремента), координате b -те цигле су $\boldsymbol{\varepsilon}_{n+1}^b$. Координате (укупне деформације) особе на почетку инкремента су $\boldsymbol{\varepsilon}_n$, а на крају инкремента су $\boldsymbol{\varepsilon}_{n+1}$. Промена координата особе услед кретања је вектор прираштаја деформација $\Delta\boldsymbol{\varepsilon}_{n+1}$:

$$\boldsymbol{\varepsilon}_{n+1} = \boldsymbol{\varepsilon}_n + \Delta\boldsymbol{\varepsilon}_{n+1} \quad 3-78$$

На почетку кретања особе из координатног почетка све цигле су у координатном почетку и сви канапи су опуштени. Како се особа удаљава од координатног почетка, цигла са најкраћим канапом ће кренути да се креће када се тај канап затегне. У том тренутку, крутост тла ће се смањити за проценат разлике између G_0^{ref} и G_{ur}^{ref} који одговара тој цигли $\Delta\omega_G^b$. У HS-Brick моделу је усвојено да свакој цигли одговара исти проценат деградације смичуће крутости (Слика 3-13), тако да се $\Delta\omega_G^b$ рачуна преко израза:

$$\Delta\omega_G^b = \frac{G_0^{ref} - G_{ur}^{ref}}{N_b G_0^{ref}} \quad 3-79$$



Слика 3-13 Степенаста S -крива у HS-Brick моделу за $N_b = 10$ цигли – преузето из (Cidny & Truty, 2020)

Уколико је N_t број канапа који су затегнути, онда је укупна редукција тангентног модула смицања:

$$\omega_G = N_t \Delta\omega_G^b \quad 3-80$$

А вредност референтног тангентног модула смицања G_t^{ref} износи:

$$G_t^{ref} = (1 - \omega_G) G_0^{ref} \quad 3-81$$

Да би могло да се одреди да ли је канап између особе и цигле затегнут, потребно је одредити дужине канапа (које су константне јер су канапи неистегљиви) и растојање између особе и цигли. Дужина канапа који повезује особу и b -ту циглу означена је са s^b . Ако је вредност референтног тангентног модула смицања која одговара b -тој по реду цигли чији канап постаје затегнут $G_{t,b}^{ref}$, онда се дужина канапа s^b одређује из услова да је једнака смичућој деформацији γ^b којој на S-криви одговара вредност од $G_{t,b}^{ref}$:

$$G_{t,b}^{ref} = G_0^{ref} (1 - (b - 0.5) \cdot \Delta\omega_G^b) = \frac{G_0^{ref}}{\left[1 + 0.385 \cdot \left(\frac{s^b}{\gamma_{0.7}}\right)\right]^2} \quad 3-82$$

Из претходног израза, дужина канапа s^b је:

$$s^b = \frac{\gamma_{0.7}}{0.385} \cdot \left(\sqrt{\frac{G_0^{ref}}{G_0^{ref} (1 - (b - 0.5) \cdot \Delta\omega_G^b)}} - 1 \right) \quad 3-83$$

Ако се са γ_{n+1}^b обележи растојање између особе и b -те цигле, онда је канап опуштен уколико је $\gamma_{n+1}^b \leq s^b$, а затегнут ако је $\gamma_{n+1}^b > s^b$. Растојање између особе и b -те цигле се одређује тако што се прво срачуна вектор растојања:

$$\check{\epsilon}^b = \epsilon_{n+1} - \epsilon_n = (\check{\epsilon}_x^b \quad \check{\epsilon}_y^b \quad \check{\epsilon}_z^b \quad \check{\gamma}_{xy}^b \quad \check{\gamma}_{yz}^b \quad \check{\gamma}_{zx}^b) \quad 3-84$$

Преко израза (2-52) и (2-54) се срачунава γ_{n+1}^b :

$$\gamma_{n+1}^b = \frac{3}{2} \cdot \sqrt{\frac{2}{9} [(\check{\epsilon}_x^b - \check{\epsilon}_y^b)^2 + (\check{\epsilon}_y^b - \check{\epsilon}_z^b)^2 + (\check{\epsilon}_z^b - \check{\epsilon}_x^b)^2]} + \frac{1}{3} (\check{\gamma}_{xy}^{b2} + \check{\gamma}_{yz}^{b2} + \check{\gamma}_{zx}^{b2}) \quad 3-85$$

Уколико на крају инкремента (или субинкремента) канап остаје опуштен, онда се цигла није померала и координате цигле за наредни инкремент су:

$$\epsilon_{n+1}^b = \epsilon_n^b \quad 3-86$$

Уколико је канап затегнут на крају инкремента (или субинкремента) онда се цигла померила и нови положај цигле је дат изразом:

$$\epsilon_{n+1}^b = \epsilon_n^b + \frac{\gamma_{n+1}^b - s^b}{\gamma_{n+1}^b} \cdot (\epsilon_{n+1} - \epsilon_n^b) \quad 3-87$$

Детаљан алгоритам како се одређује вредност G_t^{ref} за прорачун са ажурирањем положаја цигли дат је у потпоглављу 4.3. Вредност G_t^{ref} срачуната према претходном поступку се даље користи у релацијама HS модела.

Релације “Brick” модела као “overlay” модела придружене су релацијама HASP модела, како би се побољшало предвиђање понашања тла у подручју веома малих и малих деформација, као и при промени путање оптерећивања. Користи се исти “Brick” модел који је коришћен код HS-Brick модела. Улога “Brick” модела у проширеној формулацији HASP модела је да на основу претходне историје деформисања одређује колика је вредност тангентног модула смицања G . Та вредност тангентног модула смицања се онда користи даље у свим осталим релацијама HASP модела. Како би вредност Poisson-овог коефицијента μ' остала константна током прорачуна, запремински модул K' ће се срачунавати из G и μ . Зависност K' од G и μ утицаће на вредност параметра k . У наредном потпоглављу дате су све релације проширеног HASP модела.

Формулација “Brick” модела као “overlay” модела који одређује еластично понашање тла је таква да би требало да су задовољена Masing-ова правила за предвиђање хистерезисног понашања при цикличном оптерећењу. Прво правило је да при примарном оптерећењу напонско-деформацијска крива прати облик S-криве. Друго правило је да је напонско-деформацијска крива при растерећењу и поновном оптерећевању геометријски иста као код иницијалног оптерећивања, али је нумерички два пута већа (два пута већа деформација потребна за поновну деградацију модула смицања). Треће правило је да ако величина деформације након растерећења и поновног оптерећивања превазиђе претходну максималну вредност, напонско-деформацијска крива наставља да прати облик S-криве. Четврто правило је да уколико напонско-деформацијска крива током растерећивања и поновног оптерећивање пресече напонско-деформацијску криву из претходног циклуса растерећивања и поновног оптерећивања, наставља да прати ту криву из претходног циклуса.

3.3.3 Напонско-деформацијске релације HASP-E модела

У претходна два потпоглавља приказане су и описане релације које се користе за проширивање релација HASP модела на генерални напонски простор и укључивање промене модула смицања са порастом смичућих деформација. Проширење релација на генерални напонски простор извршено је укључивањем треће напонске инваријанте – Lode-ов угла у изрази за површ течења, граничну површ и површ пластичног потенцијала преко параметра M који више није константа, већ функција Lode-овог угла дата изразом (3-71). Крутост тла у подручју веома малих и малих деформација, као и промена величине крутости тла при промени путање напона и деформација извршено је прикључењем “overlay” “Brick” модела из HS-Brick модела (Cudny & Truty, 2020) релацијама HASP модела. Релације које су изабране за проширење HASP модела су изабране тако да релације проширеног конститутивног модела буду релативно једноставне за имплементацију у рачунарски програм заснован на МКЕ, да основни принципи на којима је развијен HASP модел буду задржани и да сви параметри модела буду јасно дефинисани и да се могу одредити из конвенционалних лабораторијских опита. У наставку овог потпоглавља дате су основне напонско-деформацијске релације HASP-E модела.

Основни принципи на којима се заснива HASP модел су задржани и код HASP-E модела. Разлика између релација HASP и HASP-E модела је у изразима за површ течења, граничну површ, површ пластичног потенцијала и еластичне модуле. Површ течења и

гранична површ у HASP-E моделу се понашају на исти начин као у HASP моделу. Напонска тачка се увек налази на површи течења и пластичне деформације се јављају од почетка оптерећивања. Облик површи течења и граничне површи у посматраној $q - p'$ равни је исти за оба модела и важи радијално мапирање дата изразом (3-1). Вредност Lode-овог угла за напонску тачку $A(p', q, \theta)$ и имагинарну тачку $A(\bar{p}', \bar{q}, \theta)$ је иста. Површ течења у HASP-E моделу дата је изразом:

$$F(p', q, \theta, p'_0) = \frac{q^2}{M(\theta)^2} + p'(p' - p'_0) = 0 \quad 3-88$$

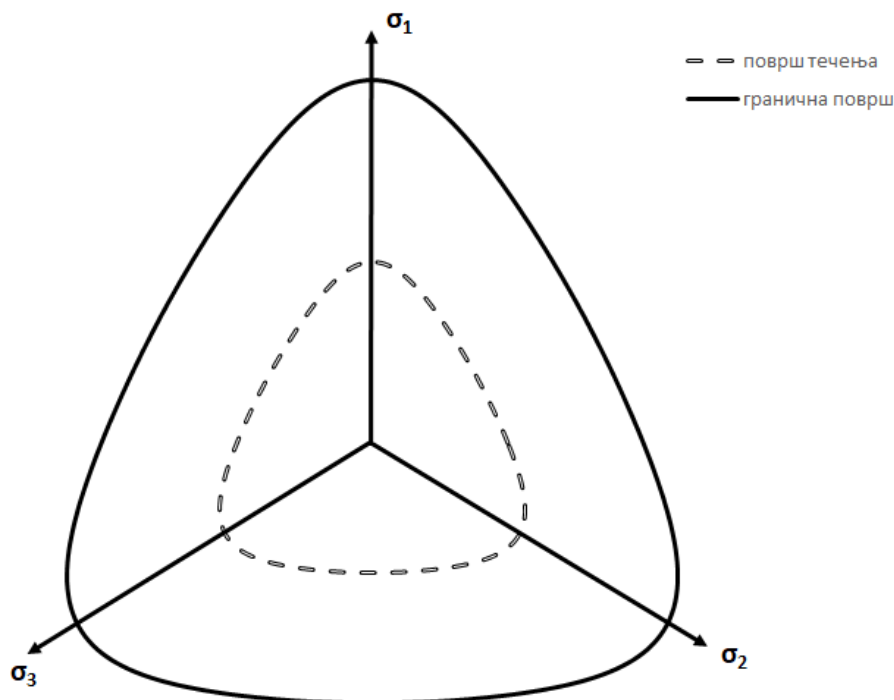
Једначина граничне површи у HASP-E моделу је:

$$\bar{F}(\bar{p}', \bar{q}, \theta, \bar{p}'_0) = \frac{\bar{q}^2}{M(\theta)^2} + \bar{p}'(\bar{p}' - \bar{p}'_0) = 0 \quad 3-89$$

Функција $M(\theta)$ је дата изразом:

$$M(\theta) = X(1 + Y \cdot \sin 3\theta)^Z \quad 3-90$$

У претходном изразу $Z = -0.229$, док се константе X и Y одређују из израза (3-75) и (3-76) на основу задатих вредности нагиба линије критичног стања у триаксијалној компресији M_c и триаксијалној екстензији M_e . Површ течења и гранична површ у девијаторској равни ће имати заобљен конвексан облик (Слика 3-14).



Слика 3-14 Облик површи течења и граничне површи у девијаторској равни за преконсолидовану глину и $\phi_c = \phi_E = \phi_{cv} = 30^\circ$ у HASP-E моделу

У HASP-E користи се неасоцијативан закон течења. Једначина површи пластичног потенцијала је:

$$P(p', q, p'_0) = \frac{q^2}{M(\theta_c)^2} + p'(p' - p'_0) = 0 \quad 3-91$$

У претходном изразу θ_c представља вредност Lode-овог угла θ_c у напонској тачки у којој су потребне вредности градијената пластичног потенцијала. У девијаторској равни површ пластичног потенцијала има облик круга.

Закон ојачања у HASP-E има исти облик као у HASP моделу:

$$dp'_0 = \frac{\nu}{\lambda - \check{\kappa}} p'_0 d\varepsilon_v^p \omega \quad 3-92$$

У претходном изразу $\check{\kappa}$ је променљива чија вредност зависи од вредности тангентног запреминског модула у складу са изразима (3-19) и (3-20). Вредност тангентног запреминског модула се одређује на основу вредности тангентног модула смицања и вредности Poisson-овог коефицијента. У зависности од нивоа деформација од почетка оптерећивања и од тога да ли је дошло до промене путање напона или деформација, мења се вредност $\check{\kappa}$. За велике деформације, када тангентни модул смицања има исту вредност као у HASP моделу, $\check{\kappa}$ има исту вредност као κ . Коефицијент ојачања ω у изразу (3-92) се рачуна преко израза (3-13) – (3-16) и има исту вредност као у HASP моделу. Односно у изразима (3-15) и (3-16) користи се константна вредност параметра κ . Пошто је $\check{\kappa}$ променљива у HASP-E моделу, то значи да линије бубрења у $\nu - \ln p'$ равни немају константни нагиб, што се коси са основним принципима на којима су формулисане релације HASP и MCC модела. Упркос томе, у HASP-E моделу задржаће се променљива вредност $\check{\kappa}$ у закону ојачања, док ће се у изразима за коефицијент ојачања ω и параметар Γ (израз (3-48)) користити константна вредност κ као у HASP моделу. Иако ово представља недоследност у формулацији HASP-E модела, у оквиру ове дисертације усвојиће се да линије бубрења имају константан нагиб κ у $\nu - \ln p'$ равни само при одређивању коефицијента ојачања ω и параметра Γ . У оквиру будућих истраживања треба формулисати нове изразе за коефицијент ојачања ω и параметар Γ за линије бубрења које имају променљив нагиб у $\nu - \ln p'$ равни.

У HASP-E моделу, у изразима (3-13) – (3-16) за одређивање вредности коефицијента ојачања ω , параметар M се мења са $M(\theta_c)$, где је θ_c вредност Lode-овог угла у напонској тачки у којој се рачуна ω . У изразу (3-48) за одређивање параметра Γ увек се користи параметар M_c , како је већ образложено у потпоглављу 3.2.1.

Еластични параметри HASP-E су тангентни модул смицања G и Poisson-ов коефицијент μ . Усвојено је да Poisson-ов коефицијент има константну вредност. Тангентни модул смицања се рачуна из референтног тангентног модула смицања G_t^{ref} преко израза:

$$G = G_t^{ref} \left(\frac{p'}{p_{ref}} \right) \quad 3-93$$

За референтни средњи ефективни напон усвојено је $p'_{ref} = 100 \text{ kPa}$. Референтни тангентни модул смицања G_t^{ref} се рачуна из "Brick" компоненте HASP-E модела према изразу (3-81) и према поступку описаном у потпоглављу 3.3.2. За S-криву је усвојен исти израз као у HSS, HS-Brick и OC_CLAY моделу. У HASP-E моделу је усвојено да се

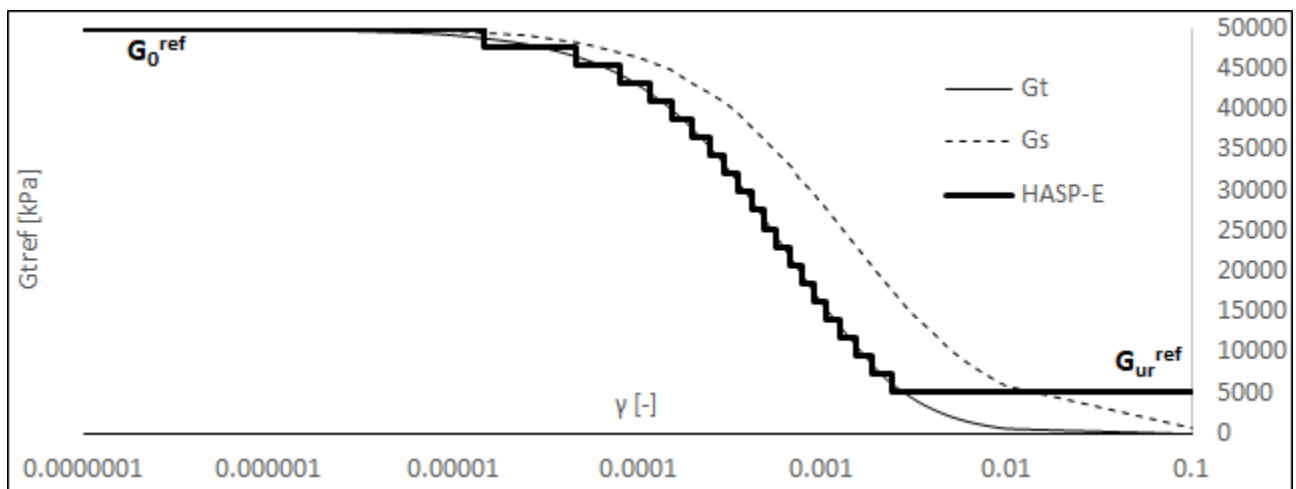
користи $N_b = 20$ цигли за описивање степенасте S-криве и да свакој цигли одговара исти проценат деградације крутости. Степенаста S-крива у HASP-E приказана је на **Слика 3-15**. Усвојено је да су на почетку прорачуна све цигле у координатном почетку у шестодимензионом деформацијском простору.

Тангентни референтни модул смицања G_t^{ref} не може бити већи од максималне вредности референтног модула смицања при веома малим деформацијама G_0^{ref} , нити мањи од референтног модула смицања при великим деформацијама G_{ur}^{ref} . G_{ur}^{ref} се рачуна према изразу:

$$G_{ur}^{ref} = \frac{3(1 - 2\mu)}{2(1 + \mu)} \cdot \frac{1 + e}{\kappa} p_{ref} \quad 3-94$$

Запремински модул се из тангентног модула смицања G рачуна према изразу:

$$K = \frac{2(1 + \mu)}{3(1 - 2\mu)} G \quad 3-95$$



Слика 3-15 Степенаста S-крива у HASP-E моделу за $N_b = 20$ цигли, $G_0^{ref} = 50$ МПа, $G_{ur}^{ref} = 5$ МПа и $\gamma_{0,7} = 5 \cdot 10^{-4}$. (Легенда: G_t - референтни тангентни модул; G_s - референтни секантни модул)

Вредност κ се одређује из израза:

$$\kappa = \frac{1 + e}{K} p' = (1 + e) \cdot p' \cdot \frac{3(1 - 2\mu)}{2G(1 + \mu)} \quad 3-96$$

У изразима (3-94) и (3-96) e и p' су тренутне вредности коефицијента порозности и средњег ефикасног напона. У HASP-E моделу тангентни запремински и тангентни модул смицања зависе од нивоа напона, нивоа деформација од почетка деформисања и од тога да ли је дошло до промене путање напона и деформација током оптерећивања.

3.3.4 Параметри HASP-E модела

Параметри HASP-E модела су: консолидациони параметри λ и k , параметри који описују дренарану смичућу чврстоћу тла у триаксијалној компресији M_c и триаксијалној екстензији M_e , специфична запремина тла при $p' = 1 \text{ kPa}$ на линији критичног стања Γ , Poisson-ов коефицијент μ , референтни модул смицања при веома малим деформацијама G_0^{ref} и смичућа деформација $\gamma_{0,7}$ при којој је секантни модул смицања редукован на 70 % иницијалне вредности. За дефинисање иницијалног стања тла потребно је познавати степен преконсолидације тла OCR или POP и иницијалну вредност коефицијента порозности e_0 .

У потпоглављу 3.2.1 описано је како се одређују консолидациони параметри λ и k , параметри дренаране смичуће чврстоће тла M_c и M_e , иницијална вредност коефицијента порозности e_0 , параметри који описују степен преконсолидације тла Γ , OCR или POP и Poisson-ов коефицијент μ . У HASP-E моделу је усвојено да је вредност Poisson-ов коефицијента константна током целог прорачуна.

Параметри који описују облик S – криве G_0^{ref} и $\gamma_{0,7}$ се могу одредити из разних теренских опита, лабораторијских опита и великог броја емпиријских образаца. У литератури као што је (Benz, 2007), (Obrzud & Truty, 2018) и (Clayton, 2011) може се наћи детаљан преглед опита и емпиријских образаца из којих се ова два параметра могу одредити.

Један од емпиријских образаца за G_0^{ref} у преконсолидованом тлу дао је (Hardin, 1978):

$$G_0^{ref} = 33 \cdot \frac{(2.97 + e)^2}{1 + e} \cdot OCR^k \cdot \left(\frac{p'}{p'_{ref}} \right)^n \quad 3-97$$

У претходном изразу експонент k зависи од индекса пластичности I_p и може имати вредности од 0 ($I_p = 0 \%$) до 0.5 ($I_p \geq 100 \%$). Експонент n може имати вредности од 0.5 до 1.0.

Вредност $\gamma_{0,7}$ се у одсуству експерименталних података може проценити преко израза:

$$\gamma_{0,7} \approx \frac{1}{9G_0^{ref}} [2c'(1 + \cos 2\phi') + \sigma'_v(1 + K_0) \sin 2\phi'] \quad 3-98$$

У претходном изразу ϕ' је угао смичуће чврстоће тла за ефективне напоне, c' је кохезија за ефективне напоне, K_0 је коефицијент притиска тла у стању мировања и σ'_v је вертикални ефективни напон.

Приликом имплементација HASP-E модела у PLAXIS, остављена је могућност да се не морају унети сви претходно набројани параметри. Ако се унесу сви параметри који су потребни за основну формулацију HASP модела, HASP-E модел ће се у потпуности понашати као основна формулација HASP модела. Уколико се унесу сви параметри сем G_0^{ref} и $\gamma_{0,7}$, "Brick" компонента HASP-E модела неће бити активна и еластични модули ће се рачунати као у основној формулацији HASP модела. Уколико се унесу сви параметри сем M_e , HASP-E модел ће користити исту површ течења, граничну површ и површ

пластичног потенцијала као у основној формулацији HASP модела (крugови у девијаторској равни и асоцијативна пластичност), док ће се еластични модули рачунати преко “Brick” компоненте.

3.4 OC_CLAY конститутивни модел

Консолидациони параметри λ и κ HASP модела су у OC_CLAY моделу замењени са модификованим параметрима λ^* и κ^* према изразима:

$$\lambda^* = \frac{\lambda}{1 + e_i^{ref}} \quad 3-99$$

$$\kappa^* = \frac{\kappa}{1 + e_i^{ref}} \quad 3-100$$

где је e_i^{ref} иницијална вредност коефицијента порозности узета као референтна вредност у односу на коју су срачунати еластични модули. На овај начин, уместо тренутне вредности коефицијента порозности e користи се запреминска деформација ε_v , као параметар који описује стање преко консолидоване глине (Слика 3-16). Једначина линије критичног стања у $\ln p' - \varepsilon_v$ равни (Bentley, 2019) је

$$\varepsilon_v = \varepsilon_\Gamma - \lambda^* \cdot \ln p' \quad 3-101$$

где је ε_Γ запреминска деформација која одговара $p' = 1 \text{ kPa}$. У OC_CLAY моделу релације HASP модела су проширене на генерални напонски простор укључивањем треће напонске инваријанте – Lode-овог угла θ преко израза који је предложио (van Eekelen, 1980):

$$M(\theta) = a \cdot [1 + b \cdot \sin(3\theta)]^n \quad 3-102$$

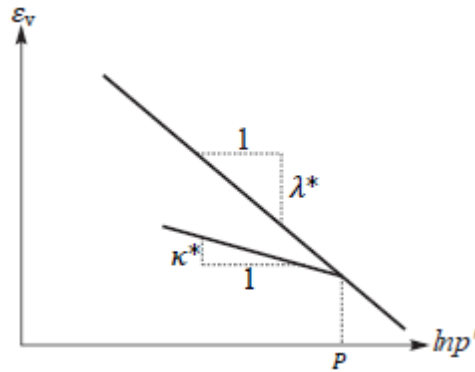
тако да израз (3-2) за површ течења има следећи облик:

$$F(p', q, \theta, p'_0) = \frac{q^2}{M(\theta)^2} + p'(p' - p'_0) = 0 \quad 3-103$$

У изразу (3-102), a и b су дати у функцији редукованих радијуса у компресији и екстензији:

$$a = \frac{r_c}{(1 + b)^n} \quad 3-104$$

$$b = \frac{\left(\frac{r_c}{r_E}\right)^{\frac{1}{n}} - 1}{\left(\frac{r_c}{r_E}\right)^{\frac{1}{n}} + 1} \quad 3-105$$



Слика 3-16 Линија нормалне консолидације и рекомпресије у $\ln p' - \varepsilon_v$ равни – преузето из (Bentley, 2019)

Редуковани радијуси у компресији и екстензији зависе од угла смичуће отпорности тла у компресији и екстензији ϕ'_C и ϕ'_E :

$$r_C = \frac{1}{\sqrt{3}} \cdot \left(\frac{2 \sin \phi'_C}{3 - \sin \phi'_C} \right) \quad 3-106$$

$$r_E = \frac{1}{\sqrt{3}} \cdot \left(\frac{2 \sin \phi'_E}{3 + \sin \phi'_E} \right) \quad 3-107$$

Углови смичуће отпорности у компресији и екстензији су усвојени да су исти и једнаки углу смичуће отпорности тла при константној запремини $\phi_C = \phi_E = \phi_{cv}$. За експонент n је усвојена вредност од -0.229 да би се осигурално да је површ течења конвексна за уобичајене вредности угла унутрашњег трења. Увођењем ове измене, у ОС_CLAY моделу облик површи течења у девијаторској равни нема више облик круга већ заобљеног шестоугла. Једначина граничне површи је на исти начин модификована као површ течења у изразу (3-103). Изрази за параметар стања за тренутну напонску тачку и имагинарну напонску тачку у ОС_CLAY моделу су модификовани у односу на изразе (3-14) и (3-15) тако да буду у складу са изменама датим у изразима (3-99), (3-100) и (3-101):

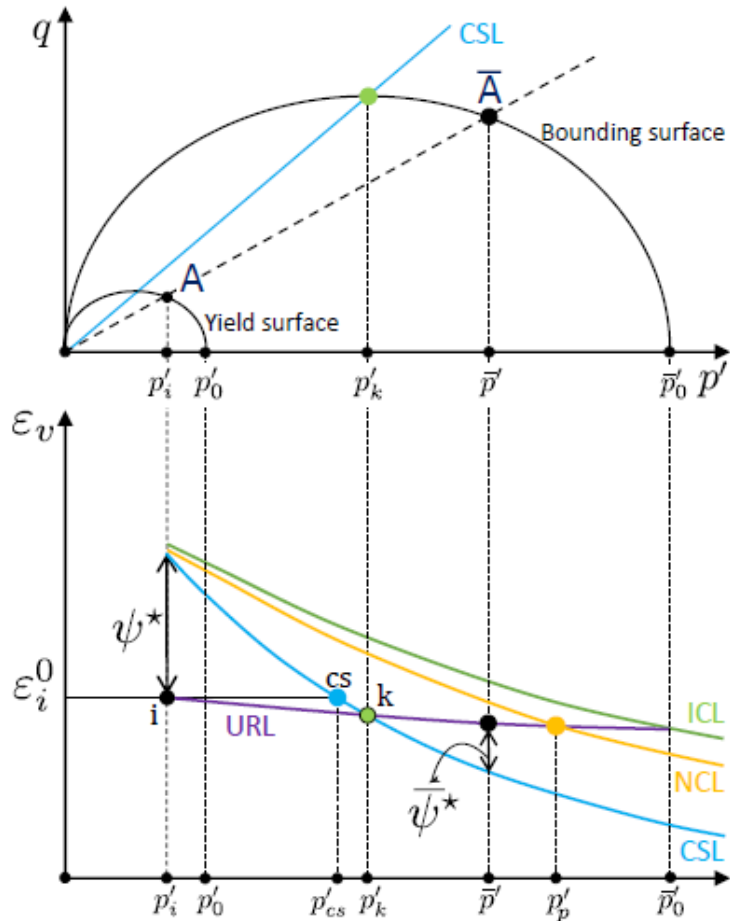
$$\Psi^* = \varepsilon_v + \lambda^* \ln p' - \varepsilon_\Gamma \quad 3-108$$

$$\bar{\Psi}^* = (\lambda^* - \kappa^*) \ln \left(\frac{2M^2}{M^2 + \eta^2} \right) \quad 3-109$$

Модификовани параметри стања у ОС_CLAY Ψ^* и $\bar{\Psi}^*$ имају исто геометријско значење као и параметри стања у HASP моделу Ψ и $\bar{\Psi}$ (Слика 3-17).

Закон ојачања у ОС_CLAY моделу има исти облик као у HASP моделу (израз 3-8). Израз за коефицијент ојачања у ОС_CLAY моделу има исти облик као у HASP моделу (3-13) уз додатак параметра h :

$$\omega = \left(1 + \frac{\bar{\Psi}^* - \Psi^*}{\bar{\Psi}^*} \right) \cdot R \cdot h \quad 3-110$$



Слика 3-17 Шематски приказ напонско-деформацијског стања у OC_CLAY моделу - преузето из (Bentley, 2019)

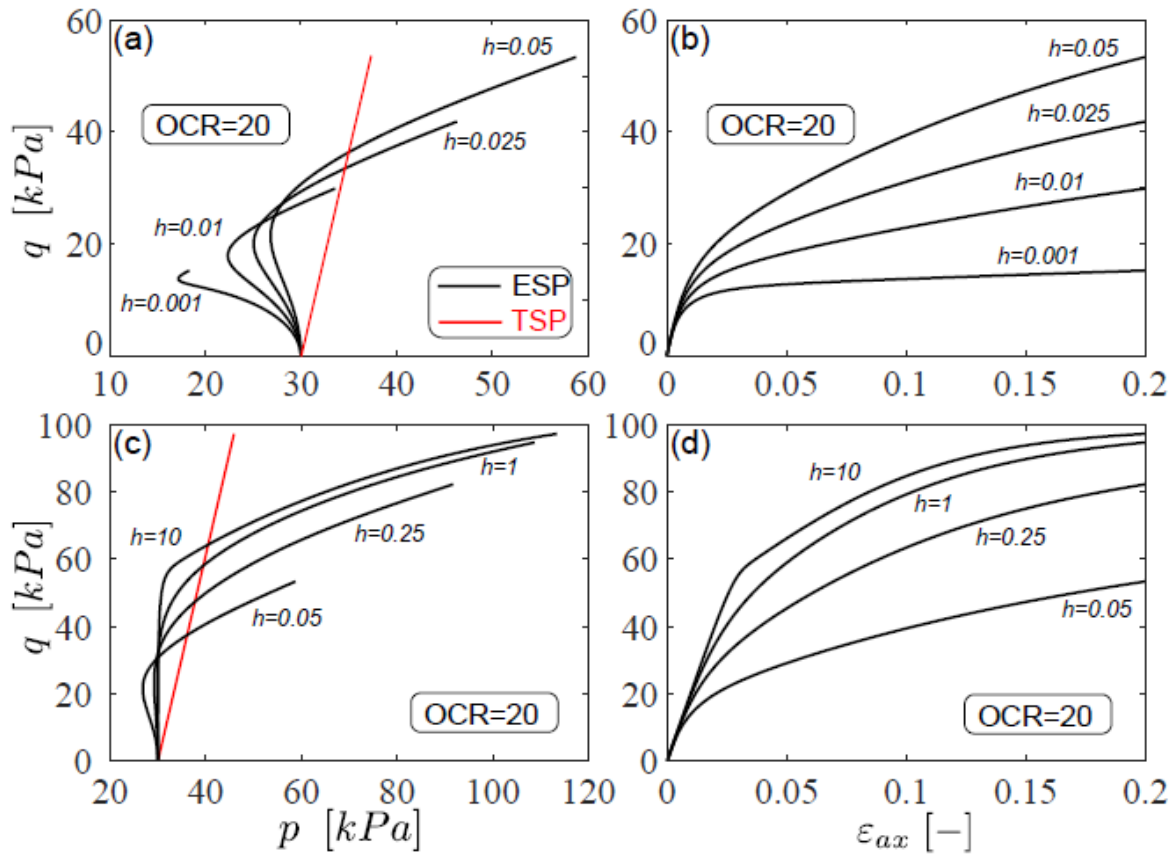
Параметар h нема физичко значење, али може побољшати калибрацију параметара OC_CLAY модела за задате путање напона. Утицај параметра h на облик добијених путања напона приказан је на (Слика 3-18). Узимањем вредности за h мање од 1.0, наглашава се "S" облик путање ефективних напона у недренираним условима. Са друге стране, уколико је h веће од 1.0, величине генерисаних порних притисака у недренираним условима се повећавају (Слика 3-19). Када је $h = 1.0$, израз (3-110) има исти облик као израз (3-8).

У OC_CLAY моделу се користи асоцијативан закон течења, тако да функција пластичног потенцијала има исти облик као функција течења (3-103).

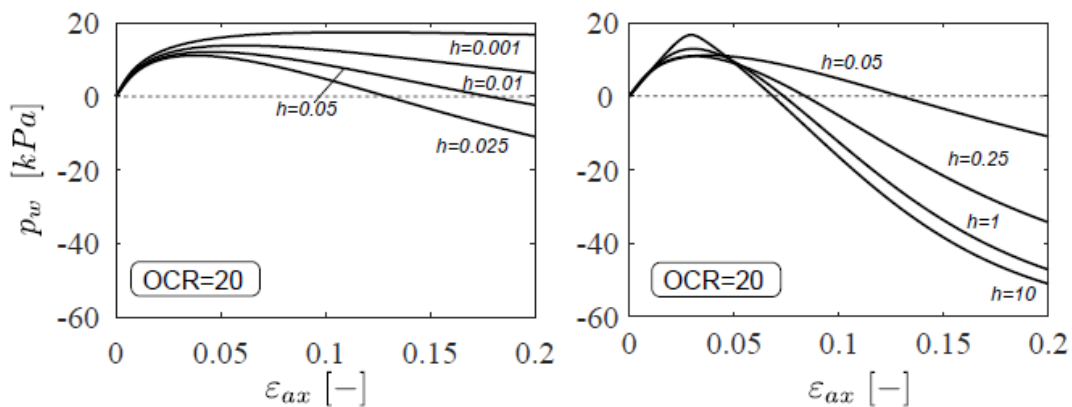
Еластични параметри OC_CLAY модела су запремински модул K_{ur} и модул смицања G_{ur} :

$$K_{ur} = \frac{E_{ur}}{3(1 - 2\mu_{ur})} \quad 3-111$$

$$G_{ur} = \frac{3K_{ur}(1 - 2\mu_{ur})}{2(1 + \mu_{ur})} \quad 3-112$$



Слика 3-18 Путање напона у $p' - q$ равни и напонско-деформације криве за лондонску глину при истој иницијалној вредности OCR за различите вредности параметра h - преузето из (Bentley, 2019)



Слика 3-19 Промена генерисаних порних притисака са порастом аксијалних деформација за лондонску глину при истој иницијалној вредности OCR за различите вредности параметра h - преузето из (Bentley, 2019)

Модул еластичности E_{ur} зависи од нивоа средњег ефективног напона према изразу:

$$E_{ur} = E_{ur}^{ref} \left(\frac{p'}{p_{ref}} \right) \quad 3-113$$

где је E_{ur}^{ref} референтни модул еластичности за $p_{ref} = 100 \text{ kPa}$:

$$E_{ur}^{ref} = p_{ref} \left[\frac{3(1 - 2\mu_{ur})}{\kappa^*} \right] \quad 3-114$$

Како би се омогућила симулација деградације модула смицања са порастом смичућих деформација, у ОС_CLAY модел је укључен Small Strain Overlay (SSO) модел (Benz, 2007). Деградација модула смицања дефинисана је изразом (3-77). Иницијална вредност модула смицања зависи од нивоа средњег ефективног напона према изразу:

$$G_0 = G_0^{ref} \left(\frac{p'}{p_{ref}} \right) \quad 3-115$$

Модул смицања срачунат преко израза (3-115) не може бити мањи од вредности модула смицања за велике деформације срачунатог преко израза (3-112). Односно, ако је $G_t < G_{ur}$ онда је $G_t = G_{ur}$. На овај начин, ОС_CLAY модел може да симулира понашање глина при цикличном оптерећењу.

Приликом примарног оптерећивања тла, ОС_CLAY модел користи као параметар деформабилности референтни едометарски модул E_{oed}^{ref} :

$$E_{oed}^{ref} = \left[\frac{p_{ref}}{\lambda^*} \right] \quad 3-116$$

Преконсолидација глине у ОС_CLAY моделу се задаје преко параметра POP . За одређивање претходног напонског стања, када је глина последњи пут била нормално консолидована, потребна је још вредност коефицијента притиска тла у стању мировања K_0^{NC} . Претходно напонско стање се онда може срачунати помоћу израза (3-28) до (3-36). Како би се срачунала иницијална вредност параметра стања Ψ^* у изразу (3-108), потребно је одредити параметар ε_Γ према изразу:

$$\varepsilon_\Gamma = \lambda^* \cdot \ln \left[\left(\frac{\eta_p^2 + M^2}{2M^2} \cdot p'_p \right)^{\frac{\lambda^* - k^*}{\lambda^*}} \cdot p'_i \frac{\kappa^*}{\lambda^*} \right] \quad 3-117$$

Претходни израз има сличан облик као израз (3-48), док су p'_p , η_p и p'_i дати изразима (3-34), (3-36) и (3-29).

На основу претходног, параметри ОС_CLAY модела су: параметри еластичности E_{ur}^{ref} и μ_{ur} ; референтни едометарски модул E_{oed}^{ref} ; референтни средњи ефективни напон p_{ref} (уобичајено се усваја да је $p_{ref} = 100 \text{ kPa}$); референтни модул смицања при малим деформацијама G_0^{ref} и смичућа деформација $\gamma_{0.7}$; угао смичуће отпорности тла при константној запремини у дренираним условима φ'_{cv} ; коефицијент притиска тла у стању мировања за нормално консолидовано тло K_0^{NC} и параметар POP , као и параметар h .

Формулација HASP-E модела је извршена по угледу на ОС_CLAY модел, али уз задржавање истог сета улазних параметара конститутивног модела и свих основних принципа на којима се заснива HASP модел. Битне разлике HASP-E и ОС_CLAY модела су:

- HASP-E модел користи “Brick” модел као overlay модел за описивање крутости тла у подручју веома малих деформација са карактеристикама хистерезисног понашања, док OC_CLAY модел користи SSO overlay модел. Оба overlay модела користе исте улазне параметре. Према (Cudny & Truty, 2020), “Brick” overlay модел превазилази неке недостатке SSO overlay модела при описивању хистерезисног понашања.

- HASP-E модел користи неасоцијативан закон течења, док OC_CLAY модел користи асоцијативан закон течења. Према (Potts & Zdravković, 1999), за опис напонско-деформацијских релација тла адекватније је коришћење неасоцијативног закона течења у односу на асоцијативни. Облик површи пластичног потенцијала у девијаторској равни код HASP-E модела је круг, док је у OC_CLAY моделу заобљени шестоугао.

- Изрази за коефицијент ојачања ω у HASP-E моделу и OC_CLAY се међусобно разликују, као и закони ојачања које користе ова два модела.

- Облик површи течења и граничне површи у девијаторској равни је исти у оба модела само када су углови смичуће отпорности у компресији и екстензији међусобно једнаки. У OC_CLAY моделу је усвојено да су ова два угла увек једнака, док HASP-E модел узима у обзир разлику ова два угла.

- HASP-E модел користи коефицијент порозности e као параметар који описује стање преконсолидоване глине, док OC_CLAY модел користи запреминску деформацију ε_v .

- Улазни параметри HASP-E и OC_CLAY модела се разликују. HASP-E користи нагибе линије критичног стања у $q - p'$ равни у триаксијалној компресији и екстензији, док OC_CLAY модел користи угао смичуће отпорности тла при константној запремини као параметар који описује чврстоћу тла у дренираним условима. Такође, разликују се и параметри који описују крутост тла у подручју великих деформација. HASP-E користи параметре λ и κ , док се у OC_CLAY моделу користе параметри E_{oed}^{ref} и E_{ur}^{ref} .

Поглавље 4 Имплементација HASP и HASP-E модела

4.1 Увод

Како је описано у Поглављу 2, примена нелинеарног конститутивног модела у анализи геотехничких проблема у МКЕ захтева имплементацију конститутивног модела у рачунарски програм. У овом поглављу приказани су алгоритми за имплементацију основне формулације HASP модела (потпоглавље 3.2) и проширене формулације HASP модела која узима у обзир промену модула смицања са променом смичућих деформација и утицај треће напонске инваријанте (потпоглавље 3.3).

За имплементацију основне и проширене формулација HASP модела коришћене су имплицитне и експлицитне методе нумеричке интеграције напона. Од имплицитних метода коришћене су: метода водећег параметра (оба модела) и алгоритам (Borja & Lee, 1990) (само HASP). Од експлицитних метода коришћен је модификовани Euler-ов поступак са аутоматском субинкрементацијом и контролом грешке, као и Runge-Kutta-Dormand-Prince (RKDP) поступак са аутоматском субинкрементацијом и контролом грешке. Обе експлицитне методе коришћене су за оба модела.

Верификација поступка имплементације конститутивних модела је извршена поређењем резултата симулације триаксијалних опита у PLAXIS-у 2D са резултатима за Euler-ову интеграцију деформација дуж познате путање напона (EIPN) и са резултатима предвиђања OC_CLAY модела. На крају поглавља извршена је анализа ефикасности, тачности и робустности алгоритама за интеграцију конститутивних релација HASP и HASP-E модела на примеру једног граничног проблема – одређивање носивости тракастог темеља у недренираним условима у програму PLAXIS.

4.2 Интеграција конститутивних релација основне формулације HASP модела

4.2.1 Метода водећег параметра

Као што је описано у потпоглављу 2.7.3.2, у методи водећег параметра све непознате величине на крају једног инкремента треба да се изразе у функцији једне непознате – водећег параметра. Решавањем водеће једначине налази се вредност водећег параметра и могу се срачунати све величине на крају инкремента. На почетку инкремента, у конфигурацији t , познати су напони ${}^t\sigma$, укупне деформације ${}^t\epsilon$, укупне пластичне деформације ${}^t\epsilon^p$ и параметри стања ${}^t\mathbf{k}$. Такође, на посматраном инкременту, познати су прираштаји деформација $\Delta\epsilon$ као и укупне деформације ${}^{t+\Delta t}\epsilon$ на крају инкремента, у конфигурацији $t + \Delta t$. Потребно је одредити напоне ${}^{t+\Delta t}\sigma$, укупне пластичне деформације ${}^{t+\Delta t}\epsilon^p$ и параметре стања ${}^{t+\Delta t}\mathbf{k}$ на крају инкремента у конфигурацији $t + \Delta t$.

Први корак у овој методи је да се све непознате величине изразе у функцији једног параметра ζ (водећи параметар):

$${}^{t+\Delta t}\boldsymbol{\sigma} = {}^{t+\Delta t}\boldsymbol{\sigma}({}^t\boldsymbol{\sigma}, {}^t\boldsymbol{\varepsilon}, {}^t\boldsymbol{\varepsilon}^p, {}^t\mathbf{k}, {}^{t+\Delta t}\boldsymbol{\varepsilon}, \zeta) \quad 4-1$$

$${}^{t+\Delta t}\boldsymbol{\varepsilon}^p = {}^{t+\Delta t}\boldsymbol{\varepsilon}^p({}^t\boldsymbol{\sigma}, {}^t\boldsymbol{\varepsilon}, {}^t\boldsymbol{\varepsilon}^p, {}^t\mathbf{k}, {}^{t+\Delta t}\boldsymbol{\varepsilon}, \zeta) \quad 4-2$$

$${}^{t+\Delta t}\mathbf{k} = {}^{t+\Delta t}\mathbf{k}({}^t\boldsymbol{\sigma}, {}^t\boldsymbol{\varepsilon}, {}^t\boldsymbol{\varepsilon}^p, {}^t\mathbf{k}, {}^{t+\Delta t}\boldsymbol{\varepsilon}, \zeta) \quad 4-3$$

Након тога потребно је формирати водећу функцију $f(\zeta)$ и наћи нулу функције:

$$f(\zeta) = 0 \quad 4-4$$

Решење претходне једначине представља вредност водећег параметра на крају инкремента ${}^{t+\Delta t}\zeta$. Једначина (4-4) је нелинеарна једначина и начини на који се она може решити су приказани у потпоглављу 4.2.1.2. За водећу функцију се најчешће бира функција течења. Када је позната вредност водећег параметра на крају инкремента, онда се могу срачунати све непознате величине на крају инкремента према изразима (4-1) – (4-3).

За одређивање матрице крутости елемената у инкрементално-итеративном поступку у МКЕ потребно је срачунати еласто-пластичну конститутивну матрицу \mathbf{D}^{ep} која треба да буде конзистентна са алгоритмом интеграције напона:

$${}^{t+\Delta t}\mathbf{D}^{ep} = \frac{\partial {}^{t+\Delta t}\boldsymbol{\sigma}}{\partial {}^{t+\Delta t}\boldsymbol{\varepsilon}} \quad 4-5$$

Као што је објашњено у потпоглављу 2.7.2, уместо еласто-пластичне матрице може се користити еластична конститутивна матрица \mathbf{D}'^e . Одређивање матрице \mathbf{D}^{ep} из израза (4-5) може бити веома алгебарски захтевно, док коришћење матрице \mathbf{D}'^e уместо матрице \mathbf{D}^{ep} не утиче на тачност решења већ само на брзину прорачуна. Због тога је у овом алгоритму за интеграцију конститутивних релација, као и у свим наредним који су приказани у овој докторској дисертацији, коришћена еластична конститутивна матрица за формирање матрице крутости елемената.

4.2.1.1 Избор водећег параметра

Да би поступак интеграције напона у методи водећег параметра био довољно ефикасан, тачан и робустан, потребно је адекватно одабрати која величина ће бити водећи параметар. За водећи параметар треба изабрати ону величину која има јасно физичко значење, да се све непознате величине могу изразити у функцији водећег параметра, да водећа функција буде монотono опадајућа и да има једну нулу. Уколико водећа функција није монотono опадајућа, конвергенција поступка тражења нуле функције може бити доведена у питање. Постојање више нула може угрозити тачност поступка, јер би онда било потребно наћи која нула функције је она која се тражи. У случају постојања више нула функције, само једна одговара физички могућем стању напона и деформација. Због тога би било неопходно увођење додатних услова које нула функције мора да задовољи, како би се избегло налажење лажних нула које одговарају немогућем стању напона и деформација. Налажење свих нула функције и провера која је „права“ нула утиче на ефикасност нумеричког поступка.

У (Којић & Bathe, 2003) приказан је поступак нумеричке интеграције МСС модела где је за водећи параметар изабран прираштај пластичне запреминске деформације, док су (Vukićević & Rakić, 2010) за исти модел изабрали средњи ефективни напон као водећи параметар. Средњи ефективни напон, p' има јасно физичко значење, са увек познатим границама могућих напонских стања и све непознате величине се могу изразити у функцији p' (Vukićević, 2010). За HASP модел такође је усвојен средњи ефективни напон као водећи параметар (Jocković, 2017), (Jocković & Vukićević, 2017) и (Jocković & Vukićević, 2018a). Као водећа функција увек је бирана функција течења.

У овој дисертацији је приказан поступак интеграције HASP модела где је за водећи параметар изабран средњи ефективни напон (Jocković, 2017), као и за HASP-E модел. Иако је средњи ефективни напон веома погодан као водећи параметар, могућа је појава више нула водеће функције, поготово у ситуацијама када путања напона прелази преко линије критичног стања унутар граничне површи (достигнуто „прво карактеристично стање“ у (Jocković, 2017)). У том случају потребно је увођење додатних услова за нулу функције што ће бити описано у потпоглављу 4.2.1.4.

Избор прираштаја пластичне запреминске деформације $\Delta\varepsilon_v^p$ као водећег параметра у HASP моделу није погодан јер је ову величину тешко предвидети због веома изражених ефеката дилатанције код тврдих преконсолидованих глина (Jocković, 2017). Као водећи параметар може се изабрати и параметар ојачања на крају инкремента ${}^{t+\Delta t}p'_0$. Ове две величине, $\Delta\varepsilon_v^p$ и ${}^{t+\Delta t}p'_0$, су прикладне да буду водећи параметар, због тога што се све друге непознате величине могу изразити у зависности од њих. Али, код ових величина је проблем више нула водеће функције веома изражен, што их чини непогодним са аспекта ефикасности нумеричког поступка. Иако због тога нису адекватне за HASP модел, могле би да се користе за једноставније конститутивне моделе као што је МСС модел.

4.2.1.2 Поступак тражења водећег параметра (нуле водеће функције)

Начин и брзина решавања једначине (4-4) веома утиче на ефикасност и тачност методе водећег параметра. Једначина (4-4) се решава нумерички и поступак њеног решавања треба да буде ефикасан, односно потребно је спровести што мање рачунских операција. Поступак решавања треба да буде и довољно тачан, односно да срачуната вредност водећег параметра ${}^{t+\Delta t}\zeta$ задовољава једначину (4-4) са одређеном толеранцијом TOL :

$$|{}^{t+\Delta t}f({}^{t+\Delta t}\zeta)| \leq TOL \quad 4-6$$

Једна од најједноставнијих метода за тражење нуле функције је метода бисекције. У оквиру ове методе потребно је дефинисати интервал водећег параметра $[{}^{t+\Delta t}\zeta_1, {}^{t+\Delta t}\zeta_2]$ на којем је водећа функција монотона и мења знак, односно:

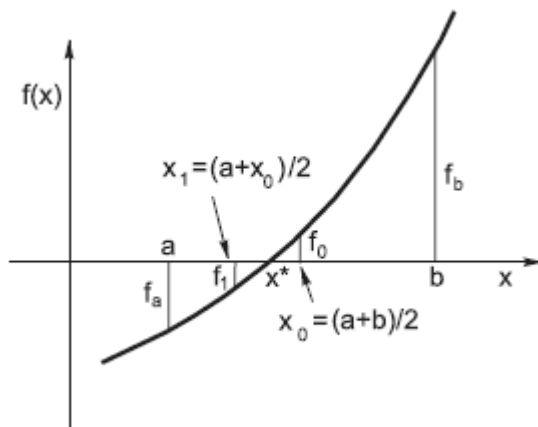
$${}^{t+\Delta t}f_1({}^{t+\Delta t}\zeta_1) \cdot {}^{t+\Delta t}f_2({}^{t+\Delta t}\zeta_2) < 0 \quad 4-7$$

Нула функције се онда тражи половљењем интервала на мање интервале тако да важи (4-7) све док се не задовољи (4-6). Интервал на коме водећа функција мења знак се може наћи у околини познате вредности водећег параметра ${}^t\zeta$. Почетне границе интервала су:

$${}^{t+\Delta t}\zeta_1 = C_1 {}^t\zeta \quad 4-8$$

$${}^{t+\Delta t}\zeta_2 = C_2 {}^t\zeta$$

Где је C_1 константа која има вредност мању од 1.0, а C_2 константа која има вредност већу од 1.0. Вредности C_1 и C_2 се одређују тако да важи (4-7), односно границе интервала се шире у малим инкрементима све док се не задовољи услов да функција мења знак на том интервалу. Након што је дефинисан интервал, методом бисекције се налази нула водеће функције (Слика 4-1).



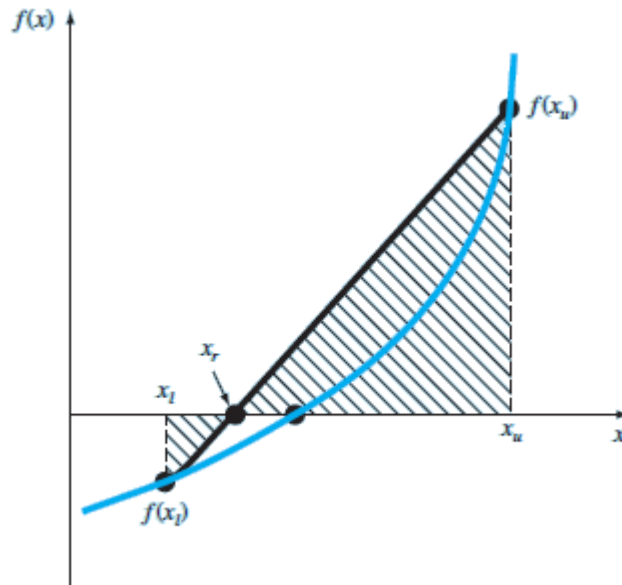
Слика 4-1 Метода бисекције за тражење нуле функције – преузето из (Кojić & Bathe, 2003)

Поступак тражење интервала где водећа функција мења знак је мана методе бисекције. За сваку пробну границу интервала потребно је срачунати вредност водећих функција ${}^{t+\Delta t}f_1$ и ${}^{t+\Delta t}f_2$. Што је већи број пробних граница интервала, поступак тражења нуле дуже траје и метода бисекције је онда потенцијално мање ефикасна од других метода. Још једна мана ове методе може да буде постојање више нула на разматраном интервалу, јер на интервалу на коме функција мења знак може постојати непаран број нула. Налажење свих нула и провера која је она нула која се тражи, додатно продужава трајање прорачуна.

Метода „лажне нуле“ (лат. regula falsi) је слична методи бисекције и узима у обзир да ако је једна од вредности функције на границама интервала ближа нули, онда ће нула функције бити ближа тој граници интервала (Слика 4-2). Потенцијална нула функције се тражи у пресеку праве линије која спаја вредности функције на границама интервала ${}^{t+\Delta t}f_1({}^{t+\Delta t}\zeta_1)$ и ${}^{t+\Delta t}f_2({}^{t+\Delta t}\zeta_2)$ и хоризонталне осе:

$${}^{t+\Delta t}\zeta_n = {}^{t+\Delta t}\zeta_2 - \frac{{}^{t+\Delta t}f_2({}^{t+\Delta t}\zeta_2) \cdot ({}^{t+\Delta t}\zeta_1 - {}^{t+\Delta t}\zeta_2)}{{}^{t+\Delta t}f_1({}^{t+\Delta t}\zeta_1) - {}^{t+\Delta t}f_2({}^{t+\Delta t}\zeta_2)} \quad 4-10$$

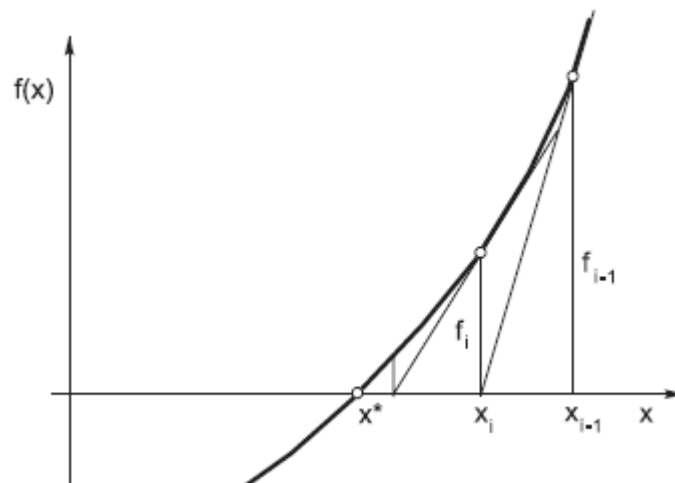
Срачуната вредност ${}^{t+\Delta t}\zeta_n$ мења ону границу интервала која има исту вредност функције као ${}^{t+\Delta t}f_n({}^{t+\Delta t}\zeta_n)$, тако да границе интервала увек задовољавају услов (4-7). Поступак се понавља све док (4-6) не буде задовољено. Као и код методе бисекције, мана ове методе је трајање поступка тражења адекватног интервала.



Слика 4-2 Метода *regula falsi* – преузето из (Chapra & Canale, 2021)

Један од најкоришћенијих поступак за тражење нуле функције је Newton-ов поступак. У оквиру овог поступка није потребно дефинисати интервал у коме се налази нула функције, што је велика предност у односу на претходне две методе. Поступак тражења нуле функције приказан је на (Слика 4-3). Потенцијална нула функције се тражи у пресеку тангенте повучене из вредности функције за претходно претпостављену нулу и хоризонталне осе:

$${}^{t+\Delta t}\zeta_{i+1} = {}^{t+\Delta t}\zeta_i - \frac{{}^{t+\Delta t}f_i({}^{t+\Delta t}\zeta_i)}{{}^{t+\Delta t}f_i'({}^{t+\Delta t}\zeta_i)} \quad 4-11$$



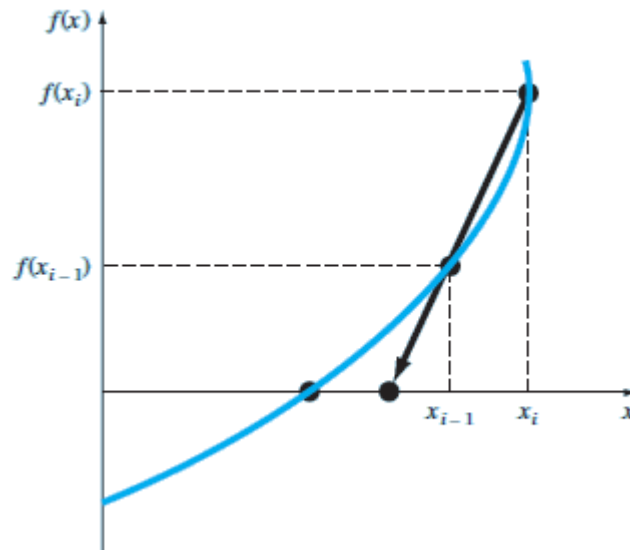
Слика 4-3 Newton-ов поступак тражења нуле функције – преузето из (Kojić & Bathe, 2003)

У Newton-овом поступку потребно је одредити први извод функције, што може код сложених функција бити алгебарски веома захтевно или чак у неким случајевима немогуће. Потреба за тражењем првог извода функције се може превазићи коришћењем секантне методе. У секантној методи, први извод функције се може

приближно срачунати преко методе коначних разлика, тако да се потенцијална нула функције може одредити из израза:

$${}^{t+\Delta t}\zeta_{i+1} = {}^{t+\Delta t}\zeta_i - \frac{{}^{t+\Delta t}f_i({}^{t+\Delta t}\zeta_i) \cdot ({}^{t+\Delta t}\zeta_{i-1} - {}^{t+\Delta t}\zeta_i)}{{}^{t+\Delta t}f_{i-1}({}^{t+\Delta t}\zeta_{i-1}) - {}^{t+\Delta t}f_i({}^{t+\Delta t}\zeta_i)} \quad 4-12$$

Да би се одредила потенцијална нула у секантној методи, потребно је знати две претходне потенцијалне нуле. Поступак тражења нуле функције у секантној методи илустрован је на (Слика 4-4).



Слика 4-4 Секантна метода за тражење нуле функције – преузето из (Charpa & Canale, 2021)

У Halley-евом поступку, нула функције се може наћи из израза:

$${}^{t+\Delta t}\zeta_{i+1} = {}^{t+\Delta t}\zeta_i - \frac{2 \cdot {}^{t+\Delta t}f_i({}^{t+\Delta t}\zeta_i) \cdot {}^{t+\Delta t}f'_i({}^{t+\Delta t}\zeta_i)}{2 \cdot [{}^{t+\Delta t}f'_i({}^{t+\Delta t}\zeta_i)]^2 - {}^{t+\Delta t}f_i({}^{t+\Delta t}\zeta_i) \cdot {}^{t+\Delta t}f''_i({}^{t+\Delta t}\zeta_i)} \quad 4-13$$

Halley-ев поступак је тачнији и брже конвергира него Newton-ова и секантна метода. У овом поступку потребно је знати претходну потенцијалну нулу функције, као и први и други извод функције. Проблем тражења првог и другог извода компликованих функција се може превазићи коришћењем методе коначних разлика. Први и други извод функције се онда могу одредити преко израза:

$${}^{t+\Delta t}f'_i({}^{t+\Delta t}\zeta_i) = \frac{{}^{t+\Delta t}f_i({}^{t+\Delta t}\zeta_i + \Delta\zeta) - {}^{t+\Delta t}f_i({}^{t+\Delta t}\zeta_i - \Delta\zeta)}{2 \cdot \Delta\zeta} \quad 4-14$$

$${}^{t+\Delta t}f''_i({}^{t+\Delta t}\zeta_i) = \frac{{}^{t+\Delta t}f_i({}^{t+\Delta t}\zeta_i + \Delta\zeta) - 2 \cdot {}^{t+\Delta t}f_i({}^{t+\Delta t}\zeta_i) + {}^{t+\Delta t}f_i({}^{t+\Delta t}\zeta_i - \Delta\zeta)}{\Delta\zeta^2} \quad 4-15$$

У претходним изразима $\Delta\zeta$ представља мали прираштај величине ζ_i . Уколико је $\Delta\zeta$ довољно мало, онда се први и други извод функције може адекватно апроксимирати преко израза (4-14) и (4-15). У оквиру ове дисертације, у методи водећег параметра коришћена је Halley-ева метода за одређивање водећег параметра.

4.2.1.3 Релације нумеричке интеграције HASP модела

У наставку биће приказане све потребне релације за интеграцију основне формулације HASP модела према методи водећег параметра. Све релације и комплетан алгоритам нумеричке интеграције су преузети из (Јосковић, 2017). За водећи параметар изабран је средњи ефективни напон на крају инкремента ${}^{t+\Delta t}p'$. У (Јосковић, 2017) коришћена је метода бисекције за тражење нуле функције, док је у овој дисертацији коришћена Halley-ева метода. Тиме се елиминише потреба за тражењем интервала где водећа функција мења знак и сам процес тражења нуле се знатно убрзава.

Као што је приказано у потпоглављу 2.7.3.2, метода водећег параметра представља генерализацију методе радијалног пресликавања у оквиру којих је први корак еластично предвиђање. Односно, претпоставља се да нема пластичних деформација и да су деформације у разматраном инкременту еластичне. Уколико су напони срачунати из еластичног предвиђања изван површи течења, онда постоје пластичне деформације на посматраном инкременту и врши се корекција напонског стања враћањем на површ течења. У HASP моделу напонска тачка се увек налази на површи течења и нема еластичне области, тако да у алгоритму нумеричке интеграције нема корака са еластичним предвиђањем. Претходна тврдња не важи у свим ситуацијама. Приликом растеређивања, када је напонска путања усмерена унутар површи течења, HASP модел може предвиђати само еластичне деформације. Како се то узима у обзир биће приказано у наставку овог потпоглавља.

Уколико је позната вредност водећег параметра ${}^{t+\Delta t}p'$, из закона течења може се одредити прираштај пластичних запреминских деформација:

$$\Delta\varepsilon_v^p = \Delta\lambda \frac{\partial P}{\partial p'} \quad 4-16$$

У претходном изразу $\Delta\lambda$ је позитивни скаларни мултипликатор. Основна формулација HASP модела користи асоцијативни закон течења, односно $F(p', q, p'_0) = P(p', q, p'_0)$. Функција течења у конфигурацији $t + \Delta t$ је:

$${}^{t+\Delta t}F({}^{t+\Delta t}p', {}^{t+\Delta t}q, {}^{t+\Delta t}p'_0) = \frac{{}^{t+\Delta t}q^2}{M^2} + {}^{t+\Delta t}p'({}^{t+\Delta t}p' - {}^{t+\Delta t}p'_0) = 0 \quad 4-17$$

У претходном изразу ${}^{t+\Delta t}q$ и ${}^{t+\Delta t}p'_0$ су девијаторски напон и параметар ојачања у конфигурацији $t + \Delta t$. Израз (4-16) се онда може написати у облику:

$$\Delta\varepsilon_v^p = \Delta\lambda(2 {}^{t+\Delta t}p' - {}^{t+\Delta t}p'_0) \quad 4-18$$

Из претходног израза, може се изразити пластични мултипликатор:

$$\Delta\lambda = \frac{\Delta\varepsilon_v^p}{2 {}^{t+\Delta t}p' - {}^{t+\Delta t}p'_0} \quad 4-19$$

Прираштај пластичне дисторзијске деформација се може одредити из закона течења:

$$\Delta\varepsilon_q^p = \Delta\lambda \frac{\partial P}{\partial q} = \Delta\lambda \frac{2 {}^{t+\Delta t}q}{M^2} \quad 4-20$$

Прираштаји пластичних девијаторских деформација се могу одредити из закона течења узимајући у обзир изразе (2-16), (2-19) и (2-51):

$$\Delta \varepsilon'^p = 3\Delta L \frac{{}^{t+\Delta t} \mathbf{s}}{M^2} \quad 4-21$$

У претходном изразу ${}^{t+\Delta t} \mathbf{s}$ је вектор девијаторских напона у конфигурацији $t + \Delta t$. Коефицијент порозности на крају инкремента ${}^{t+\Delta t} e$ се може добити интеграцијом израза (2-56):

$${}^{t+\Delta t} e = (1 + e_0) \cdot \exp(- {}^{t+\Delta t} \varepsilon_v) - 1 \quad 4-22$$

У претходном изразу уведен је негативни предзнак испред укупних запреминских деформација на крају инкремента ${}^{t+\Delta t} \varepsilon_v$ како би компресија била позитивна. Параметар ојачања на крају инкремента ${}^{t+\Delta t} p'_0$ се може одредити из израза:

$${}^{t+\Delta t} p'_0 = {}^t p'_0 \cdot \exp\left(\frac{({}^t e - {}^{t+\Delta t} e) {}^t \omega}{\lambda - \kappa}\right) \cdot \left(\frac{{}^t p'}{{}^{t+\Delta t} p'}\right)^{\frac{\kappa {}^t \omega}{\lambda - \kappa}} \quad 4-23$$

У претходном изразу ${}^t p'$, ${}^t p'_0$ и ${}^t e$ су редом средњи ефективни напон, параметар ојачања и коефицијент порозности у конфигурацији t . Прираштај пластичне запреминске деформације може се одредити из закона ојачања:

$$\Delta \varepsilon_v^p = \frac{\lambda - \kappa}{1 + {}^t e} \cdot \frac{1}{{}^t \omega} \cdot \ln\left(\frac{{}^{t+\Delta t} p'_0}{{}^t p'_0}\right) \quad 4-24$$

Вектор девијаторских напона на крају инкремента ${}^{t+\Delta t} \mathbf{s}$ може се добити из израза:

$${}^{t+\Delta t} \mathbf{s} = \frac{\Delta \varepsilon' + \frac{{}^t \mathbf{s}}{2 {}^t G}}{\frac{3\Delta L}{M^2} + \frac{1}{2 {}^t G}} \quad 4-25$$

У претходном изразу $\Delta \varepsilon'$ је прираштај девијаторских деформација на посматраном инкременту, ${}^t \mathbf{s}$ је и вектор девијаторских напона и ${}^t G$ је модул смицања у конфигурацији t .

У случају преконсолидованог тла, када напонски однос достигне вредност $\eta = M$, важи:

$$2 {}^{t+\Delta t} p' - {}^{t+\Delta t} p'_0 = 0 \quad 4-26$$

Напонски однос достиже ту вредност у два случаја. Први случај је прелазак из контракције у експанзију ($\omega \rightarrow \infty$), а други случај је када је достигнуто критично стање ($\omega = 1$). Тада је прираштај пластичне запреминске деформације једнак нули и није могуће срачунати пластични мултипликатор из израза (4-19). У том случају пластични мултипликатор се може срачунати користећи пластичну дисторзијску деформацију. Услов када се пластични мултипликатор рачуна на такав начин може се добити постављањем услова да израз (4-26) буде задовољен са одређеном толеранцијом:

$$|2 {}^{t+\Delta t}p' - {}^{t+\Delta t}p'_0| \leq TOL \quad 4-27$$

И пластични мултипликатор се онда одређује из израза:

$$\Delta\lambda = \frac{M}{2 \cdot {}^{t+\Delta t}p' \cdot {}^t\omega} \left(\Delta\varepsilon_q - \frac{{}^{t+\Delta t}q - {}^tq}{3 {}^tG} \right) \quad 4-28$$

У претходном изразу $\Delta\varepsilon_q$ је прираштај дисторзијске деформације и tq је девијаторски напон у конфигурацији t . Детаљно извођење израза (4-23), (4-24), (4-25) и (4-28) може се наћи у (Јосковић, 2017).

Уколико се у алгоритму добије да је вредност пластичног мултипликатора мања од нуле, $\Delta\lambda < 0$, што је у супротности са дефиницијом пластичног мултипликатора, онда се пластичном мултипликатору додељује вредност нула, $\Delta\lambda = 0$. У том случају нема пластичних деформација и понашање тла је потпуно еластично. Један од случаја када се може добити да је $\Delta\lambda = 0$ је случај растерећења. Ако би се у случају постојања више нула водеће функције за пронађену нулу водеће функције добило да је $\Delta\lambda < 0$, то би био индикатор да је нађена нула „лажна нула“. Тада би требало претпоставити нову иницијалну вредност водећег параметра и поновити алгоритам тражења нуле. Ако се за све нађене нуле добије да је $\Delta\lambda < 0$, онда је понашање тла еластично и поставља се $\Delta\lambda = 0$.

Напони на крају инкремента се могу срачунати из познатог средњег ефективног ${}^{t+\Delta t}p'$ напона и вектора девијаторских напона ${}^{t+\Delta t}\mathbf{s}$:

$${}^{t+\Delta t}\boldsymbol{\sigma} = \begin{bmatrix} {}^{t+\Delta t}\sigma_x \\ {}^{t+\Delta t}\sigma_y \\ {}^{t+\Delta t}\sigma_z \\ {}^{t+\Delta t}\tau_{xy} \\ {}^{t+\Delta t}\tau_{yz} \\ {}^{t+\Delta t}\tau_{zx} \end{bmatrix} = \begin{bmatrix} {}^{t+\Delta t}S_{xx} \\ {}^{t+\Delta t}S_{yy} \\ {}^{t+\Delta t}S_{zz} \\ {}^{t+\Delta t}S_{xy} \\ {}^{t+\Delta t}S_{yz} \\ {}^{t+\Delta t}S_{zx} \end{bmatrix} + \begin{bmatrix} {}^{t+\Delta t}p' \\ {}^{t+\Delta t}p' \\ {}^{t+\Delta t}p' \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad 4-29$$

У наставку ће бити дате све величине које су познате на почетку инкремента. Напони и напонске инваријанте на почетку инкремента су:

$${}^t\boldsymbol{\sigma} = ({}^t\sigma_x \quad {}^t\sigma_y \quad {}^t\sigma_z \quad {}^t\tau_{xy} \quad {}^t\tau_{yz} \quad {}^t\tau_{zx})^T \quad 4-30$$

$$4-31$$

$${}^tp = \frac{1}{3} ({}^t\sigma_x + {}^t\sigma_y + {}^t\sigma_z) \quad 4-32$$

$$4-33$$

$${}^tq = \sqrt{\frac{1}{2} \left[({}^t\sigma_x - {}^t\sigma_y)^2 + ({}^t\sigma_y - {}^t\sigma_z)^2 + ({}^t\sigma_z - {}^t\sigma_x)^2 + 6({}^t\tau_{xy}^2 + {}^t\tau_{yz}^2 + {}^t\tau_{zx}^2) \right]} \quad 4-33$$

$${}^t\eta = \frac{{}^tq}{{}^tp}$$

$${}^t\mathbf{s} = ({}^t\sigma_x - {}^t p \quad {}^t\sigma_y - {}^t p \quad {}^t\sigma_z - {}^t p \quad {}^t\tau_{xy} \quad {}^t\tau_{yz} \quad {}^t\tau_{zx})^T \quad 4-34$$

Укупне деформације и деформацијске инваријанте на почетку инкремента су:

$${}^t\boldsymbol{\varepsilon} = ({}^t\varepsilon_x \quad {}^t\varepsilon_y \quad {}^t\varepsilon_z \quad {}^t\gamma_{xy} \quad {}^t\gamma_{yz} \quad {}^t\gamma_{zx})^T \quad 4-35$$

$${}^t\varepsilon_v = {}^t\varepsilon_x + {}^t\varepsilon_y + {}^t\varepsilon_z \quad 4-36$$

$${}^t\varepsilon_q = \sqrt{\frac{2}{9}[({}^t\varepsilon_x - {}^t\varepsilon_y)^2 + ({}^t\varepsilon_y - {}^t\varepsilon_z)^2 + ({}^t\varepsilon_z - {}^t\varepsilon_x)^2] + \frac{1}{3}({}^t\gamma_{xy}^2 + {}^t\gamma_{yz}^2 + {}^t\gamma_{zx}^2)} \quad 4-37$$

$${}^t\boldsymbol{\varepsilon}' = \left({}^t\varepsilon_x - \frac{1}{3}{}^t\varepsilon_v \quad {}^t\varepsilon_y - \frac{1}{3}{}^t\varepsilon_v \quad {}^t\varepsilon_z - \frac{1}{3}{}^t\varepsilon_v \quad \frac{1}{2}{}^t\gamma_{xy} \quad \frac{1}{2}{}^t\gamma_{yz} \quad \frac{1}{2}{}^t\gamma_{zx} \right)^T \quad 4-38$$

Прираштаји деформација и инваријанте прираштаја деформација су:

$$\Delta\boldsymbol{\varepsilon} = (\Delta\varepsilon_x \quad \Delta\varepsilon_y \quad \Delta\varepsilon_z \quad \Delta\gamma_{xy} \quad \Delta\gamma_{yz} \quad \Delta\gamma_{zx})^T \quad 4-39$$

$$\Delta\varepsilon_v = \Delta\varepsilon_x + \Delta\varepsilon_y + \Delta\varepsilon_z \quad 4-40$$

$$\Delta\varepsilon_q = \sqrt{\frac{2}{9}[(\Delta\varepsilon_x - \Delta\varepsilon_y)^2 + (\Delta\varepsilon_y - \Delta\varepsilon_z)^2 + (\Delta\varepsilon_z - \Delta\varepsilon_x)^2] + \frac{1}{3}(\Delta\gamma_{xy}^2 + \Delta\gamma_{yz}^2 + \Delta\gamma_{zx}^2)} \quad 4-41$$

$$\Delta\boldsymbol{\varepsilon}' = \left(\Delta\varepsilon_x - \frac{1}{3}\Delta\varepsilon_v \quad \Delta\varepsilon_y - \frac{1}{3}\Delta\varepsilon_v \quad \Delta\varepsilon_z - \frac{1}{3}\Delta\varepsilon_v \quad \frac{1}{2}\Delta\gamma_{xy} \quad \frac{1}{2}\Delta\gamma_{yz} \quad \frac{1}{2}\Delta\gamma_{zx} \right)^T \quad 4-42$$

Укупне деформације и инваријанте укупних деформација на крају инкремента су:

$${}^{t+\Delta t}\boldsymbol{\varepsilon} = {}^t\boldsymbol{\varepsilon} + \Delta\boldsymbol{\varepsilon} = ({}^{t+\Delta t}\varepsilon_x \quad {}^{t+\Delta t}\varepsilon_y \quad {}^{t+\Delta t}\varepsilon_z \quad {}^{t+\Delta t}\gamma_{xy} \quad {}^{t+\Delta t}\gamma_{yz} \quad {}^{t+\Delta t}\gamma_{zx})^T \quad 4-43$$

$${}^{t+\Delta t}\varepsilon_v = {}^{t+\Delta t}\varepsilon_x + {}^{t+\Delta t}\varepsilon_y + {}^{t+\Delta t}\varepsilon_z \quad 4-44$$

$${}^{t+\Delta t}\varepsilon_q = \sqrt{\frac{2}{9}[({}^{t+\Delta t}\varepsilon_x - {}^{t+\Delta t}\varepsilon_y)^2 + ({}^{t+\Delta t}\varepsilon_y - {}^{t+\Delta t}\varepsilon_z)^2 + ({}^{t+\Delta t}\varepsilon_z - {}^{t+\Delta t}\varepsilon_x)^2] + \frac{1}{3}({}^{t+\Delta t}\gamma_{xy}^2 + {}^{t+\Delta t}\gamma_{yz}^2 + {}^{t+\Delta t}\gamma_{zx}^2)} \quad 4-45$$

Коефицијенти порозности и специфична запремина на почетку и крају инкремента су:

$${}^t e = (1 + e_0) \cdot \exp(-{}^t\varepsilon_v) - 1 \quad 4-46$$

$${}^t v = 1 + {}^t e \quad 4-47$$

$${}^{t+\Delta t} e = (1 + e_0) \cdot \exp(-{}^{t+\Delta t}\varepsilon_v) - 1 \quad 4-48$$

$${}^{t+\Delta t}v = 1 + {}^{t+\Delta t}e$$

Параметри стања за тренутну и имагинарну напонску тачку, степен изотропне преконсолидације и коефицијент ојачања на почетку инкремента су:

$${}^t\psi = {}^tv + \lambda \ln {}^tp' - \Gamma \quad 4-50$$

$${}^t\bar{\psi} = (\lambda - \kappa) \ln \left(\frac{2M^2}{M^2 + {}^t\eta^2} \right) \quad 4-51$$

$${}^tR = \exp \left(\frac{{}^t\bar{\psi} - {}^t\psi}{\lambda - \kappa} \right) \quad 4-52$$

$${}^t\omega = \left(1 + \frac{{}^t\bar{\psi} - {}^t\psi}{{}^t\bar{\psi}} \right) {}^tR \quad 4-53$$

Параметар ојачања на почетку инкремента је:

$${}^tp'_0 = \frac{{}^tq^2}{M^2 {}^tp'} + {}^tp' \quad 4-54$$

Запремински модул и модул смицања на почетку инкремента су:

$${}^tK = \frac{{}^tv {}^tp'}{\kappa} \quad 4-55$$

$${}^tG = \frac{3(1 + 2\mu)}{2(1 + \mu)} {}^tK \quad 4-56$$

Еластична конститутивна матрица на почетку инкремента која се користи за формирање матрице крутости коначних елемената је:

$${}^t\mathbf{D}'e = \begin{bmatrix} {}^tK' + \frac{4}{3} {}^tG & {}^tK' - \frac{2}{3} {}^tG & {}^tK' - \frac{2}{3} {}^tG & 0 & 0 & 0 \\ {}^tK' - \frac{2}{3} {}^tG & {}^tK' + \frac{4}{3} {}^tG & {}^tK' - \frac{2}{3} {}^tG & 0 & 0 & 0 \\ {}^tK' - \frac{2}{3} {}^tG & {}^tK' - \frac{2}{3} {}^tG & {}^tK' + \frac{4}{3} {}^tG & 0 & 0 & 0 \\ 0 & 0 & 0 & {}^tG & 0 & 0 \\ 0 & 0 & 0 & 0 & {}^tG & 0 \\ 0 & 0 & 0 & 0 & 0 & {}^tG \end{bmatrix} \quad 4-57$$

4.2.1.4 Алгоритам нумеричке интеграције

Поступак за интеграцију конститутивних релација HASP модела према методи водећег параметра приказан је кроз два алгоритма. Први алгоритам је општи алгоритам за тражење нуле водеће функције и рачунање напона и параметара стања на крају инкремента на основу познатих величина на почетку инкремента. Други алгоритам је алгоритам за прорачун напона и функције течења за познату вредност водећег

параметра. Други алгоритам се позива у оквиру првог алгоритма када год је потребно срачунати напоне за задату вредност водећег параметра.

Алгоритам 1. Нумеричка интеграција HASP модела методом водећег параметра

Корак 1. Познате величине на почетку инкремента: ${}^t\sigma$, ${}^t\varepsilon$, $\Delta\varepsilon$, ${}^{t+\Delta t}\varepsilon$, ${}^tD'e$ и све друге величине дате изразима (4-30)-(4-57).

Корак 2. Претпоставља се иницијална вредност водећег параметра ${}^{t+\Delta t}p'_i$ са којим се креће у тражење нуле функције. Као иницијална вредност претпоставља се вредност која одговара еластичном предвиђању:

$${}^{t+\Delta t}p'_i = {}^tp' \cdot \exp\left(\frac{{}^te - {}^{t+\Delta t}e}{\kappa}\right) \quad 4-58$$

Корак 3. Користећи **Алгоритам 2**, рачуна се ${}^{t+\Delta t}f_i({}^{t+\Delta t}p'_i)$, ${}^{t+\Delta t}f_i({}^{t+\Delta t}p'_i + \Delta p')$, ${}^{t+\Delta t}f_i({}^{t+\Delta t}p'_i - \Delta p')$. Три вредности функције течења су потребне да би могао да се срачуна први и други извод функције течења. Усвојено је $\Delta p' = 10^{-6} kPa$. Даље се рачуна ${}^{t+\Delta t}f'_i({}^{t+\Delta t}p'_i)$ преко израза (4-14) и ${}^{t+\Delta t}f''_i({}^{t+\Delta t}p'_i)$ преко израза (4-15).

Корак 4. Користи се Halley-ев поступак за тражење нуле. Следећа потенцијална нула ${}^{t+\Delta t}p'_{i+1}$ се добија преко израза (4-13). За нову вредност ${}^{t+\Delta t}p'_{i+1}$ проверава се да ли је задовољен услов:

$$|{}^{t+\Delta t}f_{i+1}({}^{t+\Delta t}p'_{i+1})| \leq FTOL \quad 4-59$$

Усвојено је $FTOL = 10^{-9}$. Уколико није задовољен услов (4-59), понављају се Корак 3 и Корак 4 за вредност ${}^{t+\Delta t}p'_{i+1}$ све док услов не буде задовољен, а максимално k_{max} пута. Усвојено је $k_{max} = 25$. С обзиром на тачност и брзину конвергенције Halley-евог поступка, након десетак итерација услов (4-59) бива задовољен.

Корак 5. Проверава се да ли је за нађену нулу ${}^{t+\Delta t}p'$ задовољен услов $\Delta\lambda \geq 0$. Уколико није задовољен тај услов, постоји могућност да је нађена „лажна нула“ и претпоставља се нова иницијална вредност ${}^{t+\Delta t}p'_i$ са којом се понављају Корак 3 и Корак 4. За нову вредност ${}^{t+\Delta t}p'_i$ може се усвојити вредност на почетку инкремента ${}^tp'$ или вредност ${}^{t+\Delta t}p'_i$ за коју се добија ${}^{t+\Delta t}p'_0 = {}^tp'_0$ (критично стање):

$${}^{t+\Delta t}p'_i = {}^tp_i \cdot \left(\frac{1}{\exp\left(\frac{({}^te - {}^{t+\Delta t}e) t\omega}{\lambda - \kappa}\right)} \right)^{\frac{\lambda - \kappa}{\kappa t\omega}} \quad 4-60$$

Уколико се и даље добија да је $\Delta\lambda < 0$, онда је понашање тла еластично и напони на крају инкремента се рачунају преко:

$${}^{t+\Delta t}\sigma = {}^t\sigma + {}^tD'e\Delta\varepsilon \quad 4-61$$

Корак 6. За срачунату вредност водећег параметра ${}^{t+\Delta t}p'$ преко **Алгоритма 2** рачунају се напони на крају инкремента ${}^{t+\Delta t}\sigma$ и све друге непознате величине у конфигурацији $t + \Delta t$.

Крај Алгоритма 1.

Алгоритам 2. Прорачун напона и функције течења у конфигурацији $t + \Delta t$ за познату вредност водећег параметра ${}^{t+\Delta t}p'$ у HASP моделу.

Корак 1. Срачуна се параметар ојачања ${}^{t+\Delta t}p'_0$ из израза (4-23).

Корак 2. Уколико је задовољен услов:

$$|2 {}^{t+\Delta t}p' - {}^{t+\Delta t}p'_0| \leq CSTOL \quad 4-62$$

Срачунава се пластични мултипликатор Δl из израза (4-28). Уколико није задовољен услов (4-62), срачунава се прво прираштај пластичних запреминских деформација $\Delta \varepsilon_v^p$ из израза (4-24), па затим пластични мултипликатор Δl из израза (4-19). Усвојено је $CSTOL = 10^{-6}$

Корак 3. Срачунава се девијаторски део тензора напона ${}^{t+\Delta t}\mathbf{s}$ из израза (4-25).

Корак 4. Срачунава се вредност функције течења ${}^{t+\Delta t}f({}^{t+\Delta t}p')$ преко израза (4-17). Вредност функције течења је потребна у **Алгоритму 1** за тражење нуле функције.

Корак 5. Срачунавају се напони ${}^{t+\Delta t}\sigma$ преко израза (4-29), као и све друге непознате величине.

Крај Алгоритма 2.

Параметри стања за тренутну напонску тачку и имагинарну напонску тачку, степен изотропне преконсолидације, коефицијент ојачања, запремински модул и модул смицања се рачунају на основу напонско-деформацијског стања на почетку инкремента. С обзиром да је метода водећег параметра имплицитна метода, исправније би било да се све те величине рачунају на основу напонско-деформацијског стања на крају инкремента. С обзиром да вредност тих величина на крају инкремента није унапред позната, због једноставности прорачуна, усвојено је да се у алгоритму нумеричке интеграције напона користе њихове вредности на почетку инкремента. Уколико је величина инкремената довољно мала, такав избор не би требало значајније да утиче на резултате прорачуна. Међутим, приликом преласка из контракције у експанзију код преконсолидованих глина, параметар ојачања ω драстично мења своју вредност ($\omega \rightarrow \infty$). У том случају требало би предвидети неку врсту поделе инкремента на мање делове, како би се адекватно узела у обзир промена параметра ојачања на посматраном инкременту.

4.2.2 Алгоритам (Vorja & Lee, 1990)

(Vorja & Lee, 1990) су развили имплицитни алгоритам за нумеричку интеграцију MCC модела. Овај алгоритам са извођењем свих потребних релација за интеграцију MCC

модела може се наћи у (Potts & Zdravković, 1999). У наставку је дат приказ овог алгоритма са одговарајућим изменама за интеграцију HASP модела.

У овом алгоритму први корак је еластично предвиђање у коме се одређују напони на крају инкремента ${}^{t+\Delta t}\boldsymbol{\sigma}^{tr}$ под претпоставком да нема пластичних деформација:

$${}^{t+\Delta t}\boldsymbol{\sigma}^{tr} = {}^t\boldsymbol{\sigma} + {}^t\mathbf{D}'^e\Delta\boldsymbol{\varepsilon} \quad 4-63$$

Иако у HASP моделу се напонска тачка увек налази на површи течења и на посматраном инкременту се јављају еластичне и пластичне деформације (сем у неким случајевима приликом растеређивања), овај корак је потребан јер се напонско стање на крају инкремента може одредити преко израза:

$${}^{t+\Delta t}\boldsymbol{\sigma} = {}^t\boldsymbol{\sigma} + {}^t\mathbf{D}'^e(\Delta\boldsymbol{\varepsilon} - \Delta\boldsymbol{\varepsilon}^p) = {}^{t+\Delta t}\boldsymbol{\sigma}^{tr} - {}^t\mathbf{D}'^e\Delta\boldsymbol{\varepsilon}^p \quad 4-64$$

У претходном изразу $\Delta\boldsymbol{\varepsilon}^p$ је вектор прираштаја пластичних деформација на посматраном инкременту који се може одредити из израза:

$$\Delta\boldsymbol{\varepsilon}^p = \Delta\lambda \frac{\partial P({}^{t+\Delta t}\boldsymbol{\sigma}, {}^{t+\Delta t}p'_0)}{\partial \boldsymbol{\sigma}} \quad 4-65$$

Основна формулација HASP модела користи асоцијативан закон течења:

$$P({}^{t+\Delta t}\boldsymbol{\sigma}, {}^{t+\Delta t}p'_0) = F({}^{t+\Delta t}\boldsymbol{\sigma}, {}^{t+\Delta t}p'_0) = \frac{{}^{t+\Delta t}q^2}{M^2} + {}^{t+\Delta t}p'({}^{t+\Delta t}p' - {}^{t+\Delta t}p'_0) = 0 \quad 4-66$$

Параметар ојачања на крају инкремента се одређује интеграцијом закона ојачања (3-8):

$${}^{t+\Delta t}p'_0 = {}^tp'_0 \cdot \exp\left(\frac{1 + {}^te}{\lambda - \kappa} {}^t\omega\Delta\varepsilon_v^p\right) \quad 4-67$$

Изводи пластичног потенцијала потребни за прорачун су:

$$\frac{\partial P({}^{t+\Delta t}\boldsymbol{\sigma}, {}^{t+\Delta t}p'_0)}{\partial p'} = \frac{\partial F({}^{t+\Delta t}\boldsymbol{\sigma}, {}^{t+\Delta t}p'_0)}{\partial p'} = 2 {}^{t+\Delta t}p' - {}^{t+\Delta t}p'_0 \quad 4-68$$

$$\frac{\partial P({}^{t+\Delta t}\boldsymbol{\sigma}, {}^{t+\Delta t}p'_0)}{\partial q} = \frac{\partial F({}^{t+\Delta t}\boldsymbol{\sigma}, {}^{t+\Delta t}p'_0)}{\partial q} = \frac{2 {}^{t+\Delta t}q}{M^2} \quad 4-69$$

$$\frac{\partial P({}^{t+\Delta t}\boldsymbol{\sigma}, {}^{t+\Delta t}p'_0)}{\partial p'_0} = \frac{\partial F({}^{t+\Delta t}\boldsymbol{\sigma}, {}^{t+\Delta t}p'_0)}{\partial p'_0} = - {}^{t+\Delta t}p' \quad 4-70$$

Прираштај пластичне запреминске деформације $\Delta\varepsilon_v^p$ је дат изразом (4-19), а прираштај пластичне дисторзијске деформације $\Delta\varepsilon_q^p$ дат је изразом (4-20). Средњи ефективни напон и девијаторски напон на крају инкремента могу се одредити из израза:

$${}^{t+\Delta t}p' = {}^{t+\Delta t}p'^{tr} - {}^tK\Delta\varepsilon_v^p = {}^{t+\Delta t}p'^{tr} - {}^tK\Delta\lambda(2 {}^{t+\Delta t}p' - {}^{t+\Delta t}p'_0) \quad 4-71$$

$${}^{t+\Delta t}q = {}^{t+\Delta t}q^{tr} - 3 {}^tG\Delta\varepsilon_q^p = {}^{t+\Delta t}q^{tr} - 3 {}^tG\Delta\lambda \frac{2 {}^{t+\Delta t}q}{M^2} \quad 4-72$$

У претходном изразу, ${}^{t+\Delta t}p^{tr}$ и ${}^{t+\Delta t}q^{tr}$ су средњи ефективни напон и девијаторски напон за напоне одређене у еластичном предвиђању ${}^{t+\Delta t}\sigma^{tr}$. Изрази (4-71) и (4-72) се могу написати и у следећем облику:

$${}^{t+\Delta t}p' = \frac{{}^{t+\Delta t}p^{tr} + {}^tK\Delta\Lambda {}^{t+\Delta t}p'_0}{1 + 2 {}^tK\Delta\Lambda} \quad 4-73$$

$${}^{t+\Delta t}q = \frac{{}^{t+\Delta t}q^{tr}}{1 + \frac{6 {}^tG\Delta\Lambda}{M^2}} \quad 4-74$$

Ако се у израз (4-67) убади (4-19) и (4-71), параметар ојачања на крају инкремента се онда може написати у следећем облику:

$${}^{t+\Delta t}p'_0 = {}^tp'_0 \cdot \exp\left(\frac{1 + {}^te}{\lambda - \kappa} {}^t\omega\Delta\Lambda \frac{2 {}^{t+\Delta t}p^{tr} - {}^{t+\Delta t}p'_0}{1 + 2 {}^tK\Delta\Lambda}\right) \quad 4-75$$

Величине дате изразима (4-73), (4-74) и (4-75) су изражене у функцији пластичног мултипликатора, који се може одредити из услова да је задовољен услов течења дат изразом (4-66), користећи Newton-ов поступак за тражење нуле функције:

$$\Delta\Lambda_{i+1} = \Delta\Lambda_i - \frac{F_i({}^{t+\Delta t}\sigma_i, {}^{t+\Delta t}p'_{0,i})}{\frac{\partial F_i({}^{t+\Delta t}\sigma_i, {}^{t+\Delta t}p'_{0,i})}{\partial \Delta\Lambda}} \quad 4-76$$

Извод функције течења по пластичном мултипликатору који је потребан у претходном изразу може се одредити преко:

$$\frac{\partial F_i({}^{t+\Delta t}\sigma_i, {}^{t+\Delta t}p'_{0,i})}{\partial \Delta\Lambda} = \frac{\partial F_i({}^{t+\Delta t}\sigma_i, {}^{t+\Delta t}p'_{0,i})}{\partial p'} \frac{\partial p'}{\partial \Delta\Lambda} + \frac{\partial F_i({}^{t+\Delta t}\sigma_i, {}^{t+\Delta t}p'_{0,i})}{\partial q} \frac{\partial q}{\partial \Delta\Lambda} + \quad 4-77$$

$$\frac{\partial F_i({}^{t+\Delta t}\sigma_i, {}^{t+\Delta t}p'_{0,i})}{\partial p'_0} \frac{\partial p'_0}{\partial \Delta\Lambda}$$

Парцијални изводи функције течења су дати изразима (4-68), (4-69) и (4-70). Детаљан поступак како се одређују изводи p' , q и p'_0 по $\Delta\Lambda$, као и коначан израз за извод функције течења по $\Delta\Lambda$ може се наћи у (Potts & Zdravković, 1999). Коначан облик израза (4-77) прилагођеног за HASP модел је:

$$\frac{\partial F_i({}^{t+\Delta t}\sigma_i, {}^{t+\Delta t}p'_{0,i})}{\partial \Delta\Lambda} = -(2 {}^{t+\Delta t}p' - {}^{t+\Delta t}p'_0) \frac{{}^tK(2 {}^{t+\Delta t}p' - {}^{t+\Delta t}p'_0) + \beta {}^{t+\Delta t}p' {}^{t+\Delta t}p'_0}{1 + \Delta\Lambda(2 {}^tK + \beta {}^{t+\Delta t}p'_0)} - \frac{2 {}^{t+\Delta t}q^2}{M^2 \left(\frac{M^2}{6 {}^tG} + \Delta\Lambda\right)} \quad 4-78$$

У претходном изразу са β је означено:

$$\beta = \frac{1 + {}^t e}{\lambda - \kappa} {}^t \omega \quad 4-79$$

Да би поступак приказан претходним изразима могао да се спроведе до краја, потребно је одредити ${}^{t+\Delta t} p'_0$ у изразу (4-75) које се налази са обе стране једнакости. Та једначина се може решити тако што се формира нова функција H :

$$H = {}^t p'_0 \cdot \exp\left(\frac{1 + {}^t e}{\lambda - \kappa} {}^t \omega \Delta \lambda \frac{2 {}^{t+\Delta t} p'^{tr} - {}^{t+\Delta t} p'_0}{1 + 2 {}^t K \Delta \lambda}\right) - {}^{t+\Delta t} p'_0 = 0 \quad 4-80$$

Нула функције H је ${}^{t+\Delta t} p'_0$ и може се наћи коришћењем Newton-овог поступка за тражење нуле функције:

$${}^{t+\Delta t} p'_{0,j+1} = {}^{t+\Delta t} p'_{0,j} - \frac{H({}^{t+\Delta t} p'_{0,j})}{\frac{\partial H({}^{t+\Delta t} p'_0)}{\partial {}^{t+\Delta t} p'_0}} \quad 4-81$$

Извод функције H по ${}^{t+\Delta t} p'_0$ је:

$$\frac{\partial H({}^{t+\Delta t} p'_0)}{\partial {}^{t+\Delta t} p'_0} = -\frac{1 + {}^t e}{\lambda - \kappa} {}^t \omega \Delta \lambda \cdot {}^t p'_0 \cdot \exp\left(\frac{1 + {}^t e}{\lambda - \kappa} {}^t \omega \Delta \lambda \frac{2 {}^{t+\Delta t} p'^{tr} - {}^{t+\Delta t} p'_0}{1 + 2 {}^t K \Delta \lambda}\right) - 1 \quad 4-82$$

4.2.2.1 Алгоритам нумеричке интеграције

Поступак за интеграцију конститутивних релација HASP модела према алгоритму (Ворја & Lee, 1990) приказан је кроз два алгоритма. У првом алгоритму тражи се вредност пластичног мултипликатора тако да буде задовољена функција течења. За срачунату вредност пластичног мултипликатора рачунају се напони и све друге непознате величине на крају инкремента. Други алгоритам налази вредност параметра ојачања ${}^{t+\Delta t} p'_0$ према изразима (4-80), (4-81) и (4-82). Други алгоритам се позива у оквиру првог алгоритма када је потребно срачунати параметар ојачања ${}^{t+\Delta t} p'_0$.

Приликом преласка из контракције у експанзију код преконсолидованих глина ($\omega \rightarrow \infty$) овај алгоритам неће успети да конвергира, односно неће моћи да се срачунају $\Delta \lambda$ и напони на крају инкремента. У том случају, како би се правазишла та ситуација, $\Delta \lambda$ ће се срачунати тако што ће се следећим величинама бити задате следеће вредности:

$${}^{t+\Delta t} p'_0 = 2 {}^{t+\Delta t} p'^{tr} \quad 4-83$$

$${}^{t+\Delta t} p' = {}^{t+\Delta t} p'^{tr} \quad 4-84$$

$${}^{t+\Delta t} q = M {}^{t+\Delta t} p' \quad 4-85$$

Пластични мултипликатор се онда може срачунати преко израза (4-28).

Алгоритам 3. Нумеричка интеграција HASP модела алгоритмом (Borja & Lee, 1990)

Корак 1. Познате величине на почетку инкремента: ${}^t\sigma$, ${}^t\epsilon$, $\Delta\epsilon$, ${}^{t+\Delta t}\epsilon$, ${}^tD'^e$ и све друге величине дате изразима (4-30)-(4-57).

Корак 2. Срачунавају се напони за еластично предвиђање ${}^{t+\Delta t}\sigma^{tr}$ преко израза (4-63), као и одговарајуће напонске инваријанте ${}^{t+\Delta t}p'^{tr}$ и ${}^{t+\Delta t}q^{tr}$.

Корак 3. Проверава се да ли је достигнуто прво карактеристично стање када долази до преласка из контракције у експанзију ($\omega \rightarrow \infty$) према услову:

$$|{}^t\omega| \geq \omega_{limit} \quad 4-86$$

Уколико је задовољен претходни услов, прелази се на Корак 8. У супротном, прелази се на Корак 4. Усвојено је $\omega_{limit} = 15000$.

Корак 4. Започиње се поступак тражења ΔL за коју је функција течења једнака нули. Као иницијална вредност ($i = 1$) узима се $\Delta L_1 = 0$.

Корак 5. Срачунава се ${}^{t+\Delta t}p'_{0,i}$ преко **Алгоритма 4**, ${}^{t+\Delta t}p'_i$ и ${}^{t+\Delta t}q_i$ преко израза (4-73) и (4-74).

Корак 6. Срачунава се вредност функције течења $F_i({}^{t+\Delta t}\sigma_i, {}^{t+\Delta t}p'_{0,i})$ према изразу (4-66) и проверава се да ли је задовољен услов:

$$|F_i({}^{t+\Delta t}\sigma_i, {}^{t+\Delta t}p'_{0,i})| \leq FTOL \quad 4-87$$

Усвојено је $FTOL = 10^{-9}$. Уколико је задовољен претходни услов, пронађена је вредност ΔL за коју је функција течења једнака нули и прелази се на Корак 9. У супротном прелази се на Корак 7.

Корак 7. Тражи се нова вредност пластичног мултипликатора ΔL_{i+1} преко израза (4-76). Са новом ΔL_{i+1} понавља се Корак 5 и Корак 6 ($i = i + 1$). Поступак се понавља k_{max} пута и усвојено је $k_{max} = 25$. Уколико је $i = k_{max}$ прелази се на Корак 9.

Корак 8. Долази до преласка из контракције у експанзију ($\omega \rightarrow \infty$). Срачунавају се ${}^{t+\Delta t}p'_{0,i}$, ${}^{t+\Delta t}p'_i$, ${}^{t+\Delta t}q_i$ и ΔL редом преко израза (4-83), (4-84), (4-85) и (4-28).

Корак 9. Проверава се да ли је срачуната вредност ΔL мања од нуле ($\Delta L < 0$). Уколико јесте, задаје се $\Delta L = 0$ и понашање тла је онда еластично.

Корак 10. Срачунавају се напони ${}^{t+\Delta t}\sigma$ на крају инкремента и завршава прорачун.

Крај Алгоритма 3.

Алгоритам 4. Прорачун ${}^{t+\Delta t}p'_0$

Корак 1. Иницијализација поступка прорачуна ${}^{t+\Delta t}p'_0$. Као иницијална вредност ($j = 1$) узима се ${}^{t+\Delta t}p'_{0,1} = {}^t p'_0$.

Корак 2. Рачуна се вредност функције $H_j(t+\Delta t p'_{0,j})$ преко израза (4-80).

Корак 3. Проверава се да ли је задовољен услов:

$$|H_j(t+\Delta t p'_{0,j})| \leq HTOL \quad 4-88$$

Усвојено је $HTOL = 10^{-6}$. Уколико је задовољен претходни услов, прорачун се завршава и иде се на Корак 5. У супротном иде се на Корак 4.

Корак 4. Тражи се нова вредност $t+\Delta t p'_{0,j+1}$ преко израза (4-81). Са новом вредношћу понавља се Корак 2 и Корак 3 ($j = j + 1$). Поступак се понавља k_{max} пута и усвојено је $k_{max} = 25$. Уколико је $j = k_{max}$ прелази се на Корак 5.

Корак 5. Добијена вредност за $t+\Delta t p'_0$ се користи у **Алгоритму 3** и завршава се прорачун.

Крај Алгоритма 4.

Из истог разлога као и у методи водећег параметра, параметри стања за тренутну напонску тачку и имагинарну напонску тачку, степен изотропне преконсолидације, коефицијент ојачања, запремински модул и модул смицања су срачунати на основу напонско-деформацијског стања на почетку инкремента и усвојено је да су све те величине константне на посматраном инкременту.

4.2.3 Модификовани Euler-ов поступак са аутоматском субинкрементацијом и контролом грешке

(Sloan, et al., 2001) су приказали експлицитни алгоритам за интеграцију релација МСС модела, који користи поделу задатог инкремента на мање инкременте (субинкременте). Величина субинкремента се одређује аутоматски тако што се процењује величина грешке у одређивању напона и параметара стања. Алгоритам одређује величину субинкремента тако да процењена грешка буде мања од неке унапред задате вредности. У наставку је приказан овај алгоритам са одговарајућим изменама за HASP модел.

Прираштаји напона и параметра ојачања на (суб)инкременту где се јављају еластичне и пластичне деформације може се одредити преко израза:

$$\Delta \sigma = D^{ep} \Delta \epsilon \quad 4-89$$

$$\Delta p'_0 = \Delta \Lambda B \quad 4-90$$

У претходном изразу, еласто-пластична конститутивна матрица D^{ep} се може одредити преко:

$$D^{ep} = D^{le} - \frac{D^{le} b a^T D^{le}}{A + a^T D^{le} b} \quad 4-91$$

Еластична конститутивна матрица се може срачунати преко израза (4-57) са запреминским модулом из (4-55) и модулом смицања из (4-56). Пластични мултипликатор $\Delta\lambda$ се може одредити преко израза:

$$\Delta\lambda = \frac{\mathbf{a}^T \mathbf{D}'^e \Delta\boldsymbol{\varepsilon}}{A + \mathbf{a}^T \mathbf{D}'^e \mathbf{b}} \quad 4-92$$

У претходним изразима вектори \mathbf{a} и \mathbf{b} садрже парцијалне изводе функција течења и пластичног потенцијала по напонима:

$$\mathbf{a} = \frac{\partial F}{\partial \boldsymbol{\sigma}} \quad 4-93$$

$$\mathbf{b} = \frac{\partial P}{\partial \boldsymbol{\sigma}} \quad 4-94$$

Пошто HASP модел користи асоцијативан закон течења онда је $\mathbf{a} = \mathbf{b}$. Величине A и B се могу одредити из израза:

$$A = - \frac{\partial F}{\partial p'_0} \frac{dp'_0}{d\Delta\lambda} \quad 4-95$$

$$B = - \frac{A}{\frac{\partial F}{\partial p'_0}} \quad 4-96$$

Ако се (4-91) и (4-92) убаце у (4-89) добија се:

$$\Delta\boldsymbol{\sigma} = \Delta\boldsymbol{\sigma}^e - \Delta\lambda \mathbf{D}^e \mathbf{b} \quad 4-97$$

уз

$$\Delta\boldsymbol{\sigma}^e = \mathbf{D}^e \Delta\boldsymbol{\varepsilon} \quad 4-98$$

Све величине (запремински модул tK , модул смицања tG , коефицијент ојачања ${}^t\omega$ вектори \mathbf{a} и \mathbf{b} , и величине A и B) које су потребне да би се срачунали напони и параметар ојачања на крају (суб)инкремента у претходним изразима се рачунају на основу напонско-деформацијског стања на почетку (суб)инкремента. Функција течења и пластични потенцијал на почетку инкремента су:

$$F({}^t\boldsymbol{\sigma}, {}^t p'_0) = P({}^t\boldsymbol{\sigma}, {}^t p'_0) = \frac{{}^t q^2}{M^2} + {}^t p'({}^t p' - {}^t p'_0) = 0 \quad 4-99$$

Парцијални изводи ове две функције по напонима су:

$$\frac{\partial F({}^t\boldsymbol{\sigma}, {}^t p'_0)}{\partial \boldsymbol{\sigma}} = \frac{\partial P({}^t\boldsymbol{\sigma}, {}^t p'_0)}{\partial \boldsymbol{\sigma}} = \frac{\partial F({}^t\boldsymbol{\sigma}, {}^t p'_0)}{\partial {}^t p'} \frac{\partial {}^t p'}{\partial {}^t \boldsymbol{\sigma}} + \frac{\partial F({}^t\boldsymbol{\sigma}, {}^t p'_0)}{\partial {}^t q} \frac{\partial {}^t q}{\partial {}^t \boldsymbol{\sigma}} \quad 4-100$$

Парцијални изводи у (4-100) су редом:

$$\frac{\partial F({}^t\sigma, {}^t p'_0)}{\partial {}^t p'} = 2 {}^t p' - {}^t p'_0 \quad 4-101$$

4-102

$$\frac{\partial F({}^t\sigma, {}^t p'_0)}{\partial {}^t q} = \frac{2 {}^t q}{M^2}$$

4-103

$$\frac{\partial {}^t p'}{\partial {}^t \sigma} = \left(\frac{1}{3} \quad \frac{1}{3} \quad \frac{1}{3} \quad 0 \quad 0 \quad 0 \right)^T$$

4-104

$$\frac{\partial {}^t q}{\partial {}^t \sigma} = \frac{3}{2 {}^t q} ({}^t \sigma'_x - {}^t p' \quad {}^t \sigma'_y - {}^t p' \quad {}^t \sigma'_z - {}^t p' \quad 2 {}^t \tau_{xy} \quad 2 {}^t \tau_{yz} \quad 2 {}^t \tau_{zx})^T$$

Коначни изрази за векторе **a** и **b** су:

$$\mathbf{a} = \mathbf{b} = \begin{bmatrix} \frac{2 {}^t p' - {}^t p'_0}{3} + 3 \frac{{}^t \sigma'_x - {}^t p'}{M^2} \\ \frac{2 {}^t p' - {}^t p'_0}{3} + 3 \frac{{}^t \sigma'_y - {}^t p'}{M^2} \\ \frac{2 {}^t p' - {}^t p'_0}{3} + 3 \frac{{}^t \sigma'_z - {}^t p'}{M^2} \\ \frac{6 {}^t \tau_{xy}}{M^2} \\ \frac{6 {}^t \tau_{yz}}{M^2} \\ \frac{6 {}^t \tau_{zx}}{M^2} \end{bmatrix} \quad 4-105$$

Да би се одредиле величине *A* и *B* потребно је наћи $\frac{dp'_0}{d\Delta\lambda}$. Закон ојачања (3-8) се може написати у облику:

$$dp'_0 = \frac{\nu\omega}{\lambda - \kappa} p'_0 d\varepsilon_v^p = \frac{\nu\omega}{\lambda - \kappa} p'_0 d\Delta\lambda \frac{\partial F}{\partial p'} \quad 4-106$$

Из (4-106) добија се:

$$\frac{dp'_0}{d\Delta\lambda} = \frac{\nu\omega}{\lambda - \kappa} p'_0 \frac{\partial F}{\partial p'} \quad 4-107$$

Величине *A* и *B* су онда:

$$A = - \frac{\partial F}{\partial p'_0} \frac{dp'_0}{d\Delta\lambda} = \frac{{}^t \nu {}^t \omega}{\lambda - \kappa} {}^t p' {}^t p'_0 (2 {}^t p' - {}^t p'_0) \quad 4-108$$

$$B = - \frac{A}{\frac{\partial F}{\partial p'_0}} = \frac{{}^t \nu {}^t \omega}{\lambda - \kappa} {}^t p'_0 (2 {}^t p' - {}^t p'_0) \quad 4-109$$

Код МСС модела, уколико се напонска тачка налази унутар површи течења на почетку инкремента потребно је одредити где ће путања напона пресећи површ течења у току деформисања на посматраном инкременту. Односно, део од прираштаја деформација на инкременту одговара потпуно еластичном понашању. На том делу, напонска путања која је кренула од напонске тачке унутар површи течења први пут додирује површ течења. Након тога, на остатку инкремента јављају се еластичне и пластичне деформације. У (Sloan, et al., 2001) дат је алгоритам преко кога може да се одреди када напонска путања долази до површи течења. Напонска тачка у HASP моделу се увек налази на површи течења и деформације су еластичне и пластичне тако да није потребно одређивати да ли је и када напонска путања пресекала површ течења.

У још једној ситуацији у МСС моделу може да се деси да на једном делу инкремента деформације буду само еластичне, а онда на остатку инкремента еластичне и пластичне. Уколико је напонска тачка на почетку инкремента била на површи течења, напонска путања може да буде усмерена унутар површи течења и да у неком тренутку током инкремента дође до и пресеке површи течења (еласто-пластично растерећење). У (Sloan, et al., 2001) је као и за случај описан у претходном пасусу, дат алгоритам како да се одреди колики део деформација на инкременту су само еластичне (напонска путања унутар површи течења), а колики део деформација су еластичне и пластичне. Пошто је у HASP моделу напонска тачка увек на површи течења и на инкременту су деформације увек еластичне и пластичне, тај алгоритам се неће користити. За претходна две методе нумеричке интеграције је у одређеним случајевима ипак узето да и у HASP моделу деформације могу бити еластичне. То је био случај када се добијало да је $\Delta l = 0$. И за експлицитне методе усвојиће се исто. Уколико се добије да је $\Delta l < 0$, ставља се $\Delta l = 0$. С обзиром да ће се у експлицитним методама користити субинкрементација, где се на сваком субинкременту одређује Δl , еласто-пластично растерећење би требало да буде аутоматски откривено. Део од укупног броја субинкремената би садржао само еластичне деформације, а остатак еластичне и пластичне деформације.

На крају сваког субинкремента у експлицитним методама, може се добити да напонска тачка не буде на површи течења:

$$F({}^{t+\Delta t}\boldsymbol{\sigma}, {}^{t+\Delta t}p'_0) > TOL \quad 4-110$$

У имплицитним методама се то не може десити, јер се напонско стање на крају инкремента тражи из услова да је напонска тачка на површи течења. У експлицитним методама тај услов није директно наметнут. Због тога је потребно спровести корекцију добијених напона и параметра ојачања на крају инкремента, како би напонска тачка била на површи течења. Тај поступак је дат у (Sloan, et al., 2001) и приказан је у наставку. Са ${}^{t+\Delta t}\boldsymbol{\sigma}^0$ и ${}^{t+\Delta t}p'_0{}^0$ су означени напони и параметар ојачања на крају (суб)инкремента, који нису на површи течења, односно за које важи (4-110). Са $\delta\boldsymbol{\sigma}$ и $\delta p'_0$ су означене мале корекције напона и параметра ојачања, тако да су нови напони и параметар ојачања који су сада ближи површи течења дати изразима:

$${}^{t+\Delta t}\boldsymbol{\sigma} = {}^{t+\Delta t}\boldsymbol{\sigma}^0 + \delta\boldsymbol{\sigma} \quad 4-111$$

$${}^{t+\Delta t}p'_0 = {}^{t+\Delta t}p'_0{}^0 + \delta p'_0 \quad 4-112$$

Изрази за кориговане напоне и параметар ојачања су:

$${}^{t+\Delta t}\boldsymbol{\sigma} = {}^{t+\Delta t}\boldsymbol{\sigma}^0 - \delta\Delta\lambda \mathbf{D}^{e,0} \mathbf{b}^0 \quad 4-113$$

$${}^{t+\Delta t}p'_0 = {}^{t+\Delta t}p'_0{}^0 + \delta\Delta\lambda B^0 \quad 4-114$$

У претходном изразу, $\delta\Delta\lambda$ је пластични мултипликатор за кориговане величине:

$$\delta\Delta\lambda = \frac{F^0({}^{t+\Delta t}\boldsymbol{\sigma}^0, {}^{t+\Delta t}p'_0{}^0)}{A^0 + \mathbf{a}^{0T} \mathbf{D}^{e,0} \mathbf{b}^0} \quad 4-115$$

Након што се одреде кориговани напони и параметар ојачања, треба срачунати поново $F({}^{t+\Delta t}\boldsymbol{\sigma}, {}^{t+\Delta t}p'_0)$. Уколико се добије да је:

$$|F({}^{t+\Delta t}\boldsymbol{\sigma}, {}^{t+\Delta t}p'_0)| > |F^0({}^{t+\Delta t}\boldsymbol{\sigma}^0, {}^{t+\Delta t}p'_0{}^0)| \quad 4-116$$

кориговани напони и параметар ојачања се налазе даље од површи течења него некориговани. Односно претходно описани поступак дивергира. У том случају, занемарује се претходна корекција и рачунају се поново кориговани напони и параметар ојачања према следећим изразима:

$$\delta\Delta\lambda = \frac{F^0({}^{t+\Delta t}\boldsymbol{\sigma}^0, {}^{t+\Delta t}p'_0{}^0)}{\mathbf{a}^{0T} \mathbf{a}^0} \quad 4-117$$

$${}^{t+\Delta t}\boldsymbol{\sigma} = {}^{t+\Delta t}\boldsymbol{\sigma}^0 - \delta\Delta\lambda \mathbf{a}^0 \quad 4-118$$

$${}^{t+\Delta t}p'_0 = {}^{t+\Delta t}p'_0{}^0 \quad 4-119$$

Уколико за кориговане напоне и параметар стања и даље важи (4-110) поступак се понавља. Еластична конститутивна матрица $\mathbf{D}^{e,0}$, коефицијент ојачања, вектори \mathbf{a}^0 и \mathbf{b}^0 и величине A^0 и B^0 одређују се из укупних деформација на крају (суб)инкремента и некоригованих напона ${}^{t+\Delta t}\boldsymbol{\sigma}^0$ и ${}^{t+\Delta t}p'_0{}^0$.

Релације које су претходно описане могу се применити на цео инкремент или се инкремент може поделити на мање субинкременте. Поделом на више субинкремената повећава се тачност нумеричке интеграције конститутивних релација, поготово код конститутивних модела са веома нелинеарним релацијама. У субинкрементацији уводи се нова величина - псеудо време T . На почетку инкремента (конфигурација t) $T = 0$, а на крају инкремента (конфигурација $t + \Delta t$) $T = 1$. Инкремент се дели на више субинкремената. На почетку n -тог субинкремента важи $T = T_{n-1}$, а на крају n -тог субинкремента важи $T = T_n = T_{n-1} + \Delta T_n$. Величина n -тог субинкремента изражена преко псеудо времена је ΔT_n и важи да је $0 < \Delta T_n \leq 1$. Величина субинкремента се одређује на основу процењене величине грешке (локална грешка у тачки нумеричке интеграције) у одређивању напона и параметра ојачања. За сваки субинкремент грешка се одређује из разлике резултата које даје модификован Euler-ов поступак другог реда и резултата које даје Euler-ов поступак првог реда. У зависности од процењене величине грешке, величина субинкремената варира током целог прорачуна. На почетку субинкремента, укупне деформације су $\boldsymbol{\varepsilon}_{n-1}$, а на крају субинкремента су $\boldsymbol{\varepsilon}_n$. Прираштај деформација на једном субинкременту је:

$$\Delta \boldsymbol{\varepsilon}_n = \boldsymbol{\varepsilon}_n - \boldsymbol{\varepsilon}_{n-1} = \Delta T_n \Delta \boldsymbol{\varepsilon} \quad 4-120$$

У претходном изразу $\Delta \boldsymbol{\varepsilon}$ је прираштај деформација на целом инкременту.

Напони $\boldsymbol{\sigma}_n$ и параметар ојачања $p'_{0,n}$ на крају једног субинкремента за Euler-ов поступак првог реда се одређују преко израза:

$$\boldsymbol{\sigma}_n = \boldsymbol{\sigma}_{n-1} + \Delta \boldsymbol{\sigma}_1 \quad 4-121$$

$$p'_{0,n} = p'_{0,n-1} + \Delta p'_{0,1} \quad 4-122$$

У претходним изразима, $\boldsymbol{\sigma}_{n-1}$ и $p'_{0,n-1}$ су напони и параметар ојачања на почетку субинкремента. Прираштаји напона $\Delta \boldsymbol{\sigma}_1$ и параметра ојачања $\Delta p'_{0,1}$ се одређују преко следећих израза на основу напонско-деформацијског стања на почетку субинкремента:

$$\Delta \boldsymbol{\sigma}_1 = \Delta \boldsymbol{\sigma}_1^e - \Delta \Lambda_1 \mathbf{D}_1^e \mathbf{b}_1 \quad 4-123$$

$$\Delta p'_{0,1} = \Delta \Lambda_1 B_1 \quad 4-124$$

$$\Delta \boldsymbol{\sigma}_1^e = \mathbf{D}_1^{e'} \Delta \boldsymbol{\varepsilon}_n \quad 4-125$$

$$\Delta \Lambda_1 = \frac{\mathbf{a}_1^T \mathbf{D}_1^{e'} \Delta \boldsymbol{\varepsilon}_n}{A_1 + \mathbf{a}_1^T \mathbf{D}_1^{e'} \mathbf{b}_1} \quad 4-126$$

$$A_1 = \frac{v_{n-1} \omega_{n-1}}{\lambda - \kappa} p'_{n-1} p'_{0,n-1} (2p'_{n-1} - p'_{0,n-1}) \quad 4-127$$

$$B_1 = \frac{v_{n-1} \omega_{n-1}}{\lambda - \kappa} p'_{0,n-1} (2p'_{n-1} - p'_{0,n-1}) \quad 4-128$$

$$\mathbf{a}_1 = \frac{\partial F(\boldsymbol{\sigma}_{n-1}, p'_{0,n-1})}{\partial \boldsymbol{\sigma}} \quad 4-129$$

$$\mathbf{b}_1 = \frac{\partial P(\boldsymbol{\sigma}_{n-1}, p'_{0,n-1})}{\partial \boldsymbol{\sigma}} \quad 4-130$$

$$\mathbf{D}_1^{e'} = \mathbf{D}^{e'}(\boldsymbol{\sigma}_{n-1}, e_{n-1}) \quad 4-131$$

Претходни изрази (4-123) – (4-131) могу да се представе у скраћеном облику:

$$\Delta \boldsymbol{\sigma}_1 = \mathbf{D}^{ep}(\boldsymbol{\sigma}_{n-1}, p'_{0,n-1}, e_{n-1}, \omega_{n-1}) \Delta \boldsymbol{\varepsilon}_n \quad 4-132$$

$$\Delta p'_{0,1} = \Delta \Lambda_1(\boldsymbol{\sigma}_{n-1}, p'_{0,n-1}, e_{n-1}, \omega_{n-1}, \Delta \boldsymbol{\varepsilon}_n) B(\boldsymbol{\sigma}_{n-1}, p'_{0,n-1}, e_{n-1}, \omega_{n-1}) \quad 4-133$$

У претходном изразима v_{n-1} је специфична запремина на почетку субинкремента, ω_{n-1} је коефицијент ојачања на почетку субинкремента, e_{n-1} је коефицијент порозности на почетку субинкремента.

Напони $\hat{\boldsymbol{\sigma}}_n$ и параметар ојачања $\hat{p}'_{0,n-1}$ на крају једног субинкремента за модификовани Euler-ов поступак другог реда се одређују преко израза:

$$\hat{\sigma}_n = \sigma_{n-1} + \frac{1}{2}(\Delta\sigma_1 + \Delta\sigma_2) \quad 4-134$$

$$\hat{p}'_{0,n} = p'_{0,n-1} + \frac{1}{2}(\Delta p'_{0,1} + \Delta p'_{0,2}) \quad 4-135$$

Прираштаји $\Delta\sigma_1$ и $\Delta p'_{0,1}$ су дати изразима (4-123 – (4-150). Прираштаји напона $\Delta\sigma_2$ и параметра ојачања $\Delta p'_{0,2}$ одређују се на основу напонско-деформацијског стања добијеног из Euler-овог поступка првог реда σ_n и $p'_{0,n}$:

$$\Delta\sigma_2 = \Delta\sigma_2^e - \Delta\Lambda_2 \mathbf{D}_2^e \mathbf{b}_2 \quad 4-136$$

$$\Delta p'_{0,2} = \Delta\Lambda_2 B_2 \quad 4-137$$

$$\Delta\sigma_2^e = \mathbf{D}_2'^e \Delta\epsilon_n \quad 4-138$$

$$\Delta\Lambda_2 = \frac{\mathbf{a}_2^T \mathbf{D}_2'^e \Delta\epsilon_n}{A_2 + \mathbf{a}_2^T \mathbf{D}_2'^e \mathbf{b}_2} \quad 4-139$$

$$A_2 = \frac{v_n \omega_n}{\lambda - \kappa} p'_n p'_{0,n} (2p'_n - p'_{0,n}) \quad 4-140$$

$$B_2 = \frac{v_n \omega_n}{\lambda - \kappa} p'_{0,n} (2p'_n - p'_{0,n}) \quad 4-141$$

$$\mathbf{a}_2 = \frac{\partial F(\sigma_n, p'_{0,n})}{\partial \sigma} \quad 4-142$$

$$\mathbf{b}_2 = \frac{\partial P(\sigma_n, p'_{0,n})}{\partial \sigma} \quad 4-143$$

$$\mathbf{D}_2'^e = \mathbf{D}'^e(\sigma_n, e_n) \quad 4-144$$

$$\omega_n = \omega(\sigma_n, e_n) \quad 4-145$$

Претходни изрази (4-136) – (4-164) могу да се представе у скраћеном облику:

$$\Delta\sigma_2 = \mathbf{D}^{ep}(\sigma_{n-1} + \Delta\sigma_1, p'_{0,n-1} + \Delta p'_{0,1}, e_n, \omega_n) \Delta\epsilon_n \quad 4-146$$

$$\Delta p'_{0,2} = \Delta\Lambda_2(\sigma_{n-1} + \Delta\sigma_1, p'_{0,n-1} + \Delta p'_{0,1}, e_n, \omega_n, \Delta\epsilon_n) B(\sigma_{n-1} + \Delta\sigma_1, p'_{0,n-1} + \Delta p'_{0,1}, e_n, \omega_n) \quad 4-147$$

У претходном изразу e_n је коефицијент порозности на крају субинкремента и v_n је специфична запремина на крају субинкремента.

Локална грешка у одређивању напона и параметра ојачања се одређује из разлике резултата модификованог Euler-овог поступка и Euler-овог поступка:

$$\begin{Bmatrix} \{\hat{\sigma}_n\} \\ \{\hat{p}'_{0,n}\} \end{Bmatrix} - \begin{Bmatrix} \{\sigma_n\} \\ \{p'_{0,n}\} \end{Bmatrix} = \begin{Bmatrix} \frac{1}{2}(\Delta\sigma_2 - \Delta\sigma_1) \\ \frac{1}{2}(\Delta p'_{0,2} - \Delta p'_{0,1}) \end{Bmatrix} \quad 4-148$$

Релативна грешка се онда одређује преко израза:

$$Er_n = \frac{1}{2} \cdot \max \left\{ \frac{\|(\Delta\sigma_2 - \Delta\sigma_1)\|}{\|\hat{\sigma}_n\|}, \frac{|\Delta p'_{0,2} - \Delta p'_{0,1}|}{\hat{p}'_{0,n}} \right\} \quad 4-149$$

У претходном изразу $\|\mathbf{r}\|$ представља интезитет вектора $\mathbf{r} = (r_1 \ r_2 \ \dots \ r_{k-1} \ r_k)^T$:

$$\|\mathbf{r}\| = \sqrt{r_1^2 + r_2^2 + \dots + r_{k-1}^2 + r_k^2} \quad 4-150$$

Уколико је Er_n мање од усвојене толеранције $STOL$, субинкремент се прихвата. У супротном, субинкремент се одбацује. Уколико се субинкремент одбацује, потребно је смањити псеудо време ΔT_n према изразу:

$$\Delta T_{n,novo} = \max\{\rho\Delta T_{n,staro}, \Delta T_{min}\} \quad 4-151$$

Уз

$$\rho = \max \left\{ 0.9 \sqrt{\frac{STOL}{Er_n}}, 0.1 \right\} \quad 4-152$$

прорачун се сада понавља са мањом величином субинкремента. Уколико се субинкремент прихвата, величина наредног субинкремента се одређује према изразу:

$$\Delta T_{n+1} = \max\{\rho\Delta T_n, \Delta T_{min}\} \quad 4-153$$

уз

$$\rho = \min \left\{ 0.9 \sqrt{\frac{STOL}{Er_n}}, 1.1 \right\} \quad 4-154$$

Уколико у претходном покушају субинкремент није успео, онда се додатно ограничава раст наредног субинкремента:

$$\rho = \min \left\{ 0.9 \sqrt{\frac{STOL}{Er_n}}, 1.0 \right\} \quad 4-155$$

Такође, треба водити рачуна да $T_{n+1} \leq 1$, као и да величина субинкремента не буде мања од минимално прописане вредности ΔT_{min} . На крају успешног субинкремента, као решење прихватају се напони и параметар ојачања за модификовани Euler-ов поступак другог реда. Након тога, проверава се да ли срачунати напони и параметар ојачања

задовољавају услов течења и врши се корекција напона и параметра ојачања враћањем на површ течења уколико је то потребно. Прорачун се понавља за све субинкременте док се не достигне псеудо време $T = 1$. Током прорачуна узастопни субинкременти могу да буду исти, да се повећавају или да се смањују, у зависности од срачунате релативне грешке.

4.2.3.1 Алгоритам нумеричке интеграције

Алгоритам 5. Нумеричка интеграција HASP модела модификованим Euler-овим поступком са аутоматском субинкрементацијом

Корак 1. Познате величине на почетку инкремента: ${}^t\sigma$, ${}^t\epsilon$, $\Delta\epsilon$, ${}^{t+\Delta t}\epsilon$, ${}^tD^{e}$ и све друге величине дате изразима (4-30)-(4-57).

Корак 2. Иницијализација субинкрементације: $T = T_0 = 0$ и $\Delta T_1 = 1$.

Корак 3. Срачунавају се: $\Delta\epsilon_n$ из (4-120), $\Delta\sigma_1$ и $\Delta p'_{0,1}$ из (4-123) - (4-131), $\Delta\sigma_2$ и $\Delta p'_{0,2}$ из (4-136) - (4-145). Уколико се при прорачуну $\Delta\sigma_1$ и $\Delta p'_{0,1}$ добије $\Delta\lambda_1 < 0$ усваја се $\Delta\lambda_1 = 0$ (деформације су еластичне). Уколико се при прорачуну $\Delta\sigma_2$ и $\Delta p'_{0,2}$ добије $\Delta\lambda_2 < 0$ усваја се $\Delta\lambda_2 = 0$ (деформације су еластичне).

Корак 4. Срачунавају се напони $\hat{\sigma}_n$ из (4-134) и параметар ојачања $\hat{p}'_{0,n}$ из (4-135) на крају субинкремента.

Корак 5. Срачунава се релативна грешка Er_n из (4-149).

Корак 6. Уколико је $Er_n > STOL$ субинкремент се одбацује и потребно је смањити псеудо време ΔT_n . Усвојено је $STOL = 10^{-4}$. Прелази се на Корак 7.

Уколико је $Er_n \leq STOL$ субинкремент се прихвата и иде се на Корак 8.

Корак 7. Потребно је смањити величину субинкремента ΔT_n према (4-151) и (4-152). Понавља се прорачун са новом величином субинкремента. Враћа се на Корак 3.

Корак 8. Пошто је субинкремент био успешан, прихватају се срачунати напони и параметар ојачања. На крају субинкремента напони и параметар ојачања су:

$$\sigma_{T_n} = \hat{\sigma}_n \quad 4-156$$

$$p'_{0,T_n} = \hat{p}'_{0,n} \quad 4-157$$

Корак 9. Проверава се да ли су напони и параметар ојачања на крају субинкремента на површи течења према:

$$|F(\sigma_{T_n}, p'_{0,T_n})| \leq FTOL \quad 4-158$$

Усвојено је $FTOL = 10^{-9}$. Уколико је испуњен претходни услов, напони и параметар ојачања су на површи течења и не мора да се врши корекција. Прелази се на Корак 10.

У случају да су $\Delta\lambda_1 = 0$ или $\Delta\lambda_2 = 0$ понашање тла је еластично и не треба да се врши корекција.

Уколико није испуњен услов (4-158) потребно је да се изврши корекција напона и параметра ојачања. Корекција се врши преко **Алгоритма 6**.

Корак 10. Одређује се величина наредног субинкремента ΔT_{n+1} преко израза (4-153), (4-154) и (4-155).

Корак 11. Одређује се укупно псеудо време на крају субинкремента:

$$T_n = T_{n-1} + \Delta T_n \quad 4-159$$

Корак 12. Проверава се да ли је величина наредног субинкремента ΔT_{n+1} мања од минимално прописане вредности ΔT_{min} . Уколико јесте мања онда је:

$$\Delta T_{n+1} = \Delta T_{min} \quad 4-160$$

Усвојено је $\Delta T_{min} = 10^{-6}$.

Корак 13. Проверава се да укупно псеудо време на крају наредног субинкремента не буде веће од 1:

$$T_{n+1} = T_n + \Delta T_{n+1} \leq 1 \quad 4-161$$

Уколико није задовољен претходни услов, онда је величина наредног субинкремента:

$$\Delta T_{n+1} = 1 - T_n \quad 4-162$$

Корак 14. Уколико је $T_n < 1$ започиње се наредни субинкремент и понављају се Кораци 3-13. У супротном, нанет је цео инкремент и срачунати су напони ${}^{t+\Delta t}\sigma$ и параметар ојачања ${}^{t+\Delta t}p'_0$ на крају инкремента. Завршава се прорачун.

Крај Алгоритма 5.

Алгоритам 6. Корекција напона и параметра ојачања – враћање на површ течења.

Корак 1. Иницијализација прорачуна. Коригују се напони ${}^{t+\Delta t}\sigma^0$ и параметар ојачања ${}^{t+\Delta t}p'_0$.

Корак 2. Срачунава се $\delta\Delta l$ из (4-115) и кориговани ${}^{t+\Delta t}\sigma$ из (4-113) и ${}^{t+\Delta t}p'_0$ из (4-114).

Корак 3. Проверава се да ли је испуњен услов (4-116). Уколико јесте, одбацује се корекција из Корак 2 и прелази се на Корак 4. Уколико није испуњен услов, кориговани ${}^{t+\Delta t}\sigma$ и ${}^{t+\Delta t}p'_0$ се прихватају и прелази се на Корак 5.

Корак 4. Срачунава се $\delta\Delta l$ из (4-117) и кориговани ${}^{t+\Delta t}\sigma$ из (4-118) и ${}^{t+\Delta t}p'_0$ из (4-119).

Корак 5. Проверава се да ли су кориговани ${}^{t+\Delta t}\sigma$ и ${}^{t+\Delta t}p'_0$ на површи течења према (4-158). Уколико јесу, прелази се на Корак 7. Уколико нису, прелази се на Корак 6.

Корак 6. Потребно је поновити корекцију ${}^{t+\Delta t}\sigma$ и ${}^{t+\Delta t}p'_0$:

$${}^{t+\Delta t}\sigma^0 = {}^{t+\Delta t}\sigma \quad 4-163$$

$${}^{t+\Delta t}p'_0 = {}^{t+\Delta t}p'_0$$

4-164

Поступак корекције се понавља. Враћа се на Корак 2. Корази 2-6 се понављају максимално k_{max} пута. Усвојено $k_{max} = 10$.

Корак 7. Прихватају се кориговани ${}^{t+\Delta t}\sigma$ и ${}^{t+\Delta t}p'_0$ и завршава се прорачун.

Крај Алгоритма 6.

4.2.4 Runge-Kutta-Dormand-Prince поступак са аутоматском субинкрементацијом и контролом грешке

Поступак нумеричке интеграције експлицитним RKDP поступком је сличан модификованом Euler-овом поступку који је описан у претходном потпоглављу. У RKDP поступку користе се решења четвртог и петог реда у нумеричкој интеграцији напона и параметра ојачања. Тачност добијених напона и параметра ојачања у овом поступку је већа него у другим методама и због тога се овај поступак често користи за проверу тачности метода нумеричке интеграције нижег реда (Sloan, et al., 2001). RKDP поступак са аутоматском субинкрементацијом и контролом грешке за МСС модел дат је у (Sloan, et al., 2001) и биће приказан у наставку са одговарајућим изменама за HASP модел. Математичка формулација RKDP интеграције четвртог и петог реда се може наћи у (Dormand & Prince, 1980). Приликом прорачуна решења вишег реда требало би на одговарајући начин узети промену коефицијента порозности e и коефицијента ојачања ω током нумеричке интеграције на једном субинкременту. Варијација e и ω утиче на прираштаје напона и параметра ојачања преко D^e и величина A и B . У (Lloret-Cabot, et al., 2016) је описано како се узима у обзир утицај варијације e у нумеричкој интеграцији у поступку RKDP.

Као и у претходној методи нумеричке интеграције, уводи се величина псеудо време T . На почетку инкремента важи $T = 0$, а на крају инкремента $T = 1$. На почетку n -тог субинкремента важи $T = T_{n-1}$, а на крају истог субинкремента $T = T_n$. Величина n -тог субинкремента је $\Delta T_n = T_n - T_{n-1}$ и варира током инкремента у зависности од процењене величине локалне грешке у одређивању напона и параметра ојачања. За сваки субинкремент локална грешка се одређује из разлике резултата решења петог и четвртог реда RKDP поступка нумеричке интеграције.

На почетку једног субинкремента напони су σ_{n-1} , параметар ојачања је $p'_{0,n-1}$, коефицијент порозности је e_{n-1} , специфична запремина је v_{n-1} , коефицијент ојачања је ω_{n-1} а укупне деформације су ϵ_{n-1} . Прираштај деформација на n -том субинкременту $\Delta\epsilon_n$ је дат у (4-120), а прираштај запреминске деформације који одговара $\Delta\epsilon_n$ је $\Delta\epsilon_{v,n}$. Решење четвртог реда се рачуна преко израза:

$$\sigma_n = \sigma_{n-1} + \frac{31}{540}\Delta\sigma_1 + \frac{190}{297}\Delta\sigma_3 - \frac{145}{108}\Delta\sigma_4 + \frac{351}{220}\Delta\sigma_5 + \frac{1}{20}\Delta\sigma_6 \quad 4-165$$

4-166

$$p'_{0,n} = p'_{0,n-1} + \frac{31}{540}\Delta p'_{0,1} + \frac{190}{297}\Delta p'_{0,3} - \frac{145}{108}\Delta p'_{0,4} + \frac{351}{220}\Delta p'_{0,5} + \frac{1}{20}\Delta p'_{0,6}$$

Решење петог реда се рачуна преко израза:

$$\hat{\sigma}_n = \sigma_{n-1} + \frac{19}{216} \Delta \sigma_1 + \frac{1000}{2079} \Delta \sigma_3 - \frac{125}{216} \Delta \sigma_4 + \frac{81}{88} \Delta \sigma_5 + \frac{5}{56} \Delta \sigma_6 \quad 4-167$$

$$\hat{p}'_{0,n} = p'_{0,n-1} + \frac{19}{216} \Delta p'_{0,1} + \frac{1000}{2079} \Delta p'_{0,3} - \frac{125}{216} \Delta p'_{0,4} + \frac{81}{88} \Delta p'_{0,5} + \frac{5}{56} \Delta p'_{0,6} \quad 4-168$$

Прираштај напона $\Delta \sigma_i$ и параметра ојачања $\Delta p'_{0,i}$ ($i = 1, 2, 3, 4, 5, 6$) се рачунају преко израза:

$$\Delta \sigma_i = \Delta \sigma_i^e - \Delta \Lambda_i \mathbf{D}'_i \mathbf{b}_i \quad 4-169$$

$$\Delta p'_{0,i} = \Delta \Lambda_i B_i \quad 4-170$$

$$\Delta \sigma_i^e = \mathbf{D}'_i{}^e \Delta \boldsymbol{\varepsilon}_n \quad 4-171$$

$$\Delta \Lambda_i = \frac{\mathbf{a}_i^T \mathbf{D}'_i{}^e \Delta \boldsymbol{\varepsilon}_n}{A_i + \mathbf{a}_i^T \mathbf{D}'_i{}^e \mathbf{b}_i} \quad 4-172$$

$$A_i = \frac{\tilde{v}_i \tilde{\omega}_i}{\lambda - \kappa} \tilde{p}'_i \tilde{p}'_{0,i} (2\tilde{p}'_i - \tilde{p}'_{0,i}) \quad 4-173$$

$$B_i = \frac{\tilde{v}_i \tilde{\omega}_i}{\lambda - \kappa} \tilde{p}'_{0,i} (2\tilde{p}'_i - \tilde{p}'_{0,i}) \quad 4-174$$

$$\mathbf{a}_i = \frac{\partial F(\tilde{\sigma}_i, \tilde{p}'_{0,i})}{\partial \boldsymbol{\sigma}} \quad 4-175$$

$$\mathbf{b}_i = \frac{\partial P(\tilde{\sigma}_i, \tilde{p}'_{0,i})}{\partial \boldsymbol{\sigma}} \quad 4-176$$

$$\mathbf{D}'_i{}^e = \mathbf{D}'^e(\tilde{\sigma}_i, \tilde{e}_i) \quad 4-177$$

Изрази (4-169) – (4-177) могу да се представе у скраћеном облику:

$$\Delta \sigma_i = \mathbf{D}^{ep}(\tilde{\sigma}_i, \tilde{p}'_{0,i}, \tilde{e}_i, \tilde{\omega}_i) \Delta \boldsymbol{\varepsilon}_n \quad 4-178$$

$$\Delta p'_{0,i} = \Delta \Lambda_i(\tilde{\sigma}_i, \tilde{p}'_{0,i}, \tilde{e}_i, \tilde{\omega}_i, \Delta \boldsymbol{\varepsilon}_n) B(\tilde{\sigma}_i, \tilde{p}'_{0,i}, \tilde{e}_i, \tilde{\omega}_i) \quad 4-179$$

Све величине у претходним изразима са тилдом изнад (\sim) су међувредности у поступку нумеричке интеграције, потребне за одређивање напона и параметра ојачања на крају субинкремента. У изразима (4-173) и (4-174) \tilde{p}'_i је средњи ефективни напон за $\tilde{\sigma}_i$, а \tilde{v}_i је специфична запремина:

$$\tilde{v}_i = 1 + \tilde{e}_i \quad 4-180$$

Прираштај напона $\Delta \sigma_1$ и параметра ојачања $\Delta p'_{0,1}$ се рачунају према изразима:

$$\Delta \sigma_1 = \mathbf{D}^{ep}(\tilde{\sigma}_1, \tilde{p}'_{0,1}, \tilde{e}_1, \tilde{\omega}_1) \Delta \boldsymbol{\varepsilon}_n \quad 4-181$$

4-182

$$\Delta p'_{0,1} = \Delta \Lambda_1(\tilde{\sigma}_1, \tilde{p}'_{0,1}, \tilde{e}_1, \tilde{\omega}_1, \Delta \varepsilon_n) B(\tilde{\sigma}_1, \tilde{p}'_{0,1}, \tilde{e}_1, \tilde{\omega}_1)$$

4-183

$$\tilde{\sigma}_1 = \sigma_{n-1}$$

4-184

$$\tilde{p}'_{0,1} = p'_{0,n-1}$$

4-185

$$\tilde{e}_1 = e_{n-1}$$

4-186

$$\tilde{\omega}_1 = \omega_{n-1}$$

Прираштај напона $\Delta \sigma_2$ и параметра ојачања $\Delta p'_{0,2}$ се рачунају преко израза:

4-187

$$\Delta \sigma_2 = \mathbf{D}^{ep}(\tilde{\sigma}_2, \tilde{p}'_{0,2}, \tilde{e}_2, \tilde{\omega}_2) \Delta \varepsilon_n$$

4-188

$$\Delta p'_{0,2} = \Delta \Lambda_2(\tilde{\sigma}_2, \tilde{p}'_{0,2}, \tilde{e}_2, \tilde{\omega}_2, \Delta \varepsilon_n) B(\tilde{\sigma}_2, \tilde{p}'_{0,2}, \tilde{e}_2, \tilde{\omega}_2)$$

4-189

$$\tilde{\sigma}_2 = \sigma_{n-1} + \frac{1}{5} \Delta \sigma_1$$

4-190

$$\tilde{p}'_{0,2} = p'_{0,n-1} + \frac{1}{5} \Delta p'_{0,1}$$

4-191

$$\tilde{e}_2 = (1 + e_{n-1}) \cdot \exp\left(-\frac{1}{5} \Delta \varepsilon_{v,n}\right) - 1$$

4-192

$$\tilde{\omega}_2 = \omega(\tilde{e}_2, \tilde{\sigma}_2)$$

Прираштај напона $\Delta \sigma_3$ и параметра ојачања $\Delta p'_{0,3}$ се рачунају преко израза:

4-193

$$\Delta \sigma_3 = \mathbf{D}^{ep}(\tilde{\sigma}_3, \tilde{p}'_{0,3}, \tilde{e}_3, \tilde{\omega}_3) \Delta \varepsilon_n$$

4-194

$$\Delta p'_{0,3} = \Delta \Lambda_3(\tilde{\sigma}_3, \tilde{p}'_{0,3}, \tilde{e}_3, \tilde{\omega}_3, \Delta \varepsilon_n) B(\tilde{\sigma}_3, \tilde{p}'_{0,3}, \tilde{e}_3, \tilde{\omega}_3)$$

4-195

$$\tilde{\sigma}_3 = \sigma_{n-1} + \frac{3}{40} \Delta \sigma_1 + \frac{9}{40} \Delta \sigma_2$$

4-196

$$\tilde{p}'_{0,3} = p'_{0,n-1} + \frac{3}{40} \Delta p'_{0,1} + \frac{9}{40} \Delta p'_{0,2}$$

4-197

$$\tilde{e}_3 = (1 + e_{n-1}) \cdot \exp\left(-\left(\frac{3}{40} + \frac{9}{40}\right) \Delta \varepsilon_{v,n}\right) - 1$$

4-198

$$\tilde{\omega}_3 = \omega(\tilde{e}_3, \tilde{\sigma}_3)$$

Прираштај напона $\Delta \sigma_4$ и параметра ојачања $\Delta p'_{0,4}$ се рачунају преко израза:

4-199

$$\Delta \sigma_4 = \mathbf{D}^{ep}(\tilde{\sigma}_4, \tilde{p}'_{0,4}, \tilde{e}_4, \tilde{\omega}_4) \Delta \varepsilon_n$$

4-200

$$\Delta p'_{0,4} = \Delta \Lambda_4(\tilde{\sigma}_4, \tilde{p}'_{0,4}, \tilde{e}_4, \tilde{\omega}_4, \Delta \varepsilon_n) B(\tilde{\sigma}_4, \tilde{p}'_{0,4}, \tilde{e}_4, \tilde{\omega}_4)$$

$$\tilde{\sigma}_4 = \sigma_{n-1} + \frac{3}{10}\Delta\sigma_1 - \frac{9}{10}\Delta\sigma_2 + \frac{6}{5}\Delta\sigma_3 \quad 4-201$$

$$\tilde{p}'_{0,4} = p'_{0,n-1} + \frac{3}{10}\Delta p'_{0,1} - \frac{9}{10}\Delta p'_{0,2} + \frac{6}{5}\Delta p'_{0,3} \quad 4-202$$

$$\tilde{e}_4 = (1 + e_{n-1}) \cdot \exp\left(-\left(\frac{3}{10} - \frac{9}{10} + \frac{6}{5}\right)\Delta\varepsilon_{v,n}\right) - 1 \quad 4-203$$

$$\tilde{\omega}_4 = \omega(\tilde{e}_4, \tilde{\sigma}_4) \quad 4-204$$

Прираштај напона $\Delta\sigma_5$ и параметра ојачања $\Delta p'_{0,5}$ се рачунају преко израза:

$$\Delta\sigma_5 = \mathbf{D}^{ep}(\tilde{\sigma}_5, \tilde{p}'_{0,5}, \tilde{e}_5, \tilde{\omega}_5)\Delta\varepsilon_n \quad 4-205$$

$$\Delta p'_{0,5} = \Delta\Lambda_5(\tilde{\sigma}_5, \tilde{p}'_{0,5}, \tilde{e}_5, \tilde{\omega}_5, \Delta\varepsilon_n)B(\tilde{\sigma}_5, \tilde{p}'_{0,5}, \tilde{e}_5, \tilde{\omega}_5) \quad 4-206$$

$$\tilde{\sigma}_5 = \sigma_{n-1} + \frac{226}{729}\Delta\sigma_1 - \frac{25}{27}\Delta\sigma_2 + \frac{880}{729}\Delta\sigma_3 + \frac{55}{729}\Delta\sigma_4 \quad 4-207$$

$$\tilde{p}'_{0,5} = p'_{0,n-1} + \frac{226}{729}\Delta p'_{0,1} - \frac{25}{27}\Delta p'_{0,2} + \frac{880}{729}\Delta p'_{0,3} + \frac{55}{729}\Delta p'_{0,4} \quad 4-208$$

$$\tilde{e}_5 = (1 + e_{n-1}) \cdot \exp\left(-\left(\frac{229}{729} - \frac{25}{27} + \frac{880}{729} + \frac{55}{729}\right)\Delta\varepsilon_{v,n}\right) - 1 \quad 4-209$$

$$\tilde{\omega}_5 = \omega(\tilde{e}_5, \tilde{\sigma}_5) \quad 4-210$$

Прираштај напона $\Delta\sigma_6$ и параметра ојачања $\Delta p'_{0,6}$ се рачунају преко израза:

$$\Delta\sigma_6 = \mathbf{D}^{ep}(\tilde{\sigma}_6, \tilde{p}'_{0,6}, \tilde{e}_6, \tilde{\omega}_6)\Delta\varepsilon_n \quad 4-211$$

$$\Delta p'_{0,6} = \Delta\Lambda_6(\tilde{\sigma}_6, \tilde{p}'_{0,6}, \tilde{e}_6, \tilde{\omega}_6, \Delta\varepsilon_n)B(\tilde{\sigma}_6, \tilde{p}'_{0,6}, \tilde{e}_6, \tilde{\omega}_6) \quad 4-212$$

$$\tilde{\sigma}_6 = \sigma_{n-1} - \frac{181}{270}\Delta\sigma_1 + \frac{5}{2}\Delta\sigma_2 - \frac{266}{297}\Delta\sigma_3 - \frac{91}{27}\Delta\sigma_4 + \frac{189}{55}\Delta\sigma_5 \quad 4-213$$

$$\tilde{p}'_{0,6} = p'_{0,n-1} - \frac{181}{270}\Delta p'_{0,1} + \frac{5}{2}\Delta p'_{0,2} - \frac{266}{297}\Delta p'_{0,3} - \frac{91}{27}\Delta p'_{0,4} + \frac{189}{55}\Delta p'_{0,5} \quad 4-214$$

$$\tilde{e}_6 = (1 + e_{n-1}) \cdot \exp\left(-\left(-\frac{181}{270} + \frac{5}{2} - \frac{266}{297} - \frac{91}{27} + \frac{189}{55}\right)\Delta\varepsilon_{v,n}\right) - 1 \quad 4-215$$

$$\tilde{\omega}_6 = \omega(\tilde{e}_6, \tilde{\sigma}_6) \quad 4-216$$

Локална грешка у одређивању напона и параметра ојачања је:

$$\begin{Bmatrix} \{\hat{\sigma}_n\} \\ \{\hat{p}'_{0,n}\} \end{Bmatrix} - \begin{Bmatrix} \{\sigma_n\} \\ \{p'_{0,n}\} \end{Bmatrix} = \begin{Bmatrix} \frac{11}{360}\Delta\sigma_1 - \frac{10}{63}\Delta\sigma_3 + \frac{55}{72}\Delta\sigma_4 - \frac{27}{40}\Delta\sigma_5 + \frac{11}{280}\Delta\sigma_6 \\ \frac{11}{360}\Delta p'_{0,1} - \frac{10}{63}\Delta p'_{0,3} + \frac{55}{72}\Delta p'_{0,4} - \frac{27}{40}\Delta p'_{0,5} + \frac{11}{280}\Delta p'_{0,6} \end{Bmatrix} \quad 4-217$$

Релативна грешка се одређује преко израза:

$$Er_n = \max \left\{ \frac{\|(\hat{\sigma}_n - \sigma_n)\|}{\|\hat{\sigma}_n\|}, \frac{|p'_{0,n} - \hat{p}'_{0,n}|}{\hat{p}'_{0,n}} \right\} \quad 4-218$$

Уколико је релативна грешка већа од задате толеранције *STOL*, субинкремент се одбацује. Нова величина субинкремента са којим се понавља прорачун се одређује према изразу (4-151), док се релативни однос нове и старе величине ρ субинкремента одређује преко израза:

$$\rho = \max \left\{ 0.9 \left(\frac{STOL}{Er_n} \right)^{\frac{1}{5}}, 0.1 \right\} \quad 4-219$$

Уколико се субинкремент прихвата, величина наредног субинкремента се одређује према изразу (4-153). Релативни однос наредног и претходног субинкремента ρ одређује се из:

$$\rho = \min \left\{ 0.9 \left(\frac{STOL}{Er_n} \right)^{\frac{1}{5}}, 1.1 \right\} \quad 4-220$$

Уколико у претходном покушају субинкремент није успео, онда се додатно ограничава раст наредног субинкремента:

$$\rho = \min \left\{ 0.9 \left(\frac{STOL}{Er_n} \right)^{\frac{1}{5}}, 1.0 \right\} \quad 4-221$$

RKDP поступак је тачнији од модификованог Euler-овог поступка, али захтева више времена за прорачун, јер је потребно срачунати шест пута напоне и параметре ојачања, док се исте величине код модификованог Euler-овог поступка само два пута рачунају.

4.2.4.1 Алгоритам нумеричке интеграције

Алгоритам 7. Нумеричка интеграција HASP модела Runge-Kutta-Dormand-Prince поступком са аутоматском субинкрементацијом

Корак 1. Познате величине на почетку инкремента: ${}^t\sigma$, ${}^t\epsilon$, $\Delta\epsilon$, ${}^{t+\Delta t}\epsilon$, ${}^tD'^e$ и све друге величине дате изразима (4-30)-(4-57).

Корак 2. Иницијализација субинкрементације: $T = T_0 = 0$ и $\Delta T_1 = 1$.

Корак 3. Срачунавају се: $\Delta \varepsilon_n$ из (4-120), прираштаји $\Delta \sigma_i$ и $\Delta p'_{0,i}$ ($i = 1, 2, 3, 4, 5, 6$) према изразима (4-169) – (4-216). Уколико се при прорачуну $\Delta \sigma_i, \Delta p'_{0,i}$ добије $\Delta \Lambda_i < 0$ усваја се $\Delta \Lambda_i = 0$ (деформације су еластичне).

Корак 4. Срачунавају се напони $\hat{\sigma}_n$ из (4-167) и параметар ојачања $\hat{p}'_{0,n}$ из (4-168) на крају субинкремента.

Корак 5. Срачунава се релативна грешка Er_n из (4-217) и (4-218).

Корак 6. Уколико је $Er_n > STOL$ субинкремент се одбацује и потребно је смањити псеудо време ΔT_n . За $STOL$ се може усвојити вредност мања од 10^{-4} како би се повећала тачност поступка интеграције. Прелази се на Корак 7.

Уколико је $Er_n \leq STOL$ субинкремент се прихвата и иде се на Корак 8.

Корак 7. Потребно је смањити величину субинкремента ΔT_n према (4-151) и (4-219). Понавља се прорачун са новом величином субинкремента. Враћа се на Корак 3.

Корак 8. Пошто је субинкремент био успешан, прихватају се срачунати напони и параметар ојачања. На крају субинкремента напони и параметар ојачања су:

$$\sigma_{T_n} = \hat{\sigma}_n \quad 4-222$$

$$p'_{0,T_n} = \hat{p}'_{0,n} \quad 4-223$$

Корак 9. Проверава се да ли су напони и параметар ојачања на крају субинкремента на површи течења према:

$$|F(\sigma_{T_n}, p'_{0,T_n})| \leq FTOL \quad 4-224$$

Усвојено је $FTOL = 10^{-9}$. Уколико је испуњен претходни услов, напони и параметар ојачања су на површи течења и не мора да се врши корекција. Прелази се на Корак 10.

У случају да је било који $\Delta \Lambda_i = 0$ ($i = 1, 2, 3, 4, 5, 6$) понашање тла је еластично и не треба да се врши корекција.

Уколико није испуњен услов (4-158) потребно је да се изврши корекција напона и параметра ојачања. Корекција се врши преко **Алгоритма 6**.

Корак 10. Одређује се величина наредног субинкремента ΔT_{n+1} преко израза (4-153), (4-220) и (4-221).

Корак 11. Одређује се укупно псеудо време на крају субинкремента:

$$T_n = T_{n-1} + \Delta T_n \quad 4-225$$

Корак 12. Проверава се да ли је величина наредног субинкремента ΔT_{n+1} мања од минимално прописане вредности ΔT_{min} . Уколико јесте мања онда је:

$$\Delta T_{n+1} = \Delta T_{min} \quad 4-226$$

Усвојено је $\Delta T_{min} = 10^{-6}$.

Корак 13. Проверава се да укупно псеудо време на крају наредног субинкремента не буде веће од 1:

$$T_{n+1} = T_n + \Delta T_{n+1} \leq 1 \quad 4-227$$

Уколико није задовољен претходни услов, онда је величина наредног субинкремента:

$$\Delta T_{n+1} = 1 - T_n \quad 4-228$$

Корак 14. Уколико је $T_n < 1$ започиње се наредни субинкремент и понављају се Кораци 3-13. У супротном, нанет је цео инкремент и срачунати су напони ${}^{t+\Delta t}\sigma$ и параметар ојачања ${}^{t+\Delta t}p'_0$ на крају инкремента. Завршава се прорачун.

Крај Алгоритма 7.

4.3 Интеграција конститутивних релација HASP-E модела

У овом потпоглављу приказани су поступци и алгоритми нумеричке интеграције релација HASP-E модела. Од имплицитних метода коришћена је само метода водећег параметра, а од експлицитних коришћене су обе методе. Општи поступак нумеричке интеграције према претходне три методе је приказан у потпоглављу 4.2. Као што је приказано на крају потпоглавља 3.3.4, приликом имплементација HASP-E у PLAXIS, остављена је могућност да се не морају унети сви параметри HASP-E. Уколико се унесу само параметри који су потребни за HASP модел, HASP-E ће се у потпуности понашати као HASP модел. Уколико се унесу сви параметри сем G_0^{ref} и $\gamma_{0,7}$, "Brick" компонента HASP-E модела неће бити активна и еластични модули се рачунају као у HASP моделу. Уколико се унесу сви параметри осим M_e , HASP-E ће користити исту површ течења, граничну површ и површ пластичног потенцијала као HASP модел (крugови у девијаторској равни и асоцијативна пластичност), док ће се еластични модули рачунати преко "Brick" компоненте.

Разлика у алгоритмима за нумеричку интеграцију између HASP и HASP-E модела је у томе на који начин се узима у обзир облик површи течења у девијаторској равни и како се рачунају еластични модули. Свака од метода за нумеричку интеграцију на различит начин узима у обзир облик површи течења, док се еластични модули рачунају на исти начин. У наставку ће бити приказан алгоритам за одређивање тангентних модула смицања и запреминског модула код HASP-E који је исти за све методе нумеричке интеграције.

Приликом иницијализације прорачуна, на основу унетих вредности за G_0^{ref} и $\gamma_{0,7}$, рачуна се проценат деградације модула смицања (процент разлике између G_0^{ref} и G_{ur}^{ref}) за сваку циглу $\Delta\omega_G^b$ из израза (3-79) и дужина канапа s^b из израза (3-83). Ове вредности су константне током целог прорачуна и током свих фаза. У алгоритму за одређивање тангентних модула у HASP-E моделу на једном (суб)инкременту рачуна се референтни тангентни модул смицања на почетку (суб)инкремента G_n^{ref} и референтни тангентни

модул смицања на крају (суб)инкремента G_{n+1}^{ref} . Уколико је на посматраном (суб)инкременту дошло до промене путање напона или деформација која узрокује да тангентни модул смицања у потпуности или делимично поврати максималну вредност G_0^{ref} , онда ће G_{n+1}^{ref} бити веће од G_n^{ref} . За прорачун напона на посматраном (суб)инкременту користиће се већа вредност од G_n^{ref} и G_{n+1}^{ref} . Референтни тангентни модул смицања на крају i -тог (суб)инкремента је једнак референтном тангентном модулу на почетку наредног $i + 1$ (суб)инкремента. У методи водећег параметра, тангентни модули се рачунају на почетку сваког инкремента, док се у експлицитним метода рачунају на почетку сваког субинкремента. У HASP-E моделу користи се $N_b = 20$ цигли, а приказани алгоритам је општи, без обзира колико се цигли користи. У **Алгоритму 8**, индекс n у променљивим се односи на почетак (суб)инкремента, а $n + 1$ на крај (суб)инкремента. Број затегнутих канапа је обележен са N_t .

Алгоритам 8. Одређивање тангентних модула у “Brick” компоненти HASP-E модела

Корак 1. Иницијализација прорачуна. Познате величине су: укупне деформације на почетку (суб)инкремента ϵ_n , прираштај укупних деформација на (суб)инкременту $\Delta\epsilon_{n+1}$, укупне деформације на крају (суб)инкремента ϵ_{n+1} , координате (деформације) цигли на почетку (суб)инкремента ϵ_n^b , дужине канапа s^b , проценат деградације модула смицања (процент разлике између G_0^{ref} и G_{ur}^{ref}) за сваку циглу $\Delta\omega_G^b$, референтни модул смицања за веома мале деформације G_0^{ref} , референтни модул смицања за велике деформације G_{ur}^{ref} , референтни тангентни модул смицања на почетку (суб)инкремента G_n^{ref} , коефицијент порозности на почетку (суб)инкремента e_n , средњи ефективни напон на почетку (суб)инкремента p'_n и Poisson-ов коефицијент μ' . Број затегнутих канапа је $N_t = 0$.

Корак 2. За сваку циглу $b = 1, \dots, N_b$ срачунати:

-вектор разлике ξ^b између укупних деформација на крају (суб)инкремента ϵ_{n+1} и деформација b -те цигле на почетку суб(инкремента) ϵ_n^b према изразу (3-84)

-растојање γ_{n+1}^b преко израза (3-85).

-Уколико је $\gamma_{n+1}^b \leq s^b$, канап b -те цигле је опуштен на крају (суб)инкремента и цигла се није померала током (суб)инкремента. Координате b -те цигле на крају (суб)инкремента ϵ_{n+1}^b су једнаке координатама на почетку (суб)инкремента ϵ_n^b (израз (3-86)).

-Уколико је $\gamma_{n+1}^b > s^b$, канап b -те цигле је затегнут на крају (суб)инкремента и цигла се померила током (суб)инкремента. Координате цигле на крају (суб)инкремента ϵ_{n+1}^b се рачунају преко израза (3-87) и бројач затегнутих канапа се повећава за један:

$$N_t = N_t + 1 \quad 4-229$$

Корак 3. Срачунава се референтни тангентни модул смицања на крају (субинкремента) G_{n+1}^{ref} према изразима (3-80) и (3-81).

Корак 4. Уколико је $G_n^{ref} > G_{n+1}^{ref}$ онда је вредност референтног тангентног модула за прорачун напона на посматраном (суб)инкременту:

$$G^{ref} = G_n^{ref} \quad 4-230$$

У супротном, дошло је до промене путање напона или деформација и:

$$G^{ref} = G_{n+1}^{ref} \quad 4-231$$

Корак 5. Рачуна се тангентни модул смицања G према изразу (3-93), тангентни запремински модул K према изразу (3-95) и вредност параметра κ која је потребна за закон ојачања из израза (3-96) за вредност средњег ефективног напона p'_n и коефицијент порозности e_n .

Корак 6. Вредности G , K и κ се враћају главном алгоритму за прорачун напона. Положаји цигли ϵ_{n+1}^b и G_{n+1}^{ref} на крају (суб)инкремента се чувају за почетак следећег (суб)инкремента. Крај прорачуна.

Крај Алгоритма 8.

4.3.1 Метода водећег параметра

За интеграцију конститутивних релација HASP-E модела методом водећег параметра потребно је унети измене у неколико израза приказаних у потпоглављу 4.2.1.3 како би се узела промена облика површи течења у девијаторској равни. Тангентни еластични модули и параметар κ потребан за закон ојачања се рачунају према **Алгоритму 8**, или према изразима (4-55) и (4-56) уколико није активна "Brick" компонента HASP-E модела (уколико је унето као улазни податак $G_0^{ref} = 0$ и $\gamma_{0,7} = 0$). Пошто у HASP-E моделу функција течења зависи од вредности Lode-овог угла, функција течења у конфигурацији $t + \Delta t$ има следећи облик:

$${}^{t+\Delta t}F({}^{t+\Delta t}p', {}^{t+\Delta t}q, {}^{t+\Delta t}\theta, {}^{t+\Delta t}p'_0) = \frac{{}^{t+\Delta t}q^2}{M({}^{t+\Delta t}\theta)^2} + {}^{t+\Delta t}p'({}^{t+\Delta t}p' - {}^{t+\Delta t}p'_0) = 0 \quad 4-232$$

У претходном изразу све величине се рачунају у конфигурацији $t + \Delta t$. Lode-ов угао ${}^{t+\Delta t}\theta$ се рачуна према изразу:

$${}^{t+\Delta t}\theta = -\frac{1}{3} \operatorname{asin} \left(\frac{27}{2} \frac{{}^{t+\Delta t}J_{3d}}{{}^{t+\Delta t}q^3} \right) \quad 4-233$$

Трећа инваријанта девијаторских напона у конфигурацији $t + \Delta t$ ${}^{t+\Delta t}J_{3d}$ се рачуна према изразу:

$$\begin{aligned} {}^{t+\Delta t}J_{3d} = & ({}^{t+\Delta t}\sigma'_x - {}^{t+\Delta t}p')({}^{t+\Delta t}\sigma'_y - {}^{t+\Delta t}p')({}^{t+\Delta t}\sigma'_z - {}^{t+\Delta t}p') \\ & + 2 {}^{t+\Delta t}\tau_{xy} {}^{t+\Delta t}\tau_{yz} {}^{t+\Delta t}\tau_{zx} - ({}^{t+\Delta t}\sigma'_x - {}^{t+\Delta t}p') {}^{t+\Delta t}\tau_{yz}^2 - ({}^{t+\Delta t}\sigma'_y \\ & - {}^{t+\Delta t}p') {}^{t+\Delta t}\tau_{zx}^2 - ({}^{t+\Delta t}\sigma'_z - {}^{t+\Delta t}p') {}^{t+\Delta t}\tau_{xy}^2 \end{aligned} \quad 4-234$$

Вредност параметра $M(t+\Delta t \theta)$ се рачуна према изразу:

$$M(t+\Delta t \theta) = X(1 + Y \cdot \sin 3 t+\Delta t \theta)^Z \quad 4-235$$

У функцији пластичног потенцијала параметар M се мења са $M(t\theta_c)$, па су онда изрази за прираштаје пластичне дисторзијске и пластичних девијаторских деформација:

$$\Delta \varepsilon_q^p = \Delta \Lambda \frac{\partial P}{\partial q} = \Delta \Lambda \frac{2 t+\Delta t q}{M(t\theta_c)^2} \quad 4-236$$

$$\Delta \varepsilon^p = 3\Delta \Lambda \frac{t+\Delta t s}{M(t\theta_c)^2} \quad 4-237$$

У претходна два израза, $t\theta_c$ је вредности Lode-овог угла на почетку инкремента:

$$t\theta_c = -\frac{1}{3} \operatorname{asin} \left(\frac{27 tJ_{3d}}{2 t q^3} \right) \quad 4-238$$

У претходном изразу tq је дато изразом (4-32) а tJ_{3d} се рачуна преко:

$$tJ_{3d} = (t\sigma'_x - t p')(t\sigma'_y - t p')(t\sigma'_z - t p') + 2 t\tau_{xy} t\tau_{yz} t\tau_{zx} - (t\sigma'_x - t p') t\tau_{yz}^2 - (t\sigma'_y - t p') t\tau_{zx}^2 - (t\sigma'_z - t p') t\tau_{xy}^2 \quad 4-239$$

Параметар $M(t\theta_c)$ се рачуна према изразу:

$$M(t\theta_c) = X(1 + Y \cdot \sin 3 t\theta_c)^Z \quad 4-240$$

Иако би требало да се вредност параметра $M(\theta_c)$ у изразима (4-236) и (4-237) срачуна за стање напона на крају инкремента (имплицитна метода), због једноставности се параметар $M(\theta_c)$ рачуна за стање напона на почетку инкремента. Пошто напонско стање на крају инкремента није познато, онда ни параметар $M(\theta_c)$ на крају инкремента није познат унапред и у том случају би било потребно увести додатну итеративну процедуру како би се тај параметар срачунао. Уколико је инкремент довољно мали, вредност параметра $M(\theta_c)$ за напонско стање на почетку инкремента и на крају инкремента не би требало битно да се разликује, па ће се из тог разлога овај параметар рачунати за напонско стање на почетку инкремента.

Израз за параметар ојачања на крају инкремента је:

$$t+\Delta t p'_0 = t p'_0 \cdot \exp \left(\frac{(t e - t+\Delta t e) t \omega}{\lambda - \check{\kappa}} \right) \cdot \left(\frac{t p'}{t+\Delta t p'} \right)^{\frac{\check{\kappa} t \omega}{\lambda - \check{\kappa}}} \quad 4-241$$

Израз за прираштај пластичне запреминске деформације је:

$$\Delta \varepsilon_v^p = \frac{\lambda - \check{\kappa}}{1 + t e} \cdot \frac{1}{t \omega} \cdot \ln \left(\frac{t+\Delta t p'_0}{t p'_0} \right) \quad 4-242$$

Изрази (4-241) и (4-242) су изведени интеграцијом закона ојачања (3-92) и параметар $\check{\kappa}$ се мења на посматраном инкременту на коме се врши интеграција. Промена параметра $\check{\kappa}$ на инкременту је степенаста у складу са степенастом S-кривом у "Brick" моделу. За велике деформације, параметар $\check{\kappa}$ има константну вредност и једнак је κ . Приликом интеграције закона ојачања (3-92) усвојено је да је вредност параметра $\check{\kappa}$ константна на посматраном инкременту. Уколико је величина инкремента довољно мала и ако се узме у обзир степенаста природа промене $\check{\kappa}$ (величина инкремента може да буде таква да је целокупна деформација на инкременту на једној степеници), вредност параметра $\check{\kappa}$ не би требало да се драстично мења током инкремента, тако да се не прави велика грешка ако се усвоји константна вредност параметра $\check{\kappa}$ на посматраном инкременту.

Пластични мултипликатор се рачуна из (4-19). Вектор девијаторских напона се рачуна из израза:

$${}^{t+\Delta t}\mathbf{s} = \frac{\Delta \boldsymbol{\varepsilon}' + \frac{{}^t\mathbf{s}}{2G}}{\frac{3\Delta\lambda}{M({}^t\theta_c)^2} + \frac{1}{2G}} \quad 4-243$$

У случају да је услов дат изразом (4-27) задовољен (достигнут напонски однос $\eta = M({}^t\theta_c)$), пластични мултипликатор се одређује из израза:

$$\Delta\lambda = \frac{M({}^t\theta_c)}{2 \cdot {}^{t+\Delta t}p' \cdot {}^t\omega} \left(\Delta \varepsilon_q - \frac{{}^{t+\Delta t}q - {}^tq}{3G} \right) \quad 4-244$$

Вредност коефицијента ојачања ${}^t\omega$ срачунава према изразима (4-50) – (4-53). У изразу (4-51) за рачунање ${}^t\bar{\Psi}$ уместо M користи $M({}^t\theta_c)$:

$${}^t\bar{\Psi} = (\lambda - \kappa) \ln \left(\frac{2M({}^t\theta_c)^2}{M({}^t\theta_c)^2 + {}^t\eta^2} \right) \quad 4-245$$

У изразима (4-245) и (4-52) се користи вредност κ која је улазни параметар модела, а не $\check{\kappa}$.

Вредност параметрај ојачања на почетку инкремента је:

$${}^t p'_0 = \frac{{}^t q^2}{M({}^t\theta_c)^2 {}^t p'} + {}^t p' \quad 4-246$$

Уколико је $M_c = M_e$, онда за све вредности Lode-овог угла, параметар M има исту вредност $M(\theta) = M = M_c = M_e$ и $X = M_c = M_e, Y = 0$.

4.3.1.1 Алгоритам нумеричке интеграције

Алгоритам 9. Нумеричка интеграција HASP-E модела методом водећег параметра

Корак 1. Познате величине на почетку инкремента: ${}^t\boldsymbol{\sigma}, {}^t\boldsymbol{\varepsilon}, \Delta \boldsymbol{\varepsilon}, {}^{t+\Delta t}\boldsymbol{\varepsilon}$. Уколико су задате вредности G_0^{ref} и $\gamma_{0,7}$, тангентни модули G и K и параметар $\check{\kappa}$ рачунају се према

Алгоритму 8. У супротном, G и K се рачунају према (4-55) и (4-56). На основу вредности G и K попуњава се матрица D'^e . Срачунава се $M({}^t\theta_c)$ из (4-238), (4-239) и (4-240), ${}^t\omega$ из (4-50), (4-52), (4-53) и (4-245), ${}^t p'_0$ из (4-246), ${}^t e$ из (4-46), ${}^{t+\Delta t}e$ из (4-48). Остале напонске и деформацијске инваријанте које су потребне за прорачун одређују се из израза (4-30) – (4-45).

Корак 2. Претпоставља се иницијална вредност водећег параметра ${}^{t+\Delta t}p'_i$ са којим се креће у тражење нуле функције. Као иницијална вредност претпоставља се вредност која одговара еластичном предвиђању:

$${}^{t+\Delta t}p'_i = {}^t p' \cdot \exp\left(\frac{{}^t e - {}^{t+\Delta t}e}{\check{\kappa}}\right) \quad 4-247$$

Корак 3. Користећи **Алгоритам 10**, рачуна се ${}^{t+\Delta t}f_i({}^{t+\Delta t}p'_i)$, ${}^{t+\Delta t}f_i({}^{t+\Delta t}p'_i + \Delta p')$, ${}^{t+\Delta t}f_i({}^{t+\Delta t}p'_i - \Delta p')$. Три вредности функције течења су потребне да би могао да се срачуна први и други извод функције течења. Усвојено је $\Delta p' = 10^{-6} kPa$. Даље се рачуна ${}^{t+\Delta t}f'_i({}^{t+\Delta t}p'_i)$ преко израза (4-14) и ${}^{t+\Delta t}f''_i({}^{t+\Delta t}p'_i)$ преко израза (4-15).

Корак 4. Користи се Halley-ев поступак за тражење нуле. Следећа потенцијална нула ${}^{t+\Delta t}p'_{i+1}$ се добија преко израза (4-13). За нову вредност ${}^{t+\Delta t}p'_{i+1}$ проверава се да ли је задовољен услов (4-59) у коме је усвојено $FTOL = 10^{-9}$. Уколико није задовољен услов (4-59), понављају се Корак 3 и Корак 4 за вредност ${}^{t+\Delta t}p'_{i+1}$ све док услов не буде задовољен, а максимално k_{max} пута. Усвојено је $k_{max} = 25$. С обзиром на тачност и брзину конвергенције Halley-евог поступка, након десетак итерација услов (4-59) бива задовољен.

Корак 5. Проверава се да ли је за нађену нулу ${}^{t+\Delta t}p'$ задовољен услов $\Delta l \geq 0$. Уколико није задовољен тај услов, постоји могућност да је нађена „лажна нула“ и претпоставља се нова иницијална вредност ${}^{t+\Delta t}p'_i$ са којом се понављају Корак 3 и Корак 4. За нову вредност ${}^{t+\Delta t}p'_i$ може се усвојити вредност на почетку инкремента ${}^t p'$ или вредност за коју се добија ${}^{t+\Delta t}p'_0 = {}^t p'_0$ (критично стање):

$${}^{t+\Delta t}p'_i = {}^t p'_i \cdot \left(\frac{1}{\exp\left(\frac{({}^t e - {}^{t+\Delta t}e) {}^t \omega}{\lambda - \check{\kappa}}\right)} \right)^{\frac{\lambda - \check{\kappa}}{\check{\kappa} {}^t \omega}} \quad 4-248$$

Уколико се и даље добија да је $\Delta l < 0$, онда је понашање тла еластично и напони на крају инкремента се рачунају преко (4-61).

Корак 6. За срачунату вредност водећег параметра ${}^{t+\Delta t}p'$ преко **Алгоритма 10** рачунају се напони на крају инкремента ${}^{t+\Delta t}\sigma$ и све друге непознате величине у конфигурацији $t + \Delta t$.

Крај Алгоритма 9.

Алгоритам 10. Прорачун напона и функције течења у конфигурацији $t + \Delta t$ за познату вредност водећег параметра ${}^{t+\Delta t}p'$ у HASP-E моделу.

Корак 1. Срачуна се параметар ојачања ${}^{t+\Delta t}p'_0$ из израза (4-241).

Корак 2. Уколико је задовољен услов (4-62) у коме је усвојено $CSTOL = 10^{-6}$, срачунава се пластични мултипликатор Δl из израза (4-244). Уколико није задовољен услов (4-62), срачунава се прво прираштај пластичних запреминских деформација $\Delta \varepsilon_v^p$ из израза (4-242), па затим пластични мултипликатор Δl из израза (4-19).

Корак 3. Срачунава се девијаторски део тензора напона ${}^{t+\Delta t}\mathbf{s}$ из израза (4-243).

Корак 4. Срачунава се вредност функције течења ${}^{t+\Delta t}f({}^{t+\Delta t}p')$ преко израза (4-232). Вредност функције течења је потребна у **Алгоритму 9** за тражење нуле функције.

Корак 5. Срачунавају се напони ${}^{t+\Delta t}\boldsymbol{\sigma}$ преко израза (4-29), као и све друге непознате величине.

Крај Алгоритма 10.

С обзиром да су при интеграцији конститутивних релација HASP-E модела методом водећег параметра усвојене две претпоставке: да се параметар $M({}^t\theta_c)$ у једначини пластичног потенцијала рачуна на основу напонског стања на почетку инкремента, а не на крају инкремента и да параметар $\check{\kappa}$ при интеграцији закона ојачања има константну вредност на инкременту, у оквиру будућих истраживања би требало предвидети поступак субинкрементације како би се минимизирала потенцијална грешка при интеграцији напона услед усвајања ове две претпоставке.

4.3.2 Модификовани Euler-ов поступак са аутоматском субинкрементацијом и контролом грешке

Као и код методе водећег параметра, еластични модули K и G и параметар $\check{\kappa}$ се одређују преко **Алгоритма 8** (уколико су задате вредности вредности G_0^{ref} и $\gamma_{0,7}$). У супротном рачунају се на исти начин као код HASP модела. Пошто у HASP-E моделу површ течења и површ пластичног потенцијала нису исте, вектори \mathbf{a} и \mathbf{b} се међусобно разликују. Функција течења на почетку инкремента дата је изразом:

$$F({}^t\boldsymbol{\sigma}, {}^tp'_0) = \frac{{}^tq^2}{M({}^t\theta)^2} + {}^tp'({}^tp' - {}^tp'_0) = 0 \quad 4-249$$

А функција пластичног потенцијала је:

$$P({}^t\boldsymbol{\sigma}, {}^tp'_0) = \frac{{}^tq^2}{M({}^t\theta_c)^2} + {}^tp'({}^tp' - {}^tp'_0) = 0 \quad 4-250$$

Парцијални изводи функције течења по напонима су:

$$\frac{\partial F({}^t\boldsymbol{\sigma}, {}^tp'_0)}{\partial \boldsymbol{\sigma}} = \frac{\partial F({}^t\boldsymbol{\sigma}, {}^tp'_0)}{\partial {}^tp'} \frac{\partial {}^tp'}{\partial {}^t\boldsymbol{\sigma}} + \frac{\partial F({}^t\boldsymbol{\sigma}, {}^tp'_0)}{\partial {}^tq} \frac{\partial {}^tq}{\partial {}^t\boldsymbol{\sigma}} + \frac{\partial F({}^t\boldsymbol{\sigma}, {}^tp'_0)}{\partial {}^t\theta} \frac{\partial {}^t\theta}{\partial {}^t\boldsymbol{\sigma}} \quad 4-251$$

Парцијални изводи у (4-251) су редом:

$$\frac{\partial F({}^t\sigma, {}^tp'_0)}{\partial {}^tp'} = 2 {}^tp' - {}^tp'_0 \quad 4-252$$

$$\frac{\partial F({}^t\sigma, {}^tp'_0)}{\partial {}^tq} = \frac{2 {}^tq}{M({}^t\theta)^2} \quad 4-253$$

$$\begin{aligned} \frac{\partial F({}^t\sigma, {}^tp'_0)}{\partial {}^t\theta} &= \frac{-2 {}^tq^2}{M({}^t\theta)^3} \cdot \frac{\partial M({}^t\theta)}{\partial {}^t\theta} \\ &= \frac{-2 {}^tq^2}{[X(1 + Y \cdot \sin 3 {}^t\theta)^Z]^3} \cdot 3XYZ \cdot (1 + Y \cdot \sin 3 {}^t\theta)^{Z-1} \cdot \cos 3 {}^t\theta \\ &= \frac{-6 {}^tq^2YZ \cdot \cos 3 {}^t\theta}{X^2} \cdot (1 + Y \cdot \sin 3 {}^t\theta)^{-2Z-1} \end{aligned} \quad 4-254$$

$$\frac{\partial {}^tp'}{\partial {}^t\sigma} = \left(\frac{1}{3} \quad \frac{1}{3} \quad \frac{1}{3} \quad 0 \quad 0 \quad 0 \right)^T = \mathbf{a}_p \quad 4-255$$

$$\frac{\partial {}^tq}{\partial {}^t\sigma} = \frac{3}{2 {}^tq} \left({}^t\sigma'_x - {}^tp' \quad {}^t\sigma'_y - {}^tp' \quad {}^t\sigma'_z - {}^tp' \quad 2 {}^t\tau_{xy} \quad 2 {}^t\tau_{yz} \quad 2 {}^t\tau_{zx} \right)^T = \mathbf{a}_q \quad 4-256$$

$$\begin{aligned} \frac{\partial {}^t\theta}{\partial \sigma} &= -\frac{1}{3} \cdot \frac{1}{\sqrt{1 - \frac{27^2 {}^tJ_{3d}^2}{2^2 {}^tq^6}}} \cdot \frac{27}{2} \cdot \frac{\partial \left(\frac{{}^tJ_{3d}}{{}^tq^3} \right)}{\partial \sigma} = \\ &= -\frac{1}{3} \cdot \frac{1}{\sqrt{1 - \frac{27^2 {}^tJ_{3d}^2}{2^2 {}^tq^6}}} \cdot \frac{27}{2} \cdot \frac{\frac{\partial {}^tJ_{3d}}{\partial \sigma} {}^tq^3 - 3 {}^tq^2 {}^tJ_{3d} \frac{\partial {}^tq}{\partial \sigma}}{{}^tq^6} \\ &= -\frac{1}{3} \cdot \frac{1}{\sqrt{1 - \frac{27^2 {}^tJ_{3d}^2}{2^2 {}^tq^6}}} \cdot \frac{27}{2} \cdot \frac{\mathbf{a}_{J3} {}^tq - 3 {}^tJ_{3d} \mathbf{a}_q}{{}^tq^4} \end{aligned} \quad 4-257$$

Коначни израз за вектор \mathbf{a} је:

$$\begin{aligned} \mathbf{a} &= (2 {}^tp' - {}^tp'_0) \cdot \mathbf{a}_p + \frac{2 {}^tq}{M({}^t\theta)^2} \cdot \mathbf{a}_q + \frac{6 {}^tq^2YZ \cdot \cos 3 {}^t\theta}{X^2} \cdot (1 + Y \cdot \sin 3 {}^t\theta)^{-2Z-1} \\ &\quad \cdot \frac{1}{3} \cdot \frac{1}{\sqrt{1 - \frac{27^2 {}^tJ_{3d}^2}{2^2 {}^tq^6}}} \cdot \frac{27}{2} \cdot \frac{\mathbf{a}_{J3} {}^tq - 3 {}^tJ_{3d} \mathbf{a}_q}{{}^tq^4} \end{aligned} \quad 4-258$$

Парцијални изводи треће инваријанте девијаторских напона је:

$$\begin{aligned}
 \mathbf{a}_{J_3} &= \frac{\partial {}^t J_{3d}}{\partial \boldsymbol{\sigma}} \\
 &= \begin{bmatrix} \frac{4 {}^t \sigma'_y {}^t \sigma'_z}{9} - \frac{2 {}^t \sigma'_x {}^t \sigma'_y}{9} - \frac{2 {}^t \sigma'_x {}^t \sigma'_z}{9} + \frac{2 {}^t \sigma_x'^2}{9} + \frac{{}^t \tau_{xy}^2}{3} + \frac{{}^t \tau_{zx}^2}{3} - \frac{2 {}^t \tau_{yz}^2}{3} - \frac{{}^t \sigma_y'^2}{9} - \frac{{}^t \sigma_z'^2}{9} \\ \frac{4 {}^t \sigma'_x {}^t \sigma'_z}{9} - \frac{2 {}^t \sigma'_x {}^t \sigma'_y}{9} - \frac{2 {}^t \sigma'_y {}^t \sigma'_z}{9} + \frac{2 {}^t \sigma_y'^2}{9} + \frac{{}^t \tau_{xy}^2}{3} + \frac{{}^t \tau_{yz}^2}{3} - \frac{2 {}^t \tau_{zx}^2}{3} - \frac{{}^t \sigma_x'^2}{9} - \frac{{}^t \sigma_z'^2}{9} \\ \frac{4 {}^t \sigma'_x {}^t \sigma'_y}{9} - \frac{2 {}^t \sigma'_x {}^t \sigma'_z}{9} - \frac{2 {}^t \sigma'_y {}^t \sigma'_z}{9} + \frac{2 {}^t \sigma_z'^2}{9} + \frac{{}^t \tau_{yz}^2}{3} + \frac{{}^t \tau_{zx}^2}{3} - \frac{2 {}^t \tau_{xy}^2}{3} - \frac{{}^t \sigma_x'^2}{9} - \frac{{}^t \sigma_y'^2}{9} \\ \frac{2 {}^t \sigma'_x {}^t \tau_{xy}}{3} + \frac{2 {}^t \sigma'_y {}^t \tau_{xy}}{3} - \frac{4 {}^t \sigma'_z {}^t \tau_{xy}}{3} + 2 {}^t \tau_{yz} {}^t \tau_{zx} \\ \frac{2 {}^t \sigma'_y {}^t \tau_{yz}}{3} + \frac{2 {}^t \sigma'_z {}^t \tau_{yz}}{3} - \frac{4 {}^t \sigma'_x {}^t \tau_{yz}}{3} + 2 {}^t \tau_{xy} {}^t \tau_{zx} \\ \frac{2 {}^t \sigma'_x {}^t \tau_{zx}}{3} + \frac{2 {}^t \sigma'_z {}^t \tau_{zx}}{3} - \frac{4 {}^t \sigma'_y {}^t \tau_{zx}}{3} + 2 {}^t \tau_{xy} {}^t \tau_{yz} \end{bmatrix} \quad 4-259
 \end{aligned}$$

Изрази за величине A и B су:

$$A = -\frac{\partial F}{\partial p'_0} \frac{dp'_0}{d\Delta\lambda} = \frac{{}^t \nu {}^t \omega}{\lambda - \check{\kappa}} {}^t p' {}^t p'_0 (2 {}^t p' - {}^t p'_0) \quad 4-260$$

$$B = -\frac{A}{\frac{\partial F}{\partial p'_0}} = \frac{{}^t \nu {}^t \omega}{\lambda - \check{\kappa}} {}^t p'_0 (2 {}^t p' - {}^t p'_0) \quad 4-261$$

Парцијални изводи функције пластичног потенцијала по напонима и вектор \mathbf{b} су:

$$\frac{\partial P({}^t \boldsymbol{\sigma}, {}^t p'_0)}{\partial \boldsymbol{\sigma}} = \frac{\partial P({}^t \boldsymbol{\sigma}, {}^t p'_0)}{\partial {}^t p'} \frac{\partial {}^t p'}{\partial {}^t \boldsymbol{\sigma}} + \frac{\partial P({}^t \boldsymbol{\sigma}, {}^t p'_0)}{\partial {}^t q} \frac{\partial {}^t q}{\partial {}^t \boldsymbol{\sigma}} \quad 4-262$$

$$\mathbf{b} = \begin{bmatrix} \frac{2 {}^t p' - {}^t p'_0}{3} + 3 \frac{{}^t \sigma'_x - {}^t p'}{M({}^t \theta_c)^2} \\ \frac{2 {}^t p' - {}^t p'_0}{3} + 3 \frac{{}^t \sigma'_y - {}^t p'}{M({}^t \theta_c)^2} \\ \frac{2 {}^t p' - {}^t p'_0}{3} + 3 \frac{{}^t \sigma'_z - {}^t p'}{M({}^t \theta_c)^2} \\ \frac{6 {}^t \tau_{xy}}{M({}^t \theta_c)^2} \\ \frac{6 {}^t \tau_{yz}}{M({}^t \theta_c)^2} \\ \frac{6 {}^t \tau_{zx}}{M({}^t \theta_c)^2} \end{bmatrix} \quad 4-263$$

Поступак корекције напона и параметара стања на крају инкремента се врши на исти начин описан у потпоглављу 4.2.2.1 и према **Алгоритму 6**. Приликом корекције напона, у функцији пластичног потенцијала се користи иста вредност параметра $M({}^t\theta_c)$ као у (4-250) која је срачуната на почетку субинкремента.

На почетку једног субинкремента напони су σ_{n-1} , параметар ојачања је $p'_{0,n-1}$, коефицијент порозности је e_{n-1} , специфична запремина је v_{n-1} , коефицијент ојачања је ω_{n-1} , а укупне деформације су ϵ_{n-1} . Прираштај деформација на n -том субинкременту $\Delta\epsilon_n$ је дат у (4-120). Коефицијент ојачања ω_{n-1} се рачуна са стањем напона на почетку субинкремента σ_{n-1} и параметром $M(\theta_{n-1})$. На крају субинкремента коефицијент порозности је e_n и специфична запремина је v_n .

Прираштаји напона $\Delta\sigma_i$ и параметра ојачања $\Delta p'_{0,i}$ ($i = 1,2$) у модификованом Euler-овом поступку другог реда се одређују према изразима:

$$\Delta\sigma_i = \Delta\sigma_i^e - \Delta\Lambda_i \mathbf{D}_i^e \mathbf{b}_i \quad 4-264$$

$$\Delta p'_{0,i} = \Delta\Lambda_i B_i \quad 4-265$$

$$\Delta\sigma_i^e = \mathbf{D}_i^e \Delta\epsilon_n \quad 4-266$$

$$\Delta\Lambda_i = \frac{\mathbf{a}_i^T \mathbf{D}_i^e \Delta\epsilon_n}{A_i + \mathbf{a}_i^T \mathbf{D}_i^e \mathbf{b}_i} \quad 4-267$$

$$A_i = \frac{\tilde{v}_i \tilde{\omega}_i}{\lambda - \tilde{\kappa}_i} \tilde{p}'_{0,i} (2\tilde{p}'_i - \tilde{p}'_{0,i}) \quad 4-268$$

$$B_i = \frac{\tilde{v}_i \tilde{\omega}_i}{\lambda - \tilde{\kappa}_i} \tilde{p}'_{0,i} (2\tilde{p}'_i - \tilde{p}'_{0,i}) \quad 4-269$$

$$\mathbf{a}_i = \frac{\partial F(\tilde{\sigma}_i, \tilde{p}'_{0,i})}{\partial \sigma} \quad 4-270$$

$$\mathbf{b}_i = \frac{\partial P(\tilde{\sigma}_i, \tilde{p}'_{0,i})}{\partial \sigma} \quad 4-271$$

$$\mathbf{D}_i^e = \mathbf{D}^e(\tilde{K}_i, \tilde{G}_i) \quad 4-272$$

Изрази (4-264) – (4-272) могу да се представе у скраћеном облику:

$$\Delta\sigma_i = \mathbf{D}^{ep}(\tilde{\sigma}_i, \tilde{p}'_{0,i}, \tilde{e}_i, \tilde{\omega}_i, \tilde{K}_i, \tilde{G}_i, \tilde{\kappa}_i) \Delta\epsilon_n \quad 4-273$$

$$\Delta p'_{0,i} = \Delta\Lambda_i(\tilde{\sigma}_i, \tilde{p}'_{0,i}, \tilde{e}_i, \tilde{\omega}_i, \Delta\epsilon_n, \tilde{K}_i, \tilde{G}_i, \tilde{\kappa}_i) B(\tilde{\sigma}_i, \tilde{p}'_{0,i}, \tilde{e}_i, \tilde{\omega}_i, \tilde{\kappa}_i) \quad 4-274$$

У изразима (4-264) и (4-272) \tilde{p}'_i је средњи ефективни напон за $\tilde{\sigma}_i$, а \tilde{v}_i је специфична запремина дата изразом (4-180).

Прираштај напона $\Delta\sigma_1$ и параметра ојачања $\Delta p'_{0,1}$ се рачунају према изразима:

$$\Delta\sigma_1 = \mathbf{D}^{ep}(\tilde{\sigma}_1, \tilde{p}'_{0,1}, \tilde{e}_1, \tilde{\omega}_1, \tilde{K}_1, \tilde{G}_1, \tilde{\kappa}_1) \Delta\epsilon_n \quad 4-275$$

$$\Delta p'_{0,1} = \Delta L_1(\tilde{\sigma}_1, \tilde{p}'_{0,1}, \tilde{e}_1, \tilde{\omega}_1, \Delta \varepsilon_n, \tilde{K}_1, \tilde{G}_1, \tilde{\kappa}_1) B(\tilde{\sigma}_1, \tilde{p}'_{0,1}, \tilde{e}_1, \tilde{\omega}_1, \tilde{\kappa}_1) \quad 4-276$$

$$\tilde{\sigma}_1 = \sigma_{n-1} \quad 4-277$$

$$\tilde{p}'_{0,1} = p'_{0,n-1} \quad 4-278$$

$$\tilde{e}_1 = e_{n-1} \quad 4-279$$

$$\tilde{\omega}_1 = \omega_{n-1} \quad 4-280$$

Уколико није активна “Brick” компонента модела, \tilde{K}_1 и \tilde{G}_1 се рачунају као код HASP-а и $\tilde{\kappa}_1 = \kappa$. У супротном, \tilde{K}_1, \tilde{G}_1 и $\tilde{\kappa}_1$ се рачунају према **Алгоритму 8** за напонско-деформацијско стање на почетку субинкремента.

Прираштај напона $\Delta \sigma_2$ и параметра ојачања $\Delta p'_{0,2}$ се рачунају према изразима:

$$\Delta \sigma_2 = D^{ep}(\tilde{\sigma}_2, \tilde{p}'_{0,2}, \tilde{e}_2, \tilde{\omega}_2, \tilde{K}_2, \tilde{G}_2, \tilde{\kappa}_2) \Delta \varepsilon_n \quad 4-281$$

$$\Delta p'_{0,2} = \Delta L_2(\tilde{\sigma}_2, \tilde{p}'_{0,2}, \tilde{e}_2, \tilde{\omega}_2, \Delta \varepsilon_n, \tilde{K}_2, \tilde{G}_2, \tilde{\kappa}_2) B(\tilde{\sigma}_2, \tilde{p}'_{0,2}, \tilde{e}_2, \tilde{\omega}_2, \tilde{\kappa}_2) \quad 4-282$$

$$\tilde{\sigma}_2 = \sigma_{n-1} + \Delta \sigma_1 \quad 4-283$$

$$\tilde{p}'_{0,2} = p'_{0,n-1} + \Delta p'_{0,1} \quad 4-284$$

$$\tilde{e}_2 = e_n \quad 4-285$$

$$\tilde{\omega}_2 = \omega(\tilde{\sigma}_2, \tilde{e}_2, \tilde{M}_2) \quad 4-286$$

$$\tilde{M}_2 = M(\theta(\tilde{\sigma}_2)) \quad 4-287$$

Уколико није активна “Brick” компонента модела, \tilde{K}_2 и \tilde{G}_2 се рачунају као код HASP-а и $\tilde{\kappa}_2 = \kappa$. У супротном, \tilde{G}_2 се рачуна из вредности \tilde{G}_1 узимајући у обзир само промену нивоа напона:

$$\tilde{G}_2 = \tilde{G}_1 \frac{\tilde{p}'_2}{\tilde{p}'_1} \quad 4-288$$

\tilde{K}_2 се рачуна из израза (3-95), а $\tilde{\kappa}_2$ из израза:

$$\tilde{\kappa}_2 = \frac{1 + \tilde{e}_2}{\tilde{K}_2} \tilde{p}'_2 \quad 4-289$$

Напони и параметар ојачања на крају субинкремента рачунају се према изразима (4-134) и (4-135), Релативна грешка, провера да ли је субинкремент био успешан и одређивање величине следећег субинкремента се ради према истом поступку описаном у потпоглављу 4.2.3.

4.3.2.1 Алгоритам нумеричке интеграције

Алгоритам 11. Нумеричка интеграција HASP-E модела модификованим Euler-овим поступком са аутоматском субинкрементацијом

Корак 1. Познате величине на почетку инкремента: ${}^t\sigma$, ${}^t\varepsilon$, $\Delta\varepsilon$, ${}^{t+\Delta t}\varepsilon$. Уколико су задате вредности G_0^{ref} и $\gamma_{0,7}$, тангентни модули G и K и параметар κ рачунају се према **Алгоритму 8**. У супротном, G и K се рачунају према (4-55) и (4-56). На основу вредности G и K попуњава се матрица D^e . Срачунава се $M({}^t\theta_c)$ из (4-238), (4-239) и (4-240), ${}^t\omega$ из (4-50), (4-52), (4-53) и (4-245), ${}^t p'_0$ из (4-246), ${}^t e$ из (4-46), ${}^{t+\Delta t}e$ из (4-48). Остале напонске и деформацијске инваријанте које су потребне за прорачун одређују се из израза (4-30) – (4-45).

Корак 2. Иницијализација субинкрементације: $T = T_0 = 0$ и $\Delta T_1 = 1$.

Корак 3. Срачунавају се: $\Delta\varepsilon_n$ из (4-120), $\Delta\sigma_1$ и $\Delta p'_{0,1}$ из (4-275) – (4-280), $\Delta\sigma_2$ и $\Delta p'_{0,2}$ из (4-281) – (4-287). Уколико се при прорачуну $\Delta\sigma_1$ и $\Delta p'_{0,1}$ добије $\Delta\lambda_1 < 0$ усваја се $\Delta\lambda_1 = 0$ (деформације су еластичне). Уколико се при прорачуну $\Delta\sigma_2$ и $\Delta p'_{0,2}$ добије $\Delta\lambda_2 < 0$ усваја се $\Delta\lambda_2 = 0$ (деформације су еластичне).

Корак 4. Срачунавају се напони $\hat{\sigma}_n$ из (4-134) и параметар ојачања $\hat{p}'_{0,n}$ из (4-135) на крају субинкремента.

Корак 5. Срачунава се релативна грешка Er_n из (4-149).

Корак 6. Уколико је $Er_n > STOL$ субинкремент се одбацује и потребно је смањити псеудо време ΔT_n . Усвојено је $STOL = 10^{-4}$. Прелази се на Корак 7.

Уколико је $Er_n \leq STOL$ субинкремент се прихвата и иде се на Корак 8.

Корак 7. Потребно је смањити величину субинкремента ΔT_n према (4-151) и (4-152). Понавља се прорачун са новом величином субинкремента. Враћа се на Корак 3.

Корак 8. Пошто је субинкремент био успешан, прихватају се срачунати напони и параметар ојачања. На крају субинкремента напони и параметар ојачања су дати изразима (4-156) и (4-157).

Корак 9. Проверава се да ли су напони и параметар ојачања на крају субинкремента на површи течења према (4-158) у коме је усвојено је $FTOL = 10^{-9}$.

Уколико је испуњен претходни услов, напони и параметар ојачања су на површи течења и не мора да се врши корекција. Прелази се на Корак 10.

У случају да су $\Delta\lambda_1 = 0$ или $\Delta\lambda_2 = 0$ понашање тла је еластично и не треба да се врши корекција.

Уколико није испуњен услов (4-158) потребно је да се изврши корекција напона и параметар ојачања. Корекција се врши преко **Алгоритма 6**.

Корак 10. Одређује се величина наредног субинкремента ΔT_{n+1} преко израза (4-153), (4-154) и (4-155).

Корак 11. Одређује се укупно псеудо време на крају субинкремента преко израза (4-159).

Корак 12. Проверава се да ли је величина наредног субинкремента ΔT_{n+1} мања од минимално прописане вредности ΔT_{min} . Уколико јесте мања онда се величина наредног субинкремента одређује преко (4-160). Усвојено је $\Delta T_{min} = 10^{-6}$.

Корак 13. Проверава се да укупно псеудо време на крају наредног субинкремента не буде веће од 1 према (4-161).

Уколико није задовољен претходни услов, онда се величина наредног субинкремента одређује преко (4-162).

Корак 14. Уколико је $T_n < 1$ започиње се наредни субинкремент и понављају се Кораци 3-13. У супротном, нанет је цео инкремент и срачунати су напони ${}^{t+\Delta t}\sigma$ и параметар ојачања ${}^{t+\Delta t}p'_0$ на крају инкремента. Завршава се прорачун.

Крај Алгоритма 11.

4.3.3 Runge-Kutta-Dormand-Prince поступак са аутоматском субинкрементацијом и контролом грешке

У RKDP методи за интеграцију конститутивних релација HASP-E модела прираштаји напона $\Delta\sigma_i$ и параметра ојачања $\Delta p'_{0,i}$ ($i = 1, 2, 3, 4, 5, 6$) се рачунају преко израза (4-264)-(4-274). Напони $\tilde{\sigma}_i$ се рачунају преко израза (4-183), (4-189), (4-195), (4-201), (4-207) и (4-213). Параметри ојачања $\tilde{p}'_{0,i}$ се рачунају преко израза (4-184), (4-190), (4-196), (4-202), (4-208) и (4-214). Коефицијенти порозности \tilde{e}_i се рачунају према (4-185), (4-191), (4-197), (4-203), (4-209) и (4-215). Коефицијенти ојачања се рачунају према изразу:

$$\tilde{\omega}_i = \omega(\tilde{\sigma}_i, \tilde{e}_i, \tilde{M}_i) \quad 4-290$$

Параметар \tilde{M}_i се рачуна према изразу:

$$\tilde{M}_i = M(\theta(\tilde{\sigma}_i)) \quad 4-291$$

Уколико није активна "Brick" компонента модела, \tilde{K}_i и \tilde{G}_i се рачунају као код HASP-а и $\tilde{\kappa}_i = \kappa$. У супротном, \tilde{G}_i ($i = 2, 3, 4, 5, 6$) се рачуна из вредности \tilde{G}_1 узимајући у обзир само промену нивоа напона:

$$\tilde{G}_i = \tilde{G}_1 \frac{\tilde{p}'_i}{\tilde{p}'_1} \quad 4-292$$

\tilde{G}_1 се рачуна преко **Алгоритма 8**, \tilde{K}_i се рачуна из израза (3-95), а $\tilde{\kappa}_i$ из израза:

$$\tilde{\kappa}_i = \frac{1 + \tilde{e}_i}{\tilde{K}_i} \tilde{p}'_i \quad 4-293$$

Напони и параметар ојачања на крају субинкремента рачунају се према изразима (4-167) и (4-168). Релативна грешка, провера да ли је субинкремент био успешан и одређивање величине следећег субинкремента се ради према истом поступку описаном у потпоглављу 4.2.4.

4.3.3.1 Алгоритам нумеричке интеграције

Алгоритам 12. Нумеричка интеграција HASP-E модела Runge-Kutta-Dormand-Prince поступком са аутоматском субинкрементацијом

Корак 1. Познате величине на почетку инкремента: ${}^t\sigma$, ${}^t\varepsilon$, $\Delta\varepsilon$, ${}^{t+\Delta t}\varepsilon$. Уколико су задате вредности G_0^{ref} и $\gamma_{0,7}$, тангентни модули G и K и параметар κ рачунају се према **Алгоритму 8**. У супротном, G и K се рачунају према (4-55) и (4-56). На основу вредности G и K попуњава се матрица D^{te} . Срачунава се $M({}^t\theta_c)$ из (4-238), (4-239) и (4-240), ${}^t\omega$ из (4-50), (4-52), (4-53) и (4-245), ${}^tp'_0$ из (4-246), te из (4-46), ${}^{t+\Delta t}e$ из (4-48). Остале напонске и деформацијске инваријанте које су потребне за прорачун одређују се из израза (4-30) – (4-45).

Корак 2. Иницијализација субинкрементације: $T = T_0 = 0$ и $\Delta T_1 = 1$.

Корак 3. Срачунавају се: $\Delta\varepsilon_n$ из (4-120), прираштаји $\Delta\sigma_i$ и $\Delta p'_{0,i}$ ($i = 1, 2, 3, 4, 5, 6$) према изразима (4-264) – (4-274). Уколико се при прорачуну $\Delta\sigma_i$, $\Delta p'_{0,i}$ добије $\Delta\lambda_i < 0$ усваја се $\Delta\lambda_i = 0$ (деформације су еластичне).

Корак 4. Срачунавају се напони $\hat{\sigma}_n$ из (4-167) и параметар ојачања $\hat{p}'_{0,n}$ из (4-168) на крају субинкремента.

Корак 5. Срачунава се релативна грешка Er_n из (4-217) и (4-218).

Корак 6. Уколико је $Er_n > STOL$ субинкремент се одбацује и потребно је смањити псеудо време ΔT_n . За $STOL$ се може усвојити вредност мања од 10^{-4} како би се повећала тачност поступка интеграције. Прелази се на Корак 7.

Уколико је $Er_n \leq STOL$ субинкремент се прихвата и иде се на Корак 8.

Корак 7. Потребно је смањити величину субинкремента ΔT_n према (4-151) и (4-219). Понавља се прорачун са новом величином субинкремента. Враћа се на Корак 3.

Корак 8. Пошто је субинкремент био успешан, прихватају се срачунати напони и параметар ојачања. На крају субинкремента напони и параметар ојачања су дати изразима (4-222) и (4-223).

Корак 9. Проверава се да ли су напони и параметар ојачања на крају субинкремента на површи течења према изразу (4-224) у коме је усвојено $FTOL = 10^{-9}$.

Уколико је испуњен претходни услов, напони и параметар ојачања су на површи течења и не мора да се врши корекција. Прелази се на Корак 10.

У случају да је било који $\Delta\lambda_i = 0$ ($i = 1, 2, 3, 4, 5, 6$) понашање тла је еластично и не треба да се врши корекција.

Уколико није испуњен услов (4-158) потребно је да се изврши корекција напона и параметра ојачања. Корекција се врши преко **Алгоритма 6**.

Корак 10. Одређује се величина наредног субинкремента ΔT_{n+1} преко израза (4-153), (4-220) и (4-221).

Корак 11. Одређује се укупно псеудо време на крају субинкремента према (4-225).

Корак 12. Проверава се да ли је величина наредног субинкремента ΔT_{n+1} мања од минимално прописане вредности ΔT_{min} . Уколико јесте мања величина наредног субинкремента одређује према (4-226). Усвојено је $\Delta T_{min} = 10^{-6}$.

Корак 13. Проверава се да укупно псеудо време на крају наредног субинкремента не буде веће од 1 према услову (4-227). Уколико није задовољен претходни услов, онда је величина наредног субинкремента дата (4-228).

Корак 14. Уколико је $T_n < 1$ започиње се наредни субинкремент и понављају се Кораци 3-13. У супротном, нанет је цео инкремент и срачунати су напони ${}^{t+\Delta t}\sigma$ и параметар ојачања ${}^{t+\Delta t}p'_0$ на крају инкремента. Завршава се прорачун.

Крај Алгоритма 12.

4.4 Верификација поступка имплементације конститутивних модела

Конститутивни модели HASP и HASP-E су имплементирани у PLAXIS 2D преко алгоритама који су приказани у претходна два потпоглавља. Верификација поступка имплементације оба модела је извршена поређењем резултата симулације триаксијалних опита на Cardiff глини (CU опит) (Banerjee & Stipho, 1978) и Fujinomori глини (CD опит) (Nakai & Hinokio, 2004) у PLAXIS-у 2D са резултатима за Euler-ову интеграцију деформација дуж познате путање напона (EIPN) који су преузети из (Josković, 2017) и са резултатима предвиђања OC_CLAY модела. Приликом симулације ова два опита, Brick компонента HASP-E модела није била активна и у том случају HASP и HASP-E модели треба да предвиђају исто понашање глине. Такође, SSO модел у OC_CLAY моделу није био активан. У HASP моделу се вредност параметра M бира тако да одговара условима триаксијалне компресије или триаксијалне екстензије у зависности од тога који се од та два опита симулира, Са друге стране у HASP-E моделу се као улазни подаци задају вредности параметра M у триаксијалној компресији (M_c) и триаксијалној екстензији (M_e) и модел на основу стања напона одређује која је тренутна вредност параметра M .

Верификација поступка имплементације HASP-E модела је извршена поређењем резултата симулације цикличног триаксијалног опита на Fujinomori глини (CD опит) (Nakai & Hinokio, 2004) и Newfield глини (CU опит) (Sangrey, et al., 1969) у PLAXIS-у 2D са резултатима предвиђања OC_CLAY модела.

За приказивање резултата на дијаграмима у овом потпоглављу коришћене су следеће ознаке:

-за HASP модел имплементиран методом водећег параметра: HASP_GPM

-за HASP модел имплементиран методом (Borja & Lee, 1990): HASP_BL

-за HASP модел имплементиран модификованим Euler-овим поступком са аутоматском субинкрементацијом: HASP_modEuler

-за HASP модел имплементиран RKDP поступком са аутоматском субинкрементацијом: HASP_RKDP

-за HASP-E модел уколико “Brick” компонента није активна имплементиран методом водећег параметра: HASP-E*_GPM

-за HASP-E модел уколико “Brick” компонента није активна имплементиран модификованим Euler-овим поступком са аутоматском субинкрементацијом: HASP-E*_modEuler

-за HASP-E модел уколико “Brick” компонента није активна имплементиран RKDP поступком са аутоматском субинкрементацијом: HASP-E*_RKDP

-за HASP-E модел са активном “Brick” компонентом методом водећег параметра: HASP-E_GPM

-за HASP-E модел са активном “Brick” компонентом имплементиран модификованим Euler-овим поступком са аутоматском субинкрементацијом: HASP-E_modEuler

-за HASP-E модел са активном “Brick” компонентом имплементиран RKDP поступком са аутоматском субинкрементацијом: HASP-E_RKDP

-за OC_CLAY модел са неактивним SSO моделом: OC_CLAY

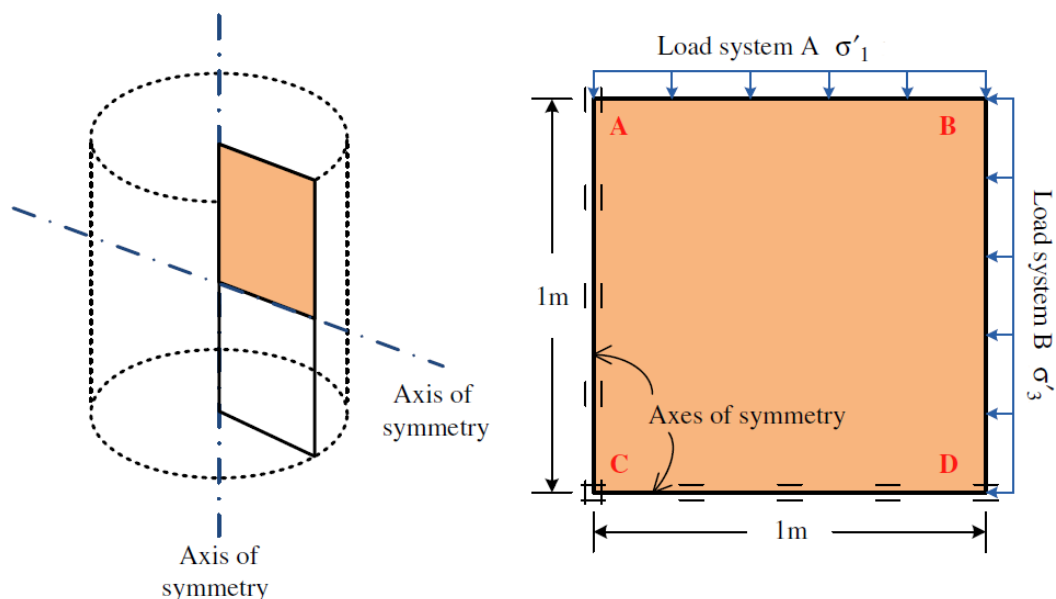
-за OC_CLAY модел са активним SSO моделом: OC_CLAY_SSO

За експлицитне методе интеграције коришћена је вредност толеранције на локалну грешку од $STOL = 10^{-4}$ како би се скратило трајање прорачуна. У свим симулацијама оптерећење је нането у 100 инкремената.

У свим нумеричким симулацијама у овом потпоглављу у OC_CLAY моделу следећи параметри модела имају вредности: $p'_{ref} = 100 \text{ kPa}$, $h = 1.0$ и $elReload = 0$

4.4.1 Модел триаксијалног опита у PLAXIS-у 2D

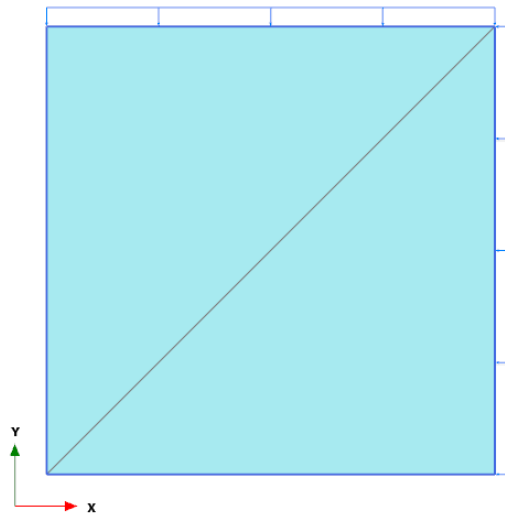
Триаксијални опит је симулиран у PLAXIS-у 2D користећи стање осне симетрије (axisymmetric). Димензије модела и гранични услови за триаксијални опит приказани су на **Слика 4-5**. Услед осне симетрије, довољно је анализирати само четвртину узорка. Димензије узорка тла у нумеричком моделу су $1 \text{ m} \times 1 \text{ m}$, али тако нереалистично велике димензије не утичу на добијене резултате, јер је тло моделирано без запреминске тежине. У наставку су описани детаљи нумеричког модела.



Слика 4-5 Модел триаксијалног опита у PLAXIS-у 2D – преузето из (Surarak, et al., 2012)

Као што је приказано на **Слика 4-5**, аксијални напони σ'_1 се наносе као расподељено оптерећење на горњу ивицу модела, док се бочни напони $\sigma'_2 = \sigma'_3 = \sigma'_r$ наносе на десну ивицу модела као расподељено оптерећење. Померања у оба правца на десној и горњој ивици су слободна (Free). На доњој ивици су спречена вертикална померања, а на левој ивици су спречена хоризонтална померања. Коришћена је веома груба мрежа са само два троугаона коначна елемента са 15 чворова (**Слика 4-6**). У консолидованом дренаiranом триаксијалном опиту (CD опит) понашање тла је дренаирано (Drainage type: Drained), док је у консолидованом недренаiranом опиту (CU) понашање тла недренаирано типа А (Drainage type: Undrained A). Тип А недренаиране анализе је описан у потпоглављу 2.7.4. За оба опита, тип анализе је еласто-пластична напонска анализа (Plastic analysis). Недренаирана еласто-пластична анализа типа А у PLAXIS-у 2D омогућава потпуни развој додатних порних притисака и струјање воде у порама тла се потпуно занемарује, тако да није потребно задавати коефицијент водопропусности тла.

У симулацији триаксијалних опита увек је разматрано минимално три фазе прорачуна. Прва фаза је иницијална фаза у којој се генеришу почетни напони и порни притисци у тлу (Initial phase). Пошто тло нема тежину и није задат ниво воде, почетни напони и порни притисци су једнаки нули. Друга фаза је изотропна компресија у којем се наноси свестрани притисак на горњој и десној ивици модела ($\sigma'_1 = \sigma'_2 = \sigma'_3 = p'_i$). Пошто су почетни напони у иницијалној фази једнаки нули и конститутивни модели (HASP и HASP-E) су тако имплементирани да не могу да покрену прорачун ако су почетни напони нула, тло је у овој фази моделирано као линеарно еластичан материјал. Пошто је нането оптерећење изотропно и без обзира које су вредности еластичних параметара коришћени, у било којој тачки тла нормални напони ће бити међусобно једнаки ($\sigma'_x = \sigma'_y = \sigma'_z = p'_i$) и смичући напони су једнаки нули ($\tau_{xy} = 0$). Да не би дошло до генерисања додатних порних притисака у овој фази, понашање тла је увек дренаирано. На почетку треће фазе, ресетују се померања, деформације и параметри стања и тло се моделира са одговарајућим конститутивним моделом и типом дренаиране анализе.



Слика 4-6 Мрежа коначних елемената за модел триаксијалног опита у PLAXIS-у 2D

У трећој фази у опиту триаксијалне компресије вредност вертикалног ефективног напона σ'_1 се повећава до лома или док се не достигне одређен ниво аксијалних деформација (20 % – 25 %), док се вредност бочних ефективних напона σ'_r не мења. У трећој фази у опиту триаксијалне екстензије вредност вертикалног ефективног напона σ'_1 се смањује до лома или док се не достигне одређен ниво аксијалних деформација, док се вредност бочних ефективних напона σ'_r не мења. У трећој фази у опитима са константним ефективним напоном ($p' = const$) у триаксијалној компресији вредност вертикалног ефективног напона σ'_1 се повећава док се вредност бочних ефективних напона σ'_r смањује до лома или док се не достигне одређен ниво аксијалних деформација. У трећој фази у опитима са константним ефективним напоном ($p' = const$) у триаксијалној екстензији вредност вертикалног ефективног напона σ'_1 се смањује док се вредност бочних ефективних напона σ'_r повећава до лома или док се не достигне одређен ниво аксијалних деформација.

У цикличним триаксијалним опитима након друге фазе постоји већи број фаза у којима се вертикални ефективни напон σ'_1 у једној фази повећава, а у наредној смањује до неке вредности. У цикличним триаксијалним опитима са константним средњим ефективним напоном се у једној фази вертикални ефективни напон σ'_1 се повећава или смањује док се бочни ефективни напони σ'_r смањују или повећавају, а у наредној фази обрнуто.

Пошто сваки опит креће од фазе изотропне компресије онда су коефицијенти бочних притисака тла $K_0^{NC} = 1.0$.

4.4.2 CU триаксијални опити на Cardiff глини

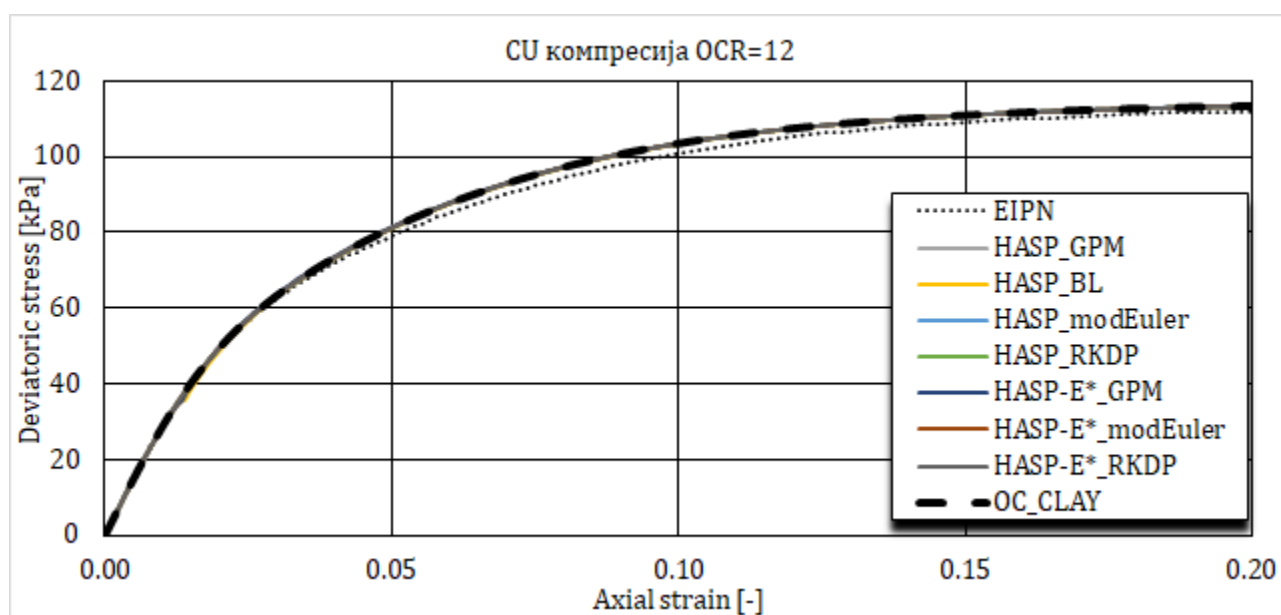
Иницијални параметри за симулацију CU триаксијалних опита на Cardiff глини дати су у **Табела 4-1**. Вредности иницијалног коефицијента порозности су одређене из познатог положаја линије критичног стања у компресионој равни Γ који је публикован у литератури у складу са изразом (3-48) на основу публикованог степена преко консолидације и иницијалног напонског стања.

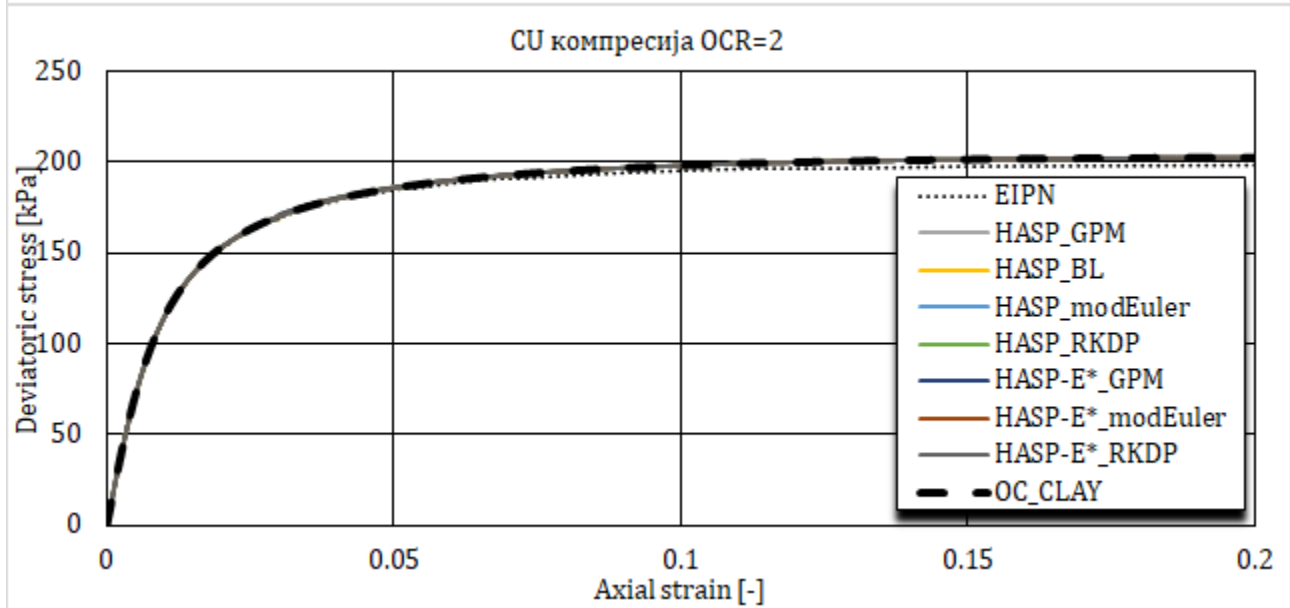
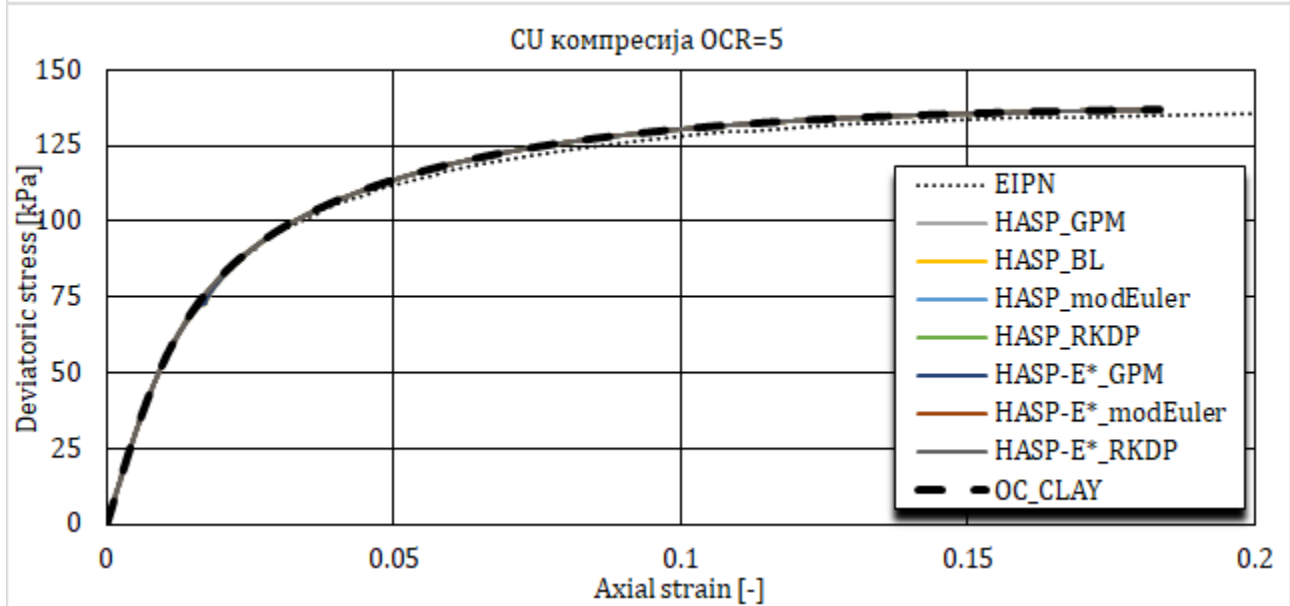
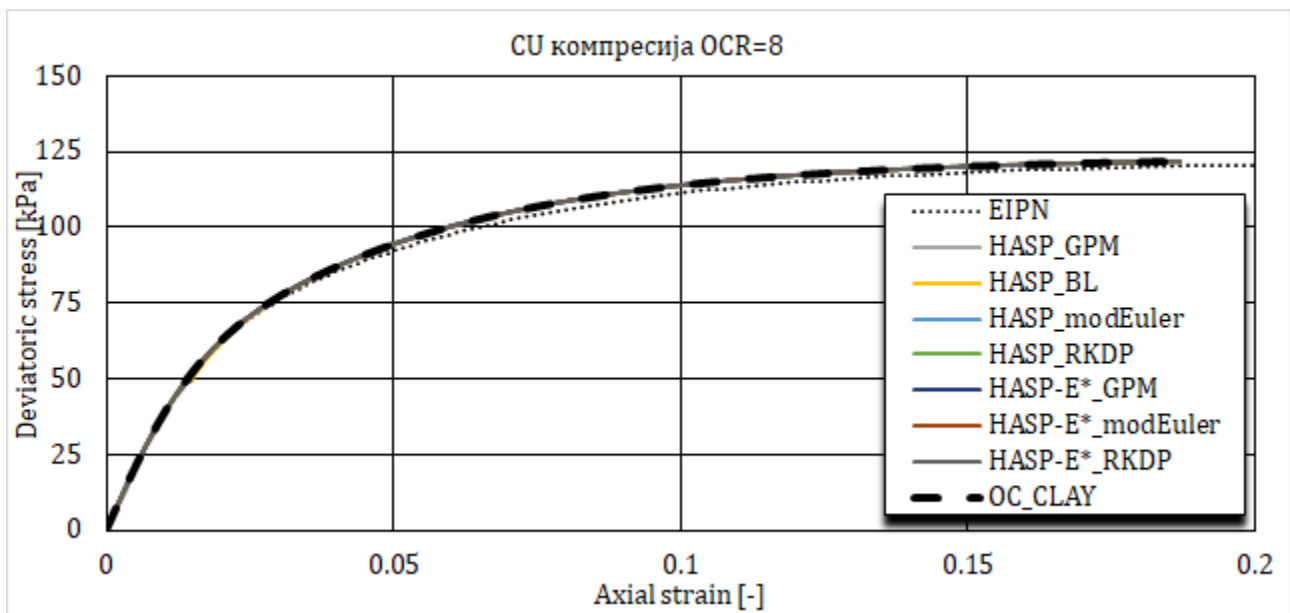
Табела 4-1 Иницијални услови за симулацију CU триаксијалних опита на Cardiff глини

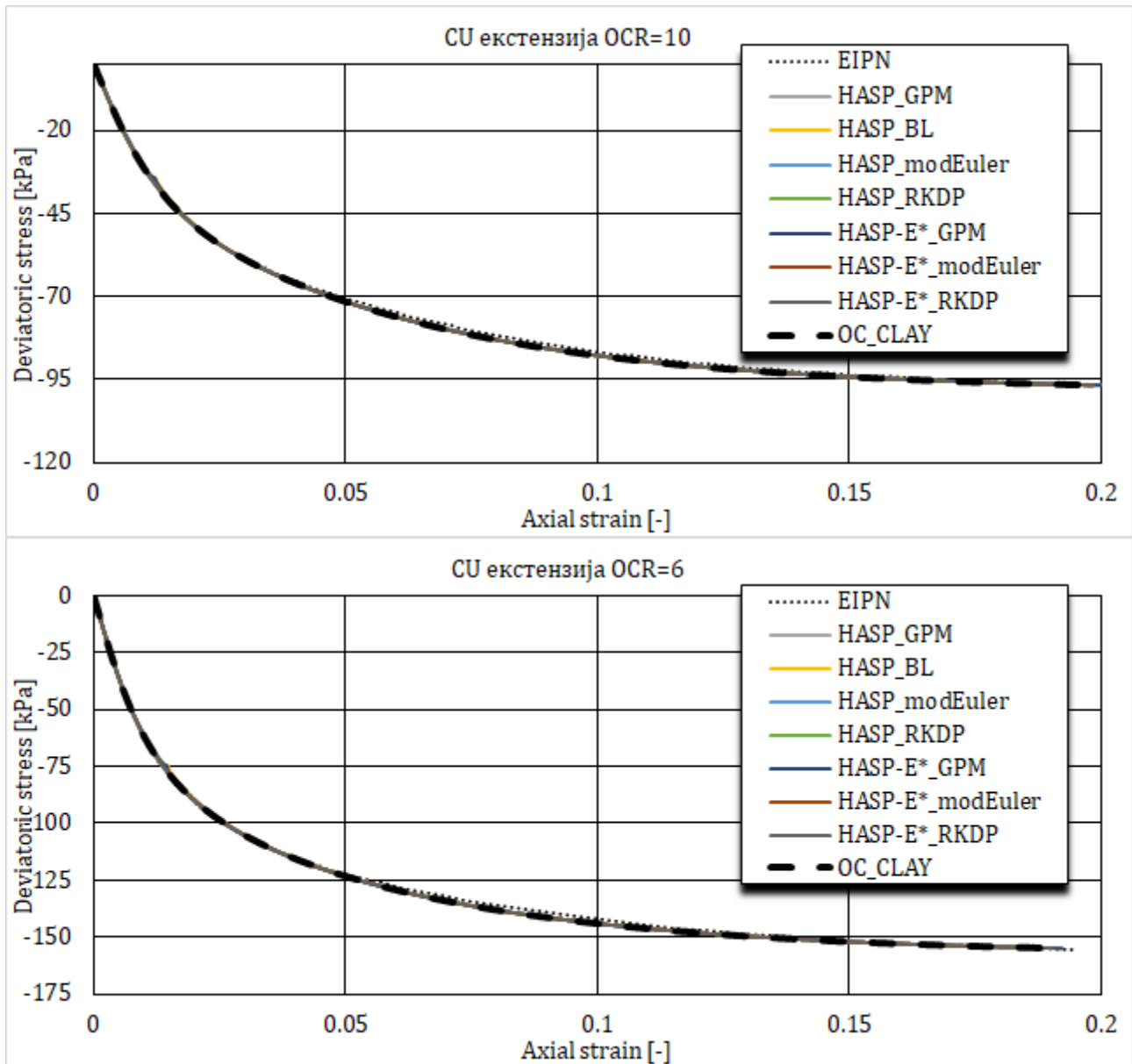
Врста оптерећивања	$OCR[-]$	$p'_i [kPa]$	$e_o [-]$	$\Gamma [-]$
компресија	12	34.5	0.973	2.63
	8	48.2	0.963	2.63
	5	73	0.947	2.63
	2	193	0.893	2.63
екстензија	10	41.5	0.963	2.629
	6	92	0.895	2.627

Табела 4-2 Параметри HASP, HASP-E и OC_CLAY модела за симулацију CU триаксијалних опита на Cardiff глини

Модел	Параметри							
HASP	λ	κ	M			μ		
	$[-]$	$[-]$	$[-]$			$[-]$		
компресија	0.140	0.050	1.05			0.2		
екстензија			0.85					
HASP-E	λ	κ	M_c	M_e	μ	G_0^{ref}	$\gamma_{0,7}$	
	$[-]$	$[-]$	$[-]$	$[-]$	$[-]$	$[kPa]$	$[-]$	
компресија	0.140	0.050	1.05	0.85	0.2	-	-	
екстензија								
OC_CLAY	OCR	E_{ur}^{ref}	μ	E_{oed}^{ref}	G_0^{ref}	$\gamma_{0,7}$	ϕ_{cs}	POP
	$[-]$	$[kPa]$	$[-]$	$[kPa]$	$[kPa]$	$[-]$	$[\circ]$	$[kPa]$
компресија	12	7102.8	0.2	1409.29	-	-	26.54	379.5
	8	7066.8	0.2	1402.14	-	-	26.54	337.4
	5	7009.2	0.2	1390.71	-	-	26.54	292
	2	6814.8	0.2	1352.14	-	-	26.54	193
екстензија	10	7066.8	0.2	1402.14	-	-	29.68	373.5
	6	6822	0.2	1353.57	-	-	29.68	460



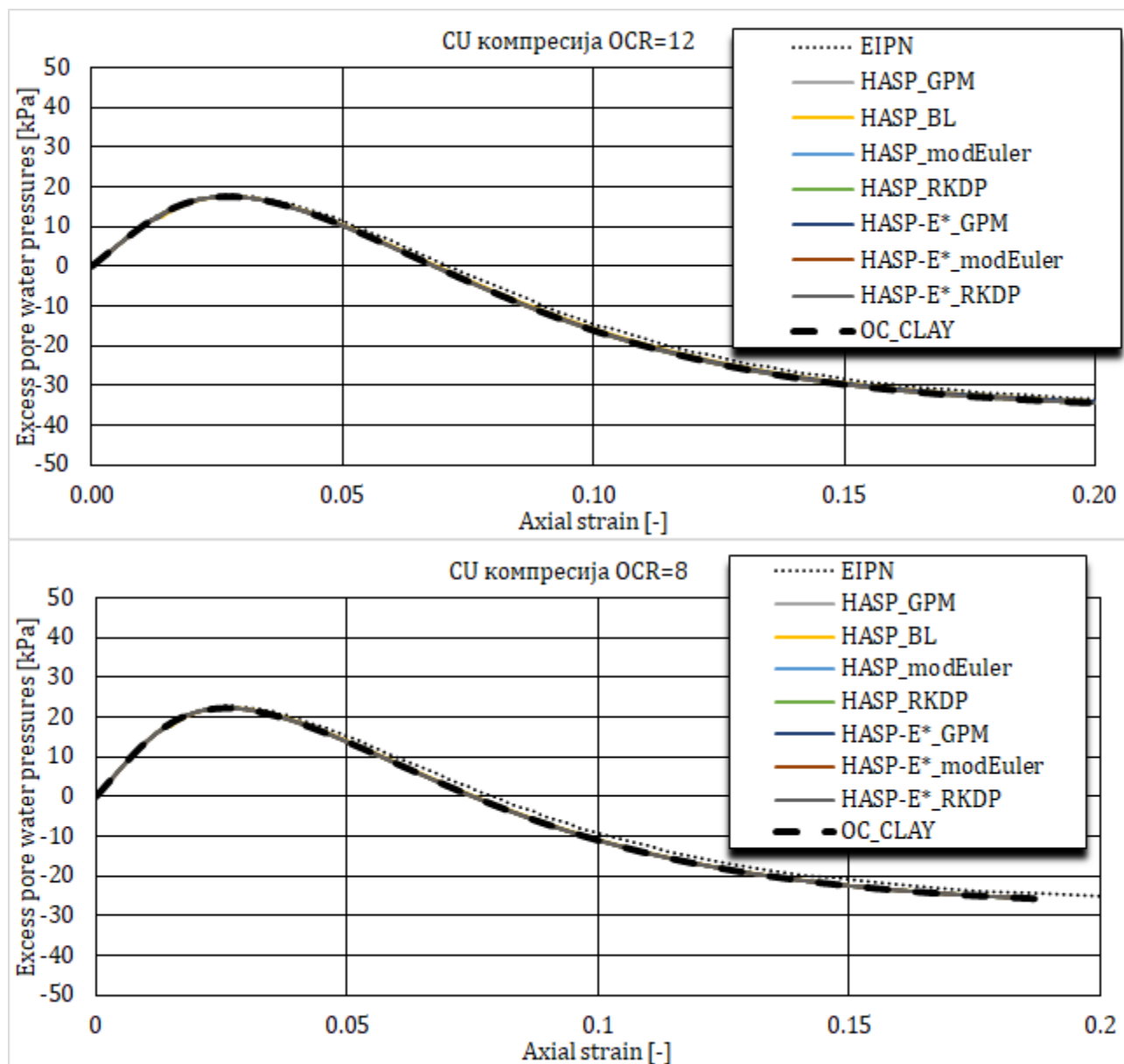


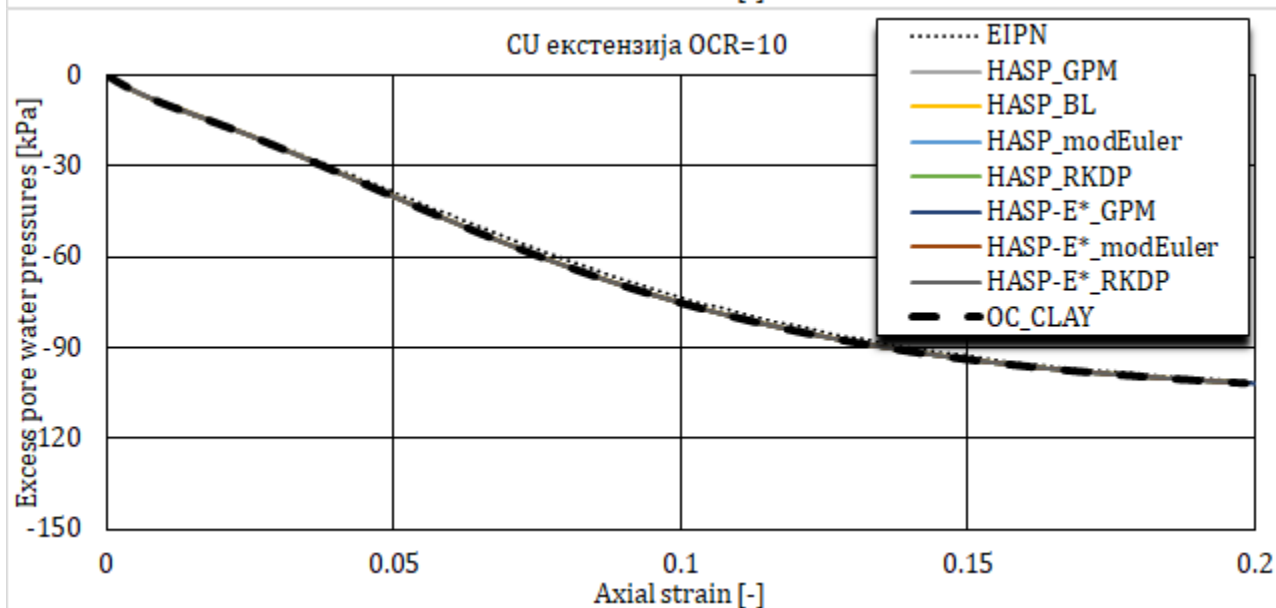
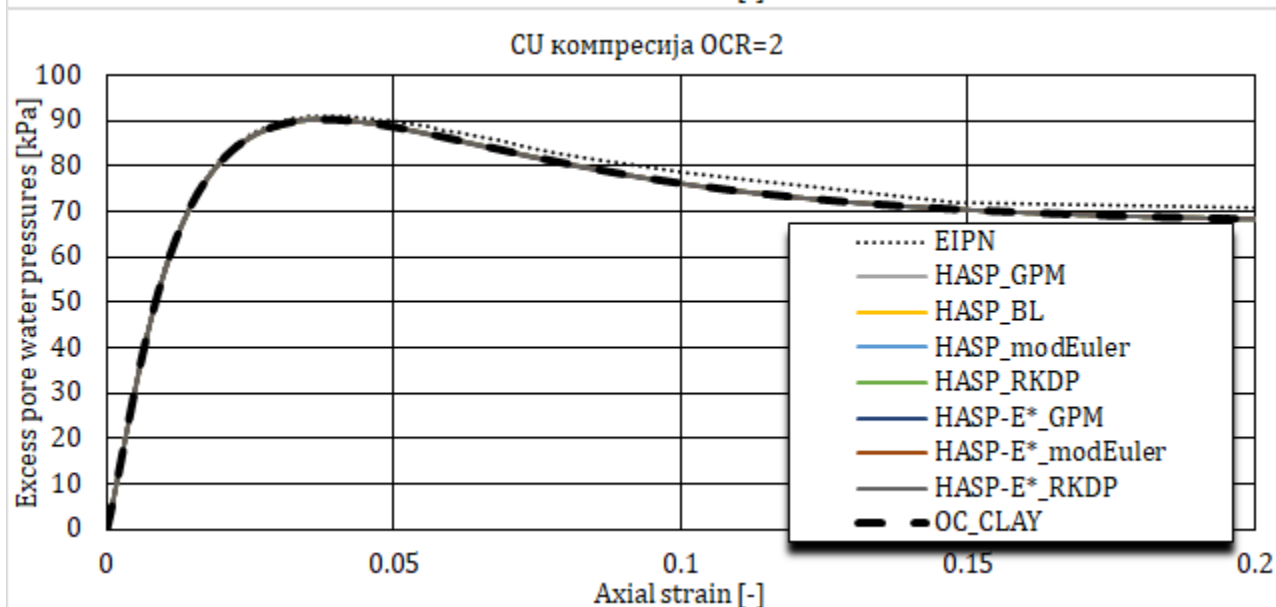
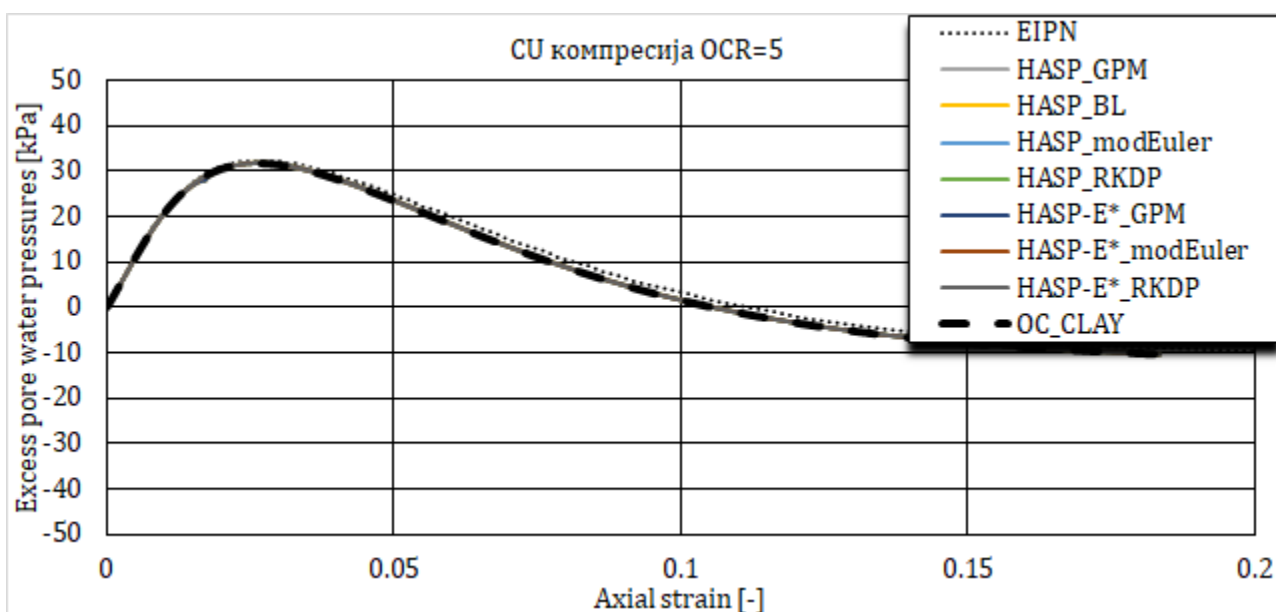


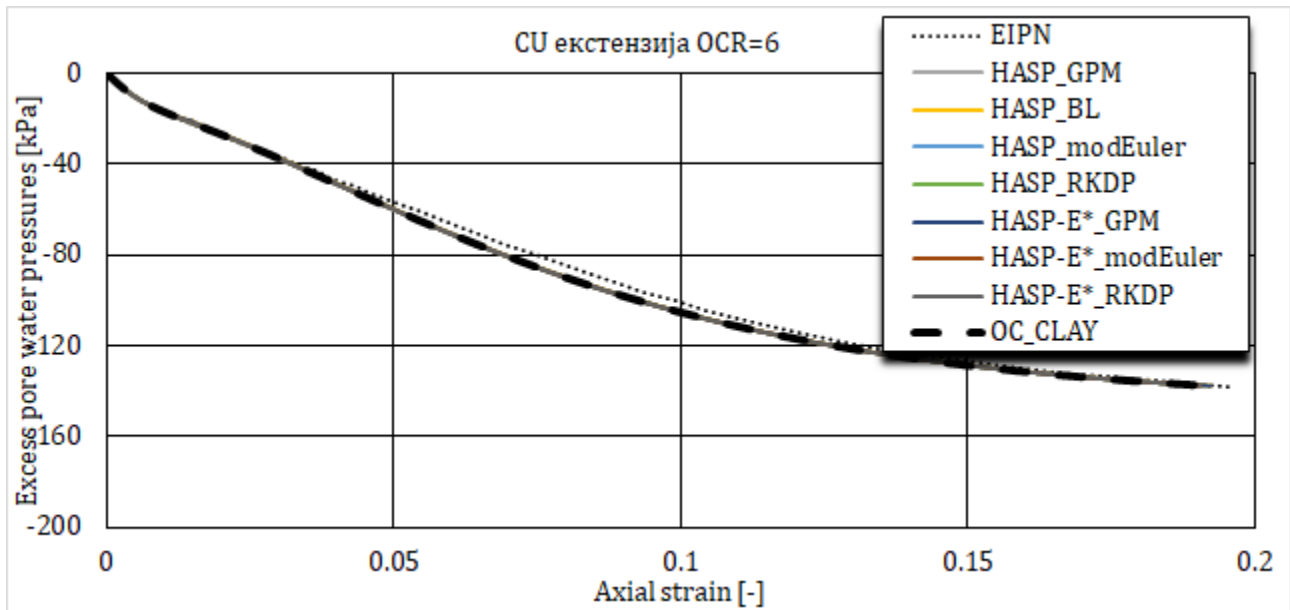
Слика 4-7 Напонско-деформацијске криве за CU триаксијалне опите на Cardiff глини

Параметри конститутивних модела потребни за нумеричке симулације дати су у **Табела 4-2**. За HASP и HASP-E моделе, параметри су преузети из (Јосковић, 2017) и (Banerjee & Stipho, 1978). За OC_CLAY модел, параметри су одређени на основу израза датих у потпоглављу 3.4. “Brick” компонента HASP-E и SSO модел у OC_CLAY моделу нису били активни, због чега нису били потребни параметри G_0^{ref} и $\gamma_{0,7}$. Симулирано је укупно шест опита: четири недренирана опита триаксијалне компресије са степенима преконсолидације 2, 5, 8, 12 и два недренирана опита триаксијалне екстензије са степенима преконсолидације 6 и 10. Резултати нумеричких симулација опита дати су на **Слика 4-7** и **Слика 4-8**. Резултати су приказани за четири различите методе нумеричке интеграције за HASP модел, три различите методе нумеричке интеграције за HASP-E модел и OC_CLAY модел и упоређени су са резултатима EIPN нумеричке интеграције који су преузети из (Јосковић, 2017). Вредности девијаторског напона при лому (достигнута аксијална деформација од 20 %) и екстремне вредности генерисаних порних притисака приказани су у **Табела 4-3** и **Табела 4-4**.

Према приказаним резултатима све методе нумеричке интеграције HASP и HASP-E се изузетно добро поклапају са резултатима EIPN методе нумеричке интеграције који су коришћени за верификацију имплементације HASP модела у (Jocković, 2017), као и са предвиђањима OC_CLAY модела. За HASP и OC_CLAY је било потребно унети вредност параметра M тако да одговара условима триаксијалне компресије или екстензије како би се добило адекватно предвиђање понашање глине. Са друге стране, у HASP-E моделу су била задате обе вредности (M_c и M_e) и модел је на основу стања напона адекватно предвидео понашање глине. Из приказаних резултата се може још уочити да оба HASP модела и OC_CLAY предвиђају идентично понашање тла.







Слика 4-8 Генерисани порни притисци за CU триаксијалне опите на Cardiff глини

Табела 4-3 Девијаторски напон при лому у kPa за CU триаксијалне опите на Cardiff глини

Метода нумеричке интеграције	Компресија				Екстензија	
	OCR = 12	OCR = 8	OCR = 5	OCR = 2	OCR = 10	OCR = 6
EIPN	114.0	121.49	135.59	198.76	-98.30	-155.40
HASP_GPM	114.26	121.88	136.93	202.55	-97.70	-154.79
HASP_BL	114.26	121.86	136.93	202.55	-96.65	-154.86
HASP_modEuler	114.30	122.01	136.99	202.54	-96.69	-154.80
HASP_RKDP	114.30	122.01	136.99	202.54	-96.69	-154.80
HASP-E*_GPM	114.26	121.88	136.93	202.55	-96.72	-154.88
HASP-E*_modEuler	114.30	122.01	136.99	202.54	-96.69	-154.80
HASP-E*_RKDP	114.30	122.01	136.99	202.54	-96.69	-154.80
OC_CLAY	114.30	122.01	136.99	202.54	-96.69	-154.79

Табела 4-4 Максимална вредност генерисаних порних притисака у kPa за CU триаксијалне опите на Cardiff глини

Метода нумеричке интеграције	Компресија				Екстензија	
	OCR = 12	OCR = 8	OCR = 5	OCR = 2	OCR = 10	OCR = 6
EIPN	17.80	22.67	32.21	91.05	-101.43	-138.14
HASP_GPM	17.53	22.33	31.83	90.34	-101.64	-137.44
HASP_BL	17.54	22.33	31.83	90.35	-101.60	-137.66
HASP_modEuler	17.39	22.21	31.68	90.23	-101.71	-137.45
HASP_RKDP	17.39	22.21	31.68	90.23	-101.71	-137.45
HASP-E*_GPM	17.53	22.33	31.83	90.34	-101.82	-137.70
HASP-E*_modEuler	17.39	22.21	31.68	90.23	-101.71	-137.46
HASP-E*_RKDP	17.39	22.22	31.68	90.23	-101.71	-137.45

OC_CLAY	17.39	22.21	31.68	90.23	-101.70	-137.44
---------	-------	-------	-------	-------	---------	---------

4.4.3 CD триаксијални опити на Fujiomori глини

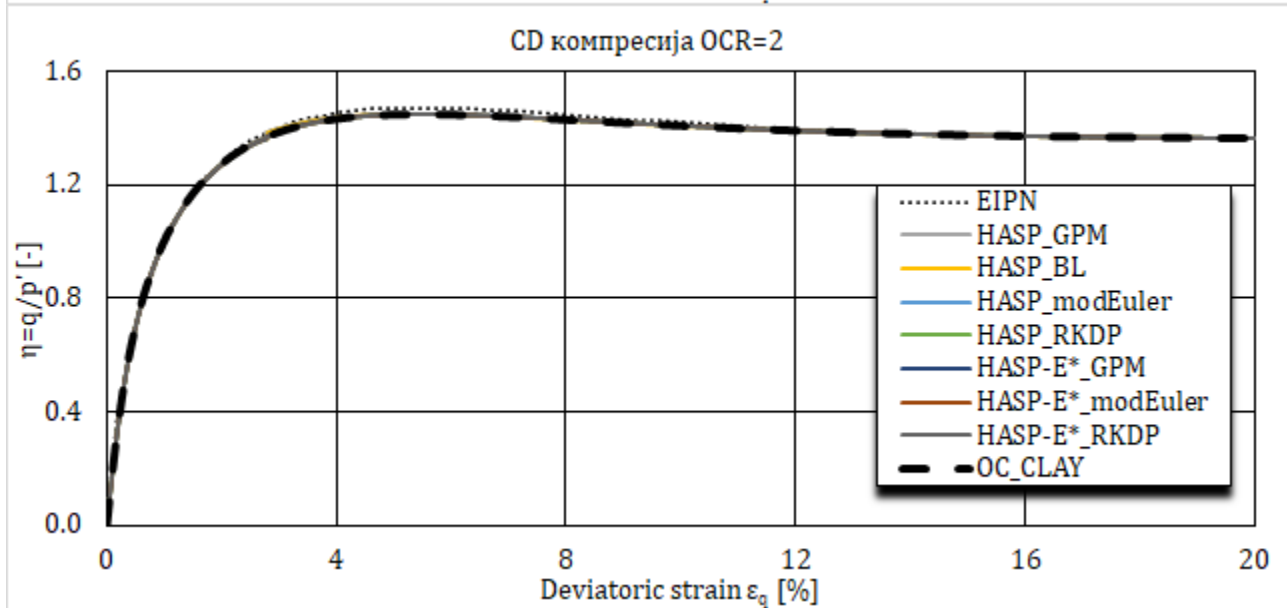
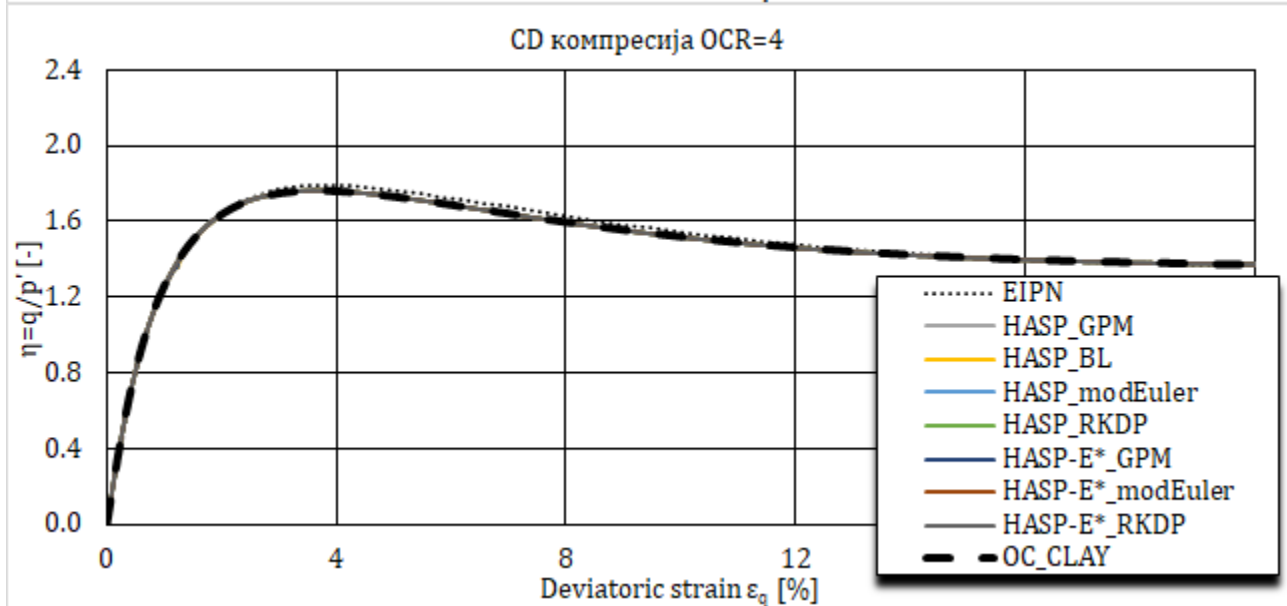
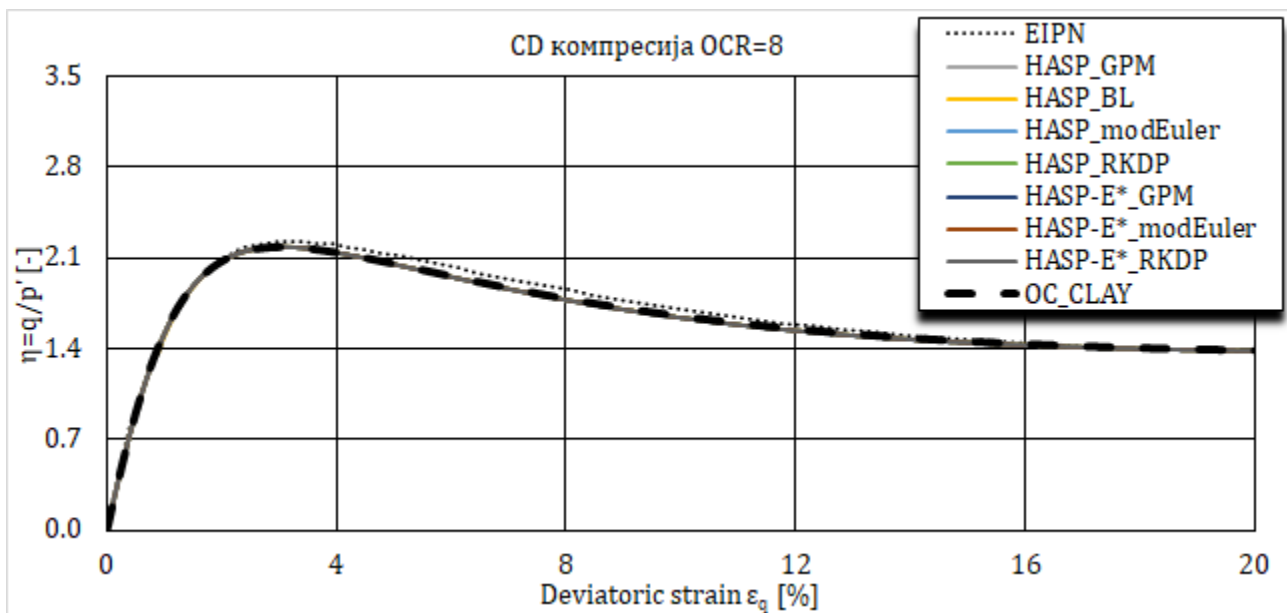
Иницијални услови за симулацију CD опита триаксијалне компресије и екстензије са константним средњим ефективним напоном дати су у **Табела 4-5**. Иницијални коефицијент порозности је одређен у складу са изразом (3-48) за познату вредност параметра Γ , степен преконсолидације и иницијални средњи ефективни напон. У **Табела 4-6** дати су параметри конститутивних модела. За HASP моделе параметри су преузети из (Nakai & Hinokio, 2004) и (Jocković, 2017), док су параметри OC_CLAY модела одређени на основу израза датих у потпоглављу 3.4. "Brick" компонента HASP-E и SSO модел у OC_CLAY моделу нису били активни. Спроведене су нумеричке симулације укупно осам опита: четири опита триаксијалне компресије и четири опита триаксијалне екстензије. Степени преконсолидације глине су 8, 4, 2 и 1. У свим опитима, средњи ефективни напон је имао константну вредност: $p' = 98 \text{ kPa}$ за узорак са $OCR = 8$ и $p' = 196 \text{ kPa}$ за остале степене преконсолидације.

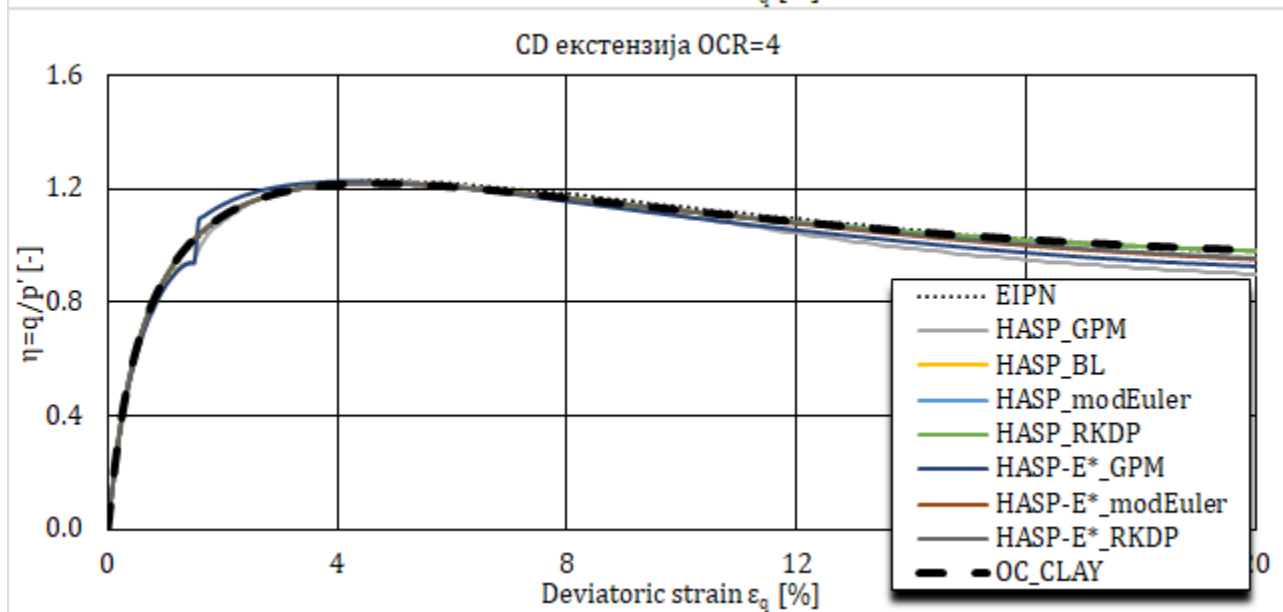
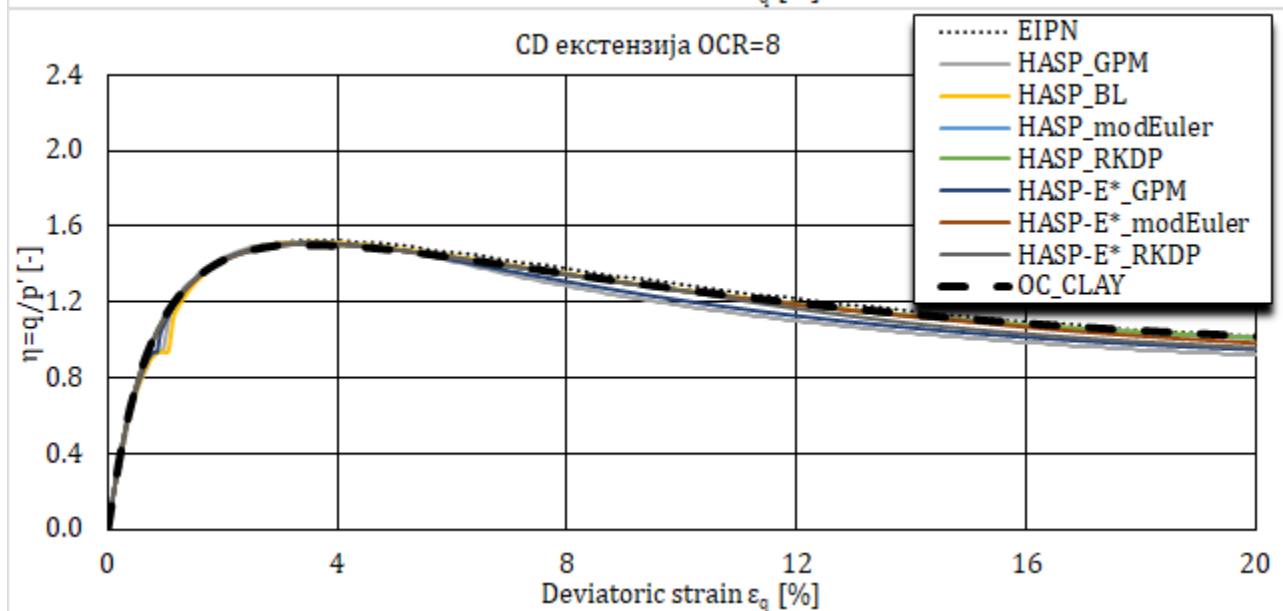
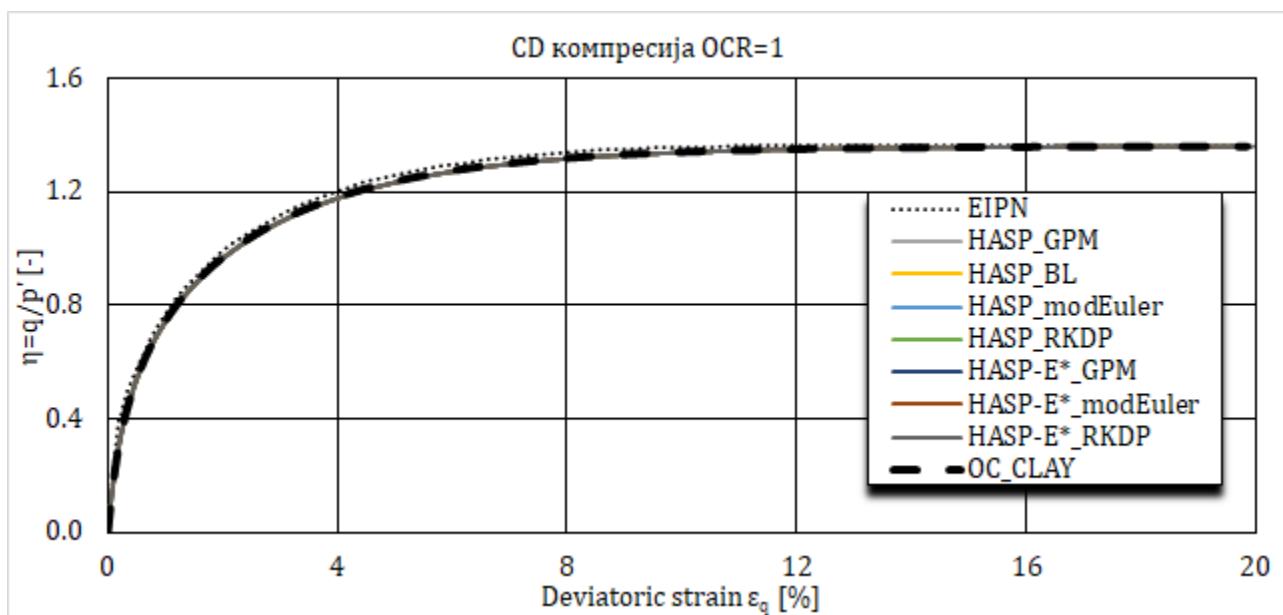
Табела 4-5 Иницијални услови за симулацију CD триаксијалних опита на Fujiomori глини

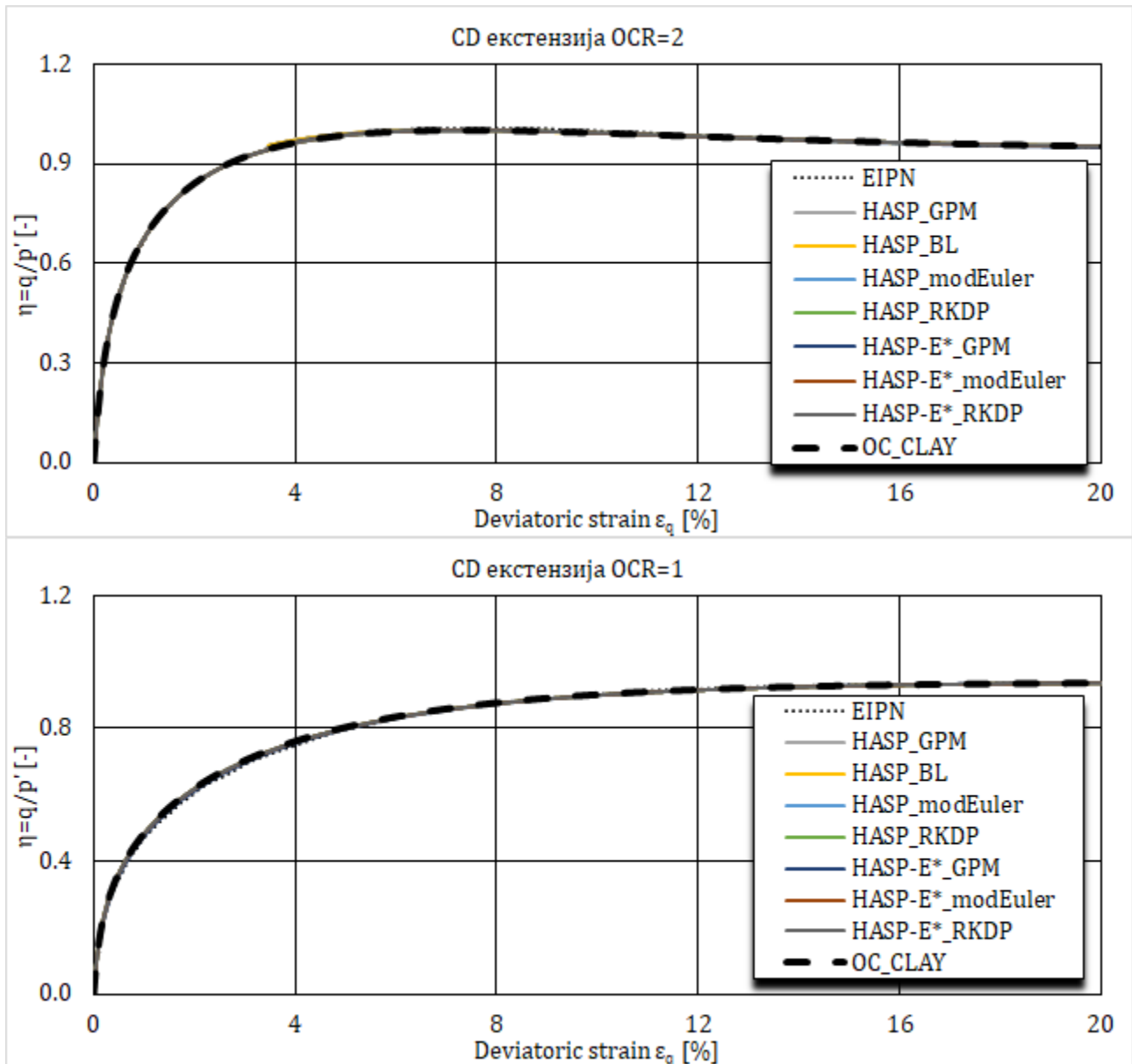
Врста оптерећивања	$OCR[-]$	$p'_i [\text{kPa}]$	$e_o [-]$	$\Gamma [-]$
компресија и екстензија	8	98	0.727	2.23
	4	196	0.713	2.23
	2	196	0.760	2.23
	1	196	0.81	2.23

Табела 4-6 Параметри HASP, HASP-E и OC_CLAY модела за симулацију CD триаксијалних опита на Fujiomori глини

Модел	Параметри							
HASP	λ [-]	κ [-]	M [-]	μ [-]				
компресија екстензија	0.089	0.02	1.36	0.2				
			0.94					
HASP-E	λ [-]	κ [-]	M_c [-]	M_e [-]	μ [-]	G_0^{ref} [kPa]	$\gamma_{0,7}$ [-]	
	0.89	0.02	1.36	0.94	0.2	-	-	
OC_CLAY	OCR [-]	E_{ur}^{ref} [kPa]	μ [-]	E_{oed}^{ref} [kPa]	G_0^{ref} [kPa]	$\gamma_{0,7}$ [-]	ϕ_{cs} [°]	POP [kPa]
компресија	8	15543	0.2	1940.45	-	-	33.67	686
	4	15417	0.2	1924.72	-	-	33.67	588
	2	15840	0.2	1977.53	-	-	33.67	196
	1	16290	0.2	2033.71	-	-	33.67	0
екстензија	8	15543	0.2	1940.45	-	-	33.87	686
	4	15417	0.2	1924.72	-	-	33.87	588
	2	15840	0.2	1977.53	-	-	33.87	196
	1	16290	0.2	2033.71	-	-	33.87	0

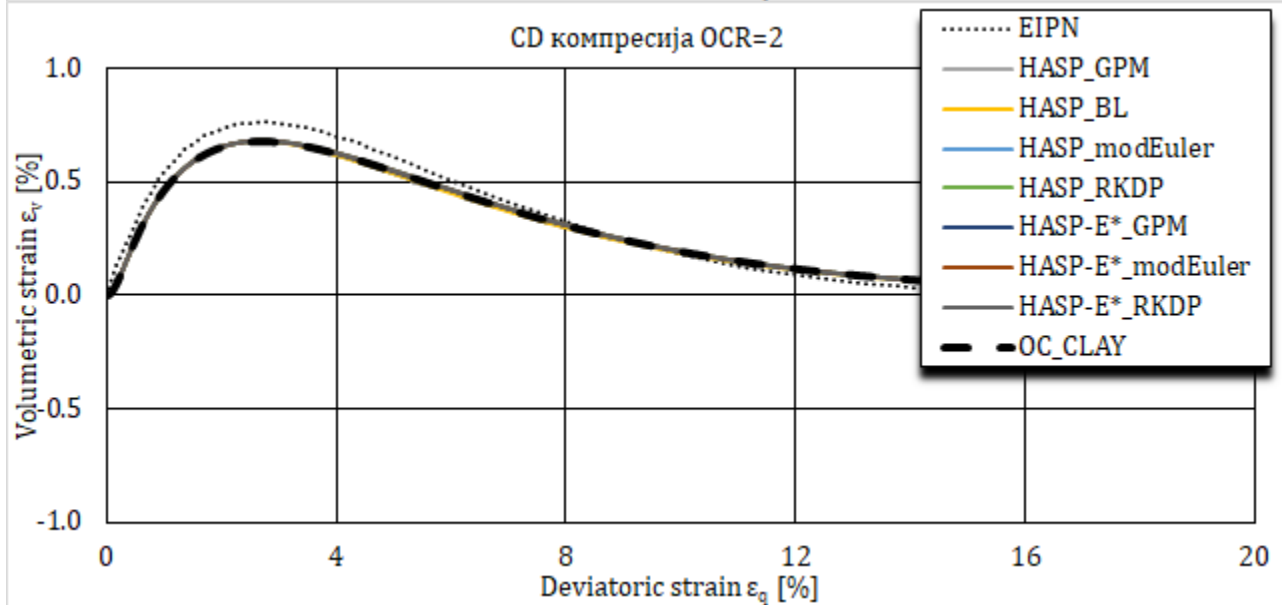
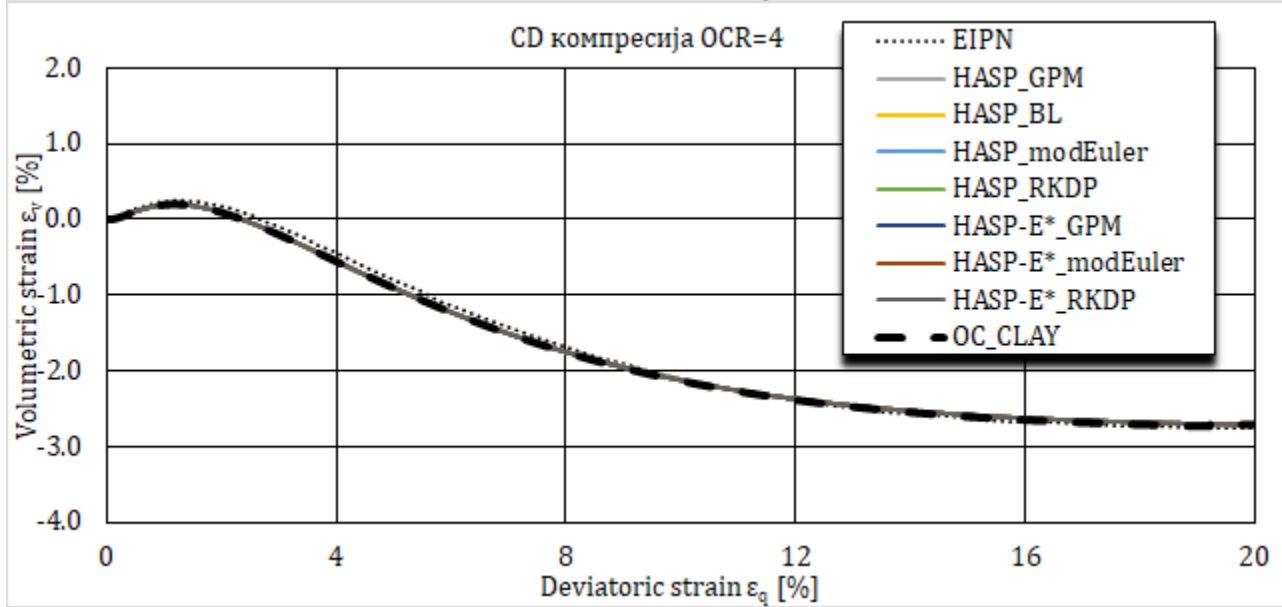
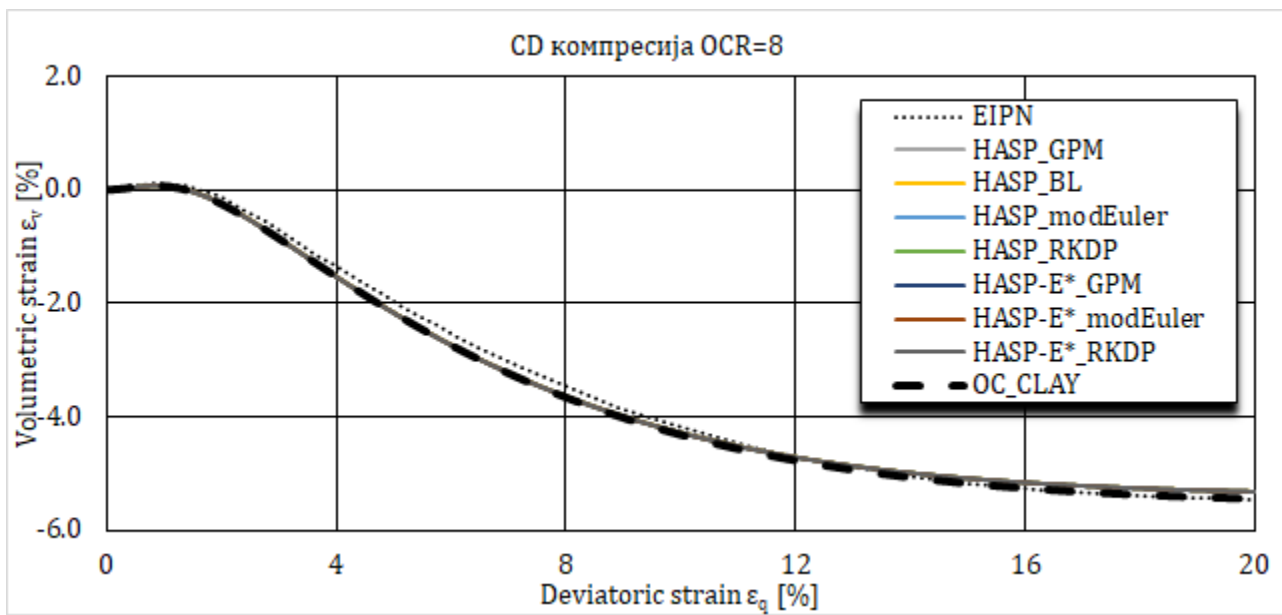


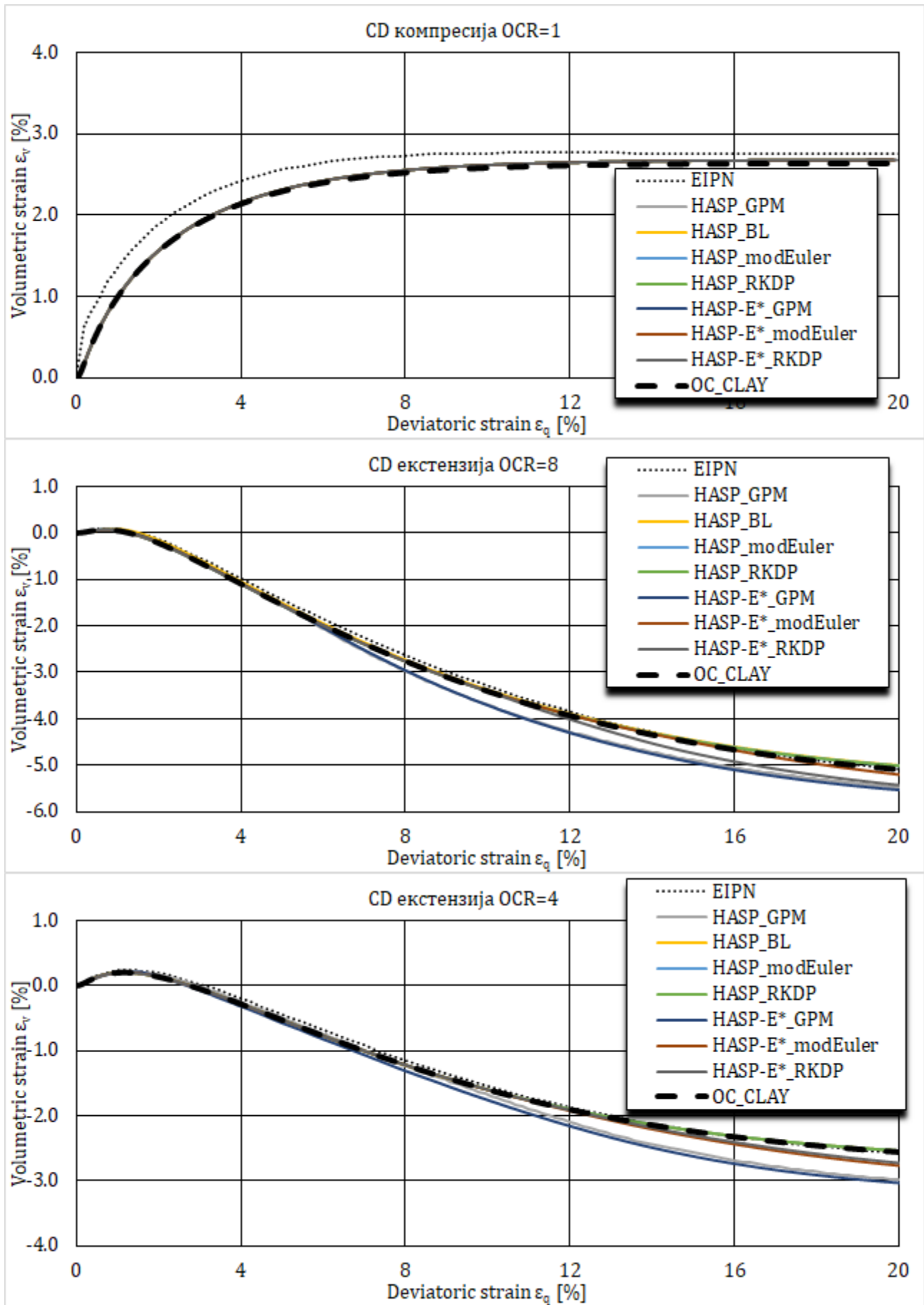


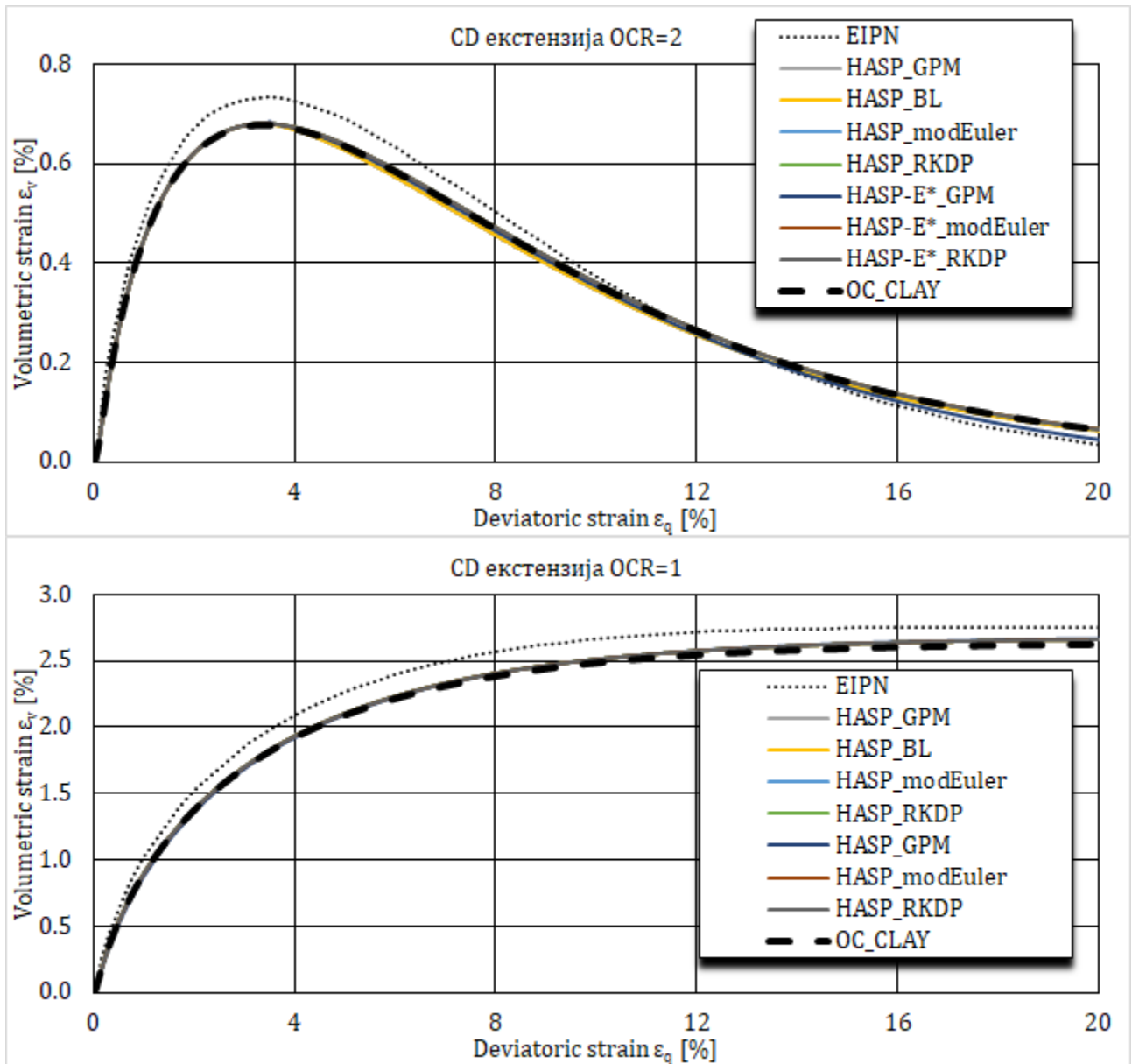


Слика 4-9 Напонско-деформацијске криве за CD триаксијалне опите на Fujinotori глини

Резултати нумеричких симулација опита дати су на **Слика 4-9** и **Слика 4-10**. Вредности вршног напонског односа η_f и вредност запреминске деформације при достигнутој девијаторској деформацији од 20 % приказани су у **Табела 4-7** и **Табела 4-8**. Резултати нумеричке симулације опита се одлично поклапају са резултатима EIPN нумеричке интеграције из (Јосковић, 2017). Код опита триаксијалне екстензије за степене преконсолидације од 8 и 4, HASP модели имплементирани са имплицитним методама имају потешкоћа када напонска путања прелази први пут линију критичног стања. Тај проблем је већ описан у потпоглављима 4.2 и 4.3 у алгоритмима за имплицитну интеграцију. Алгоритми имплицитних метода нумеричке интеграције могу да се изборе са тим, али се јавља грешка у предвиђању напона. У зависности од величине инкремента, зависи и величина грешке. Без обзира на то, све методе нумеричке интеграције предвиђају исту вредност вршног напонског односа и разлике у предвиђањима се јављају тек при већим девијаторским деформацијама.







Слика 4-10 Запреминске деформације за CD триаксијалне опите на Фуџијотори глини

Табела 4-7 Вршни напонски однос $\eta_f = q/p'$ за CD триаксијални опит на Фуџијотори глини

Метода нумеричке интеграције	Компресија				Екстензија			
	OCR [-]							
	8	4	2	1	8	4	2	1
EIPN	2.229	1.791	1.470	1.363	1.526	1.228	1.008	0.937
HASP_GPM	2.186	1.765	1.449	1.359	1.518	1.222	1.001	0.938
HASP_BL	2.185	1.764	1.449	1.359	1.519	1.219	1.001	0.938
HASP_modEuler	2.183	1.763	1.448	1.359	1.509	1.219	1.001	0.938
HASP_RKDP	2.183	1.763	1.448	1.359	1.509	1.219	1.001	0.938
HASP-E*_GPM	2.186	1.764	1.449	1.359	1.514	1.227	1.001	0.941
HASP-E*_modEuler	2.183	1.763	1.448	1.359	1.508	1.219	1.001	0.938
HASP-E*_RKDP	2.183	1.763	1.448	1.359	1.509	1.219	1.001	0.938

OC_CLAY	2.183	1.763	1.448	1.359	1.500	1.218	1.001	0.938
---------	-------	-------	-------	-------	-------	-------	-------	-------

Табела 4-8 Запреминска деформација у % при девијаторској деформацији од 20 % за CD триаксијални опит на Фујинотори глини

Метода нумеричке интеграције	Компресија				Екстензија			
	OCR [-]							
	8	4	2	1	8	4	2	1
EIPN	-5.46	-2.75	-0.016	2.75	-5.09	-2.57	0.035	2.76
HASP_GPM	-5.32	-2.71	0.012	2.68	-5.47	-2.98	0.062	2.66
HASP_BL	-5.32	-2.71	0.012	2.68	-5.00	-2.54	0.062	2.66
HASP_modEuler	-5.32	-2.71	0.012	2.68	-5.01	-2.54	0.066	2.66
HASP_RKDP	-5.32	-2.71	0.012	2.68	-5.01	-2.54	0.066	2.66
HASP-E*_GPM	-5.32	-2.71	0.012	2.68	-5.53	-3.03	0.045	2.67
HASP-E*_modEuler	-5.32	-2.71	0.012	2.68	-5.20	-2.76	0.066	2.66
HASP-E*_RKDP	-5.32	-2.71	0.013	2.68	-5.43	-2.72	0.065	2.66
OC_CLAY	-5.44	-2.71	0.012	2.64	-5.09	-2.56	0.065	2.63

4.4.4 CD циклични триаксијални опит на Фујинотори глини

Симулиран је CD циклични триаксијални опит на Фујинотори глини. Иницијални услови за нумеричку симулацију су дати у **Табела 4-9**. Параметри конститутивних модела су дати у **Табела 4-10** и преузети су из (Nakai & Hinokio, 2004) и (Moghadam, et al., 2022). Иницијални коефицијент порозности је одређен из израза (3-48) на основу познате вредности параметра Γ , степена преконсолидације и иницијалног напонског стања. Радијални напон је био константан током целог опита.

Сprovedено је укупно осам нумеричких симулација: три са HASP-E моделом са неактивном “Brick” компонентном, три са HASP-E моделом са активном “Brick” компонентном, једна са OC_CLAY са неактивним SSO моделом и једна са OC_CLAY са активним SSO моделом. Нумеричке симулације које су спроведене са HASP-E се међусобно разликују по методама нумеричке интеграције које су коришћене.

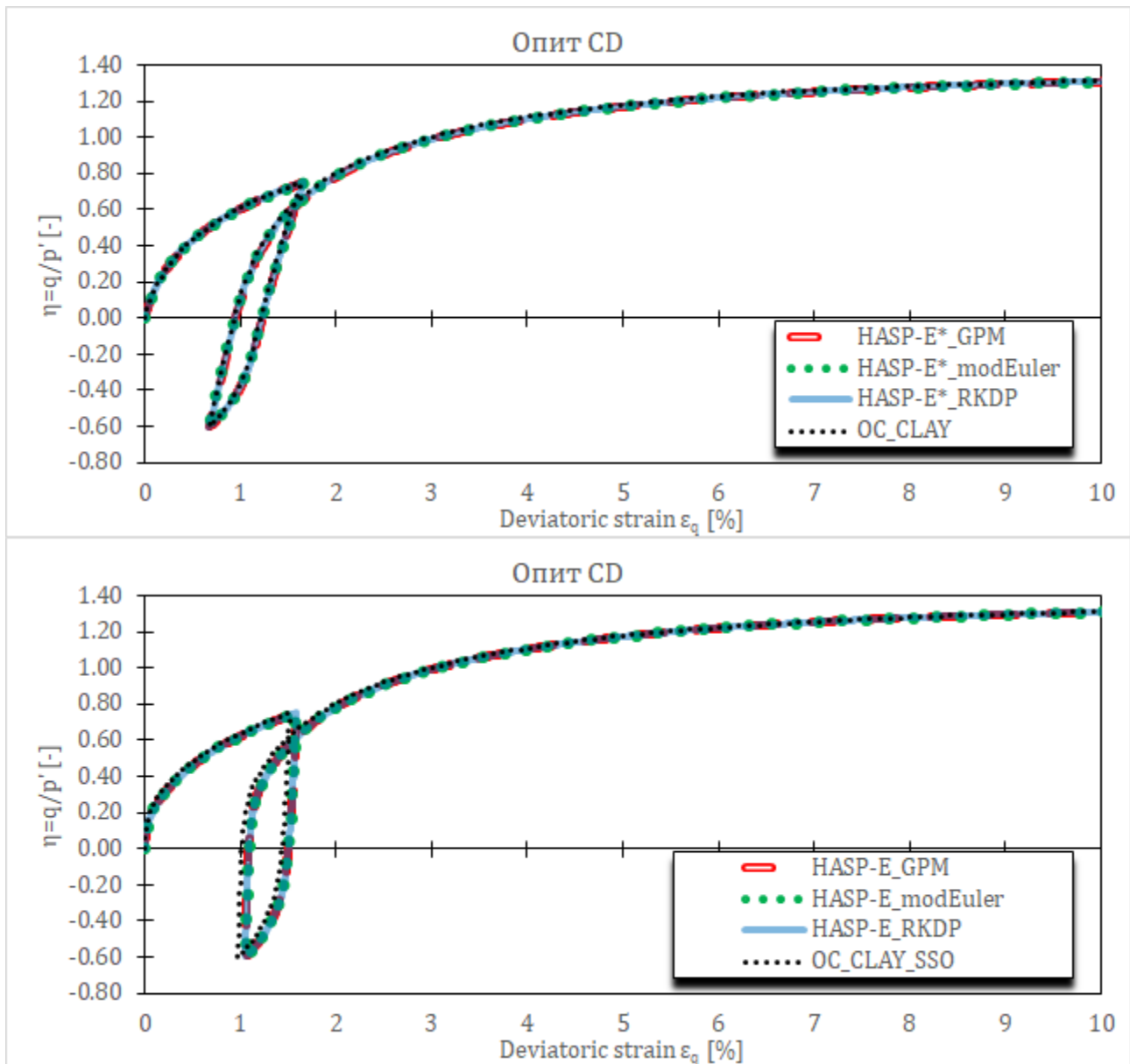
Табела 4-9 Иницијални услови за симулацију CD цикличног триаксијалног опита на Фујинотори глини

Опит	OCR[-]	p'_i [kPa]	e_o [-]	Γ [-]
CD	1	196	0.81	2.23

Табела 4-10 Параметри HASP-E и OC_CLAY модела за симулацију CD цикличног триаксијалног опита на Фујинотори глини

Модел		Параметри						
HASP-E	λ	κ	M_c	M_e	μ	G_0^{ref}	$\gamma_{0,7}$	
	[-]	[-]	[-]	[-]	[-]	[kPa]	[-]	
	0.089	0.02	1.36	0.94	0.2	67875	0.00025	
OC_CLAY	Опит	E_{ur}^{ref}	μ	E_{oed}^{ref}	G_0^{ref}	$\gamma_{0,7}$	ϕ_{cs}	POP
		[kPa]	[-]	[kPa]	[kPa]	[-]	[°]	[kPa]
	CD	16290	0.2	2033.71	67875	0.00025	33.67	0

Параметри који описују S-криву нису били на располагању, па су вредности G_0^{ref} и $\gamma_{0,7}$ усвојени. За усвојену вредност G_0^{ref} , однос модула смицања при веома малим деформацијама и при великим деформацијама је десет пута. За вредност параметра $\gamma_{0,7}$ усвојена је вредност од $\gamma_{0,7} = 2.5 \cdot 10^{-4}$.



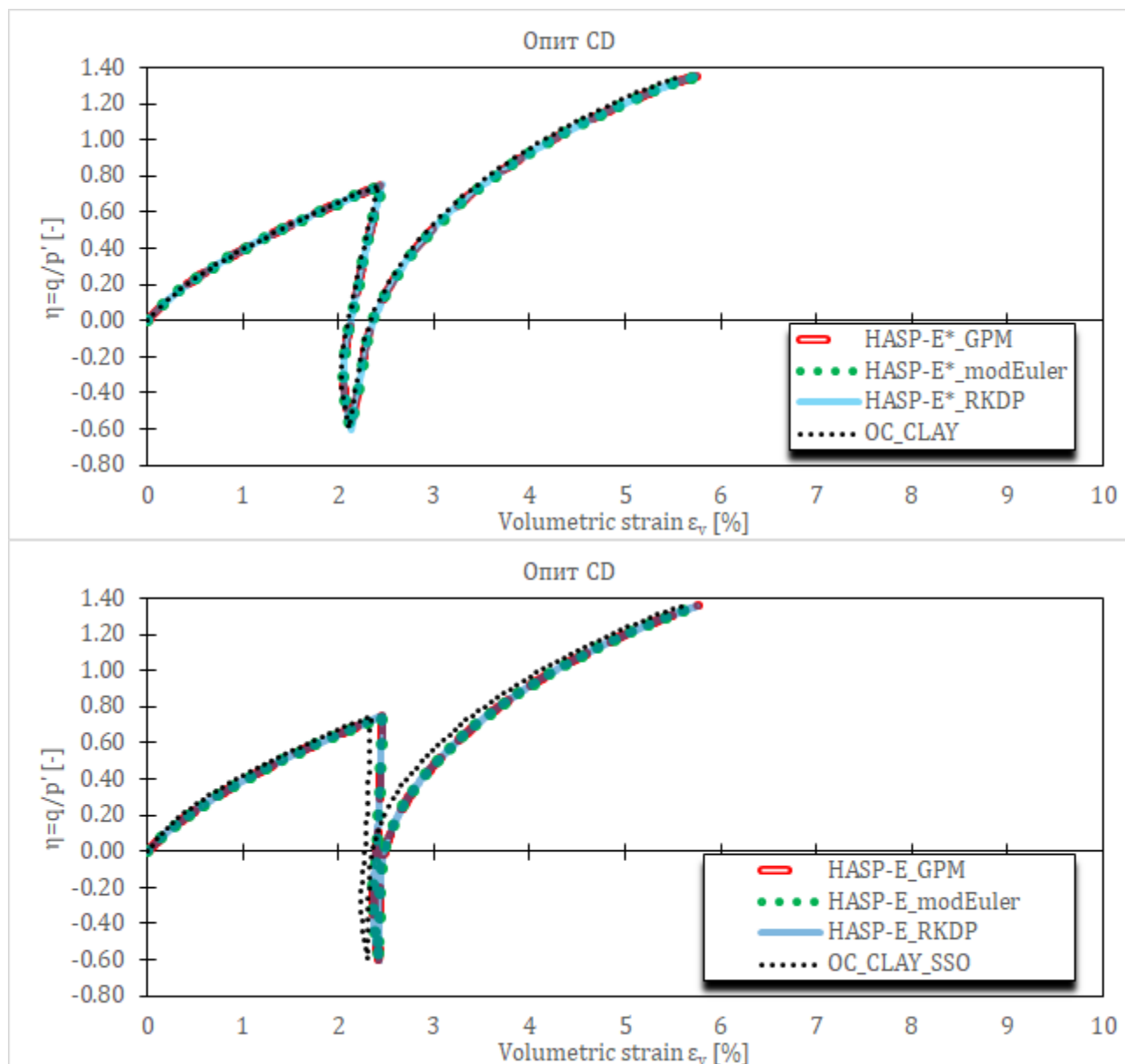
Слика 4-11 Напонско-деформацијске криве за CD циклични триаксијални опит на Fujiyotorigi глини

У опиту, напонски однос је прво повећан до 0.75 (компресија), па смањен до -0.6 (екстензија) и онда повећаван до лома у компресији.

Резултати нумеричких симулација опита дати су на **Слика 4-11** и **Слика 4-12**.

Из приказаних резултата може се уочити да HASP-E за све методе нумеричке интеграције и OC_CLAY предвиђају исто понашање тла. Мале разлике у предвиђеном понашању глине између HASP-E модела са активном “Brick” компонентном и OC_CLAY

са активним SSO моделом могу бити последица коришћења степенасте S-криве у HASP-E уместо континуалне криве у OC_CLAY моделу.



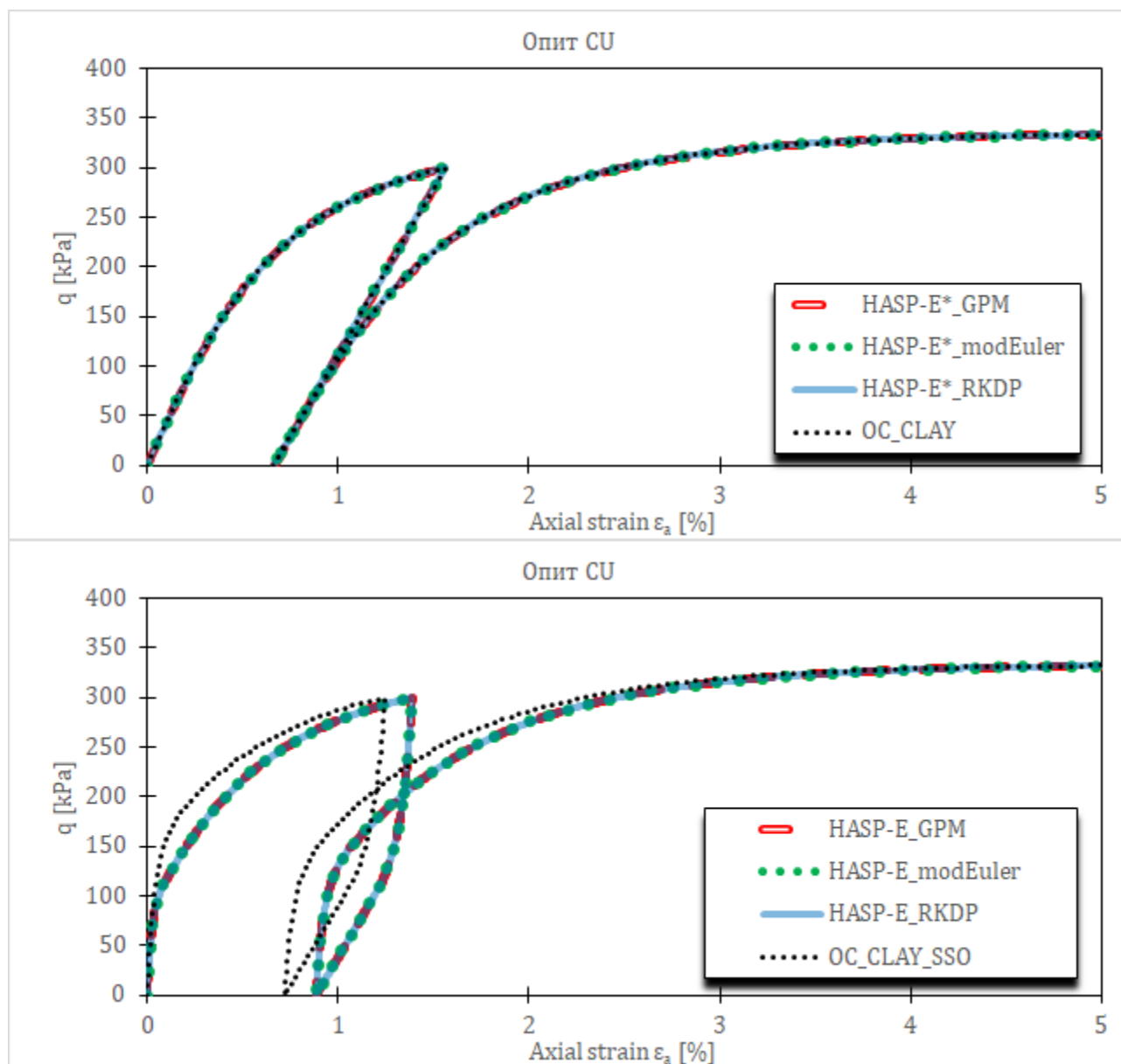
Слика 4-12 Запреминске деформације за CD циклични триаксијални опит на Фуџимотори глини

4.4.5 CU циклични триаксијални опити на Newfield глини

Симулиран је CU циклични триаксијални опит на Newfield глини. Иницијални услови за нумеричку симулацију су дати у Табела 4-11. Параметри конститутивних модела су дати у Табела 4-12 и преузети су из (Moghadam, et al., 2022). Иницијална вредност коефицијента порозности је одређена из израза (3-48) на основу познате вредности параметра Γ , степена преконсолидације и иницијалног напонског стања. У опиту је девијаторски напон повећан до $q_{cyc} = 300 \text{ kPa}$ (компресија), па смањен до нуле (један циклус), а након тога повећаван је до лома. Коришћени су исти конститутивни модели и методе нумеричке интеграције као у потпоглављу 4.4.4.

Табела 4-11 Иницијални услови за симулацију CU цикличног триаксијалног опита на Newfield глини

Опит	OCR[-]	p'_i [kPa]	e_o [-]	Γ [-]
CU	1	393	0.71	2.1



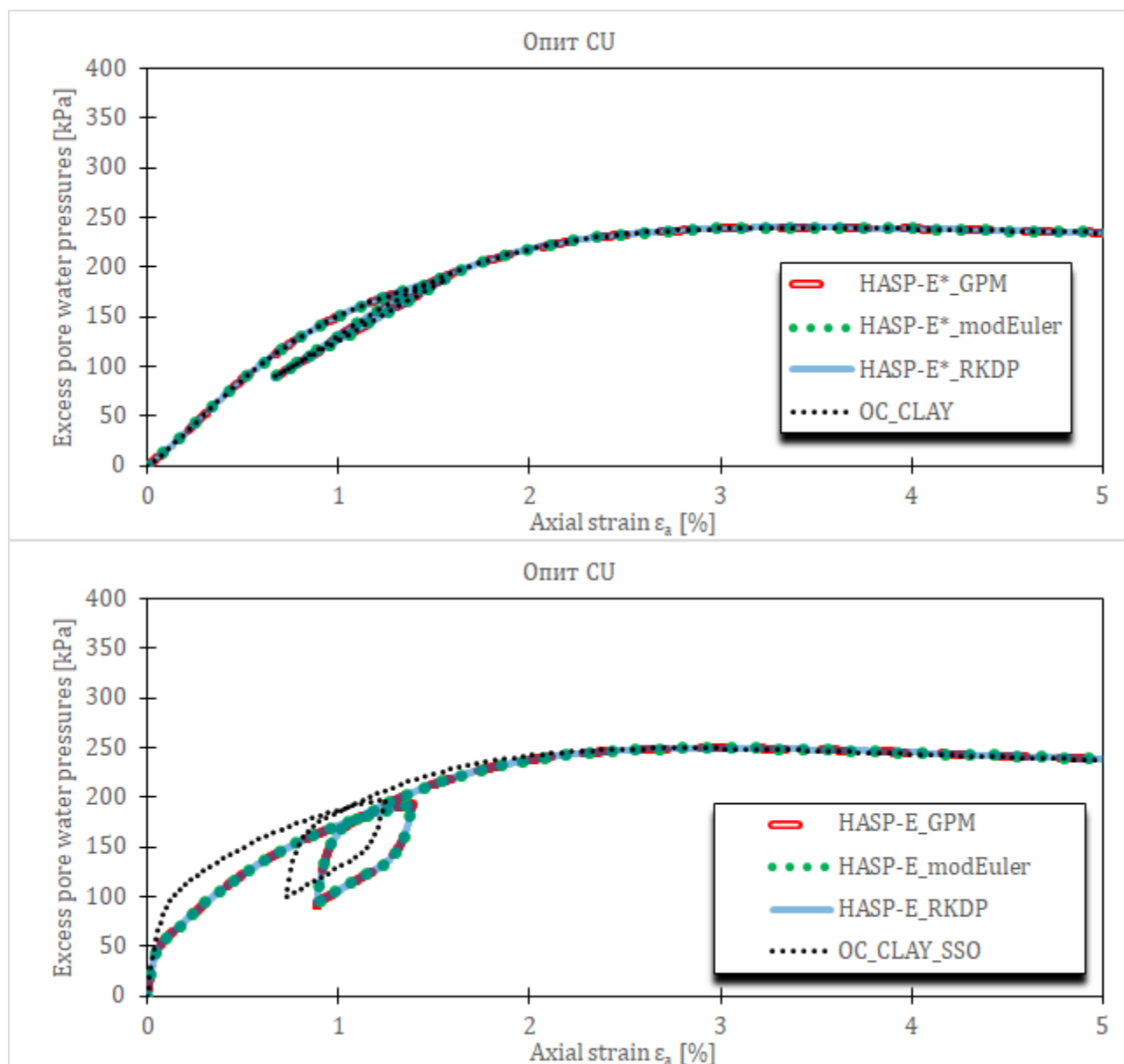
Слика 4-13 Напонско-деформацијске криве за CU циклични триаксијални опит на Newfield глини

Пошто параметри који описују S-криву нису били на располагању, усвојене су вредности G_0^{ref} и $\gamma_{0,7}$. За усвојену вредност G_0^{ref} однос модула смицања при веома малим деформацијама и при великим деформацијама је десет пута. За вредност параметра $\gamma_{0,7}$ усвојена је вредност од $\gamma_{0,7} = 2.5 \cdot 10^{-4}$.

Резултати нумеричких симулација опита су дати на **Слика 4-13** и **Слика 4-14**.

Табела 4-12 Параметри *HASP-E* и *OC_CLAY* модела за симулацију *CU* цикличног триаксијалног опита на *Newfield* глини

Модел	Параметри							
<i>HASP-E</i>	λ	κ	M_c	M_e	μ	G_0^{ref}	$\gamma_{0,7}$	
	[-]	[-]	[-]	[-]	[-]	[kPa]	[-]	
	0.07	0.035	1.2	0.8	0.2	36643	0.00025	
<i>OC_CLAY</i>	Опит	E_{ur}^{ref}	μ	E_{oed}^{ref}	G_0^{ref}	$\gamma_{0,7}$	ϕ_{cs}	POP
		[kPa]	[-]	[kPa]	[kPa]	[-]	[°]	[kPa]
	CU	8794.29	0.2	2442.86	36643	0.00025	30	0



Слика 4-14 Генерисани порни притисци за *CU* циклични триаксијални опит на *Newfield* глини

Из приказаних резултата може се уочити да HASP-E модел без активне “Brick” компоненте и OC_CLAY модел без активног SSO модела дају идентично предвиђање понашања глине. Као и у потпоглављу 4.4.4, разлике у предвиђању понашања глине између HASP-E са активном “Brick” компонентом и OC_CLAY модела са активним SSO моделом се могу објаснити у разликама између S-кривих које користе ова два модела.

4.5 Анализа ефикасности, тачности и робустности алгоритама за интеграцију конститутивних релација

Алгоритми за нумеричку интеграцију конститутивних релација из потпоглавља 4.2 и 4.3 су међусобно упоређени према критеријумима ефикасности, тачности и робустности кроз анализу граничног проблема. Један алгоритам је ефикаснији од другог алгоритма уколико захтева мањи утрошак рачунарског времена потребног за рачунање решења са захтеваним нивоом тачности. Нумеричко решење се сматра тачним уколико се мало разликује од тачног решења. До тачног решења се најчешће долази користећи нумеричке методе и исте алгоритме, али се оптерећење наноси у што је могуће мањим инкрементима како би се добило нумеричко решење велике тачности. Робустност алгоритма се огледа у томе да варијације у односу на почетне услове оптерећивања и величину инкремента оптерећења не утичу битно на тачност нумеричког решења. Перформансе алгоритама за нумеричку интеграцију конститутивних релација према претходним критеријумима зависе од три фактора (Carow & Rackwitz, 2021): почетних услова оптерећивања, величине инкремента оптерећења и путање напона.

Извршена је упоредна анализа три алгоритма за нумеричку интеграцију конститутивних релација HASP модела и то: метода водећег параметра, алгоритам (Vorja & Lee, 1990) и модификовани Euler-ов поступак са аутоматском субинкрементацијом. За интеграцију конститутивних релација HASP-E модела анализирани су метода водећег параметра и модификовани Euler-ов поступак са аутоматском субинкрементацијом. За оба конститутивна модела, RKDP поступак са аутоматском субинкрементацијом са малом толеранцијом на локалну грешку је коришћен за одређивање „тачног“ решења. Верификација поступка имплементације свих набројаних алгоритама за оба модела је извршена у потпоглављу 4.4.

За гранични проблем је изабран пример одређивања носивости крутог тракастог темеља са храпавом контактном површином који се налази на површини терена у недренираним условима на слоју водом засићене глине.

У анализи су варирани следећи параметри: метода нумеричке интеграције конститутивних релација, иницијални услови (иницијални степен преконсолидације слоја глине) и величина инкремента доношења оптерећења.

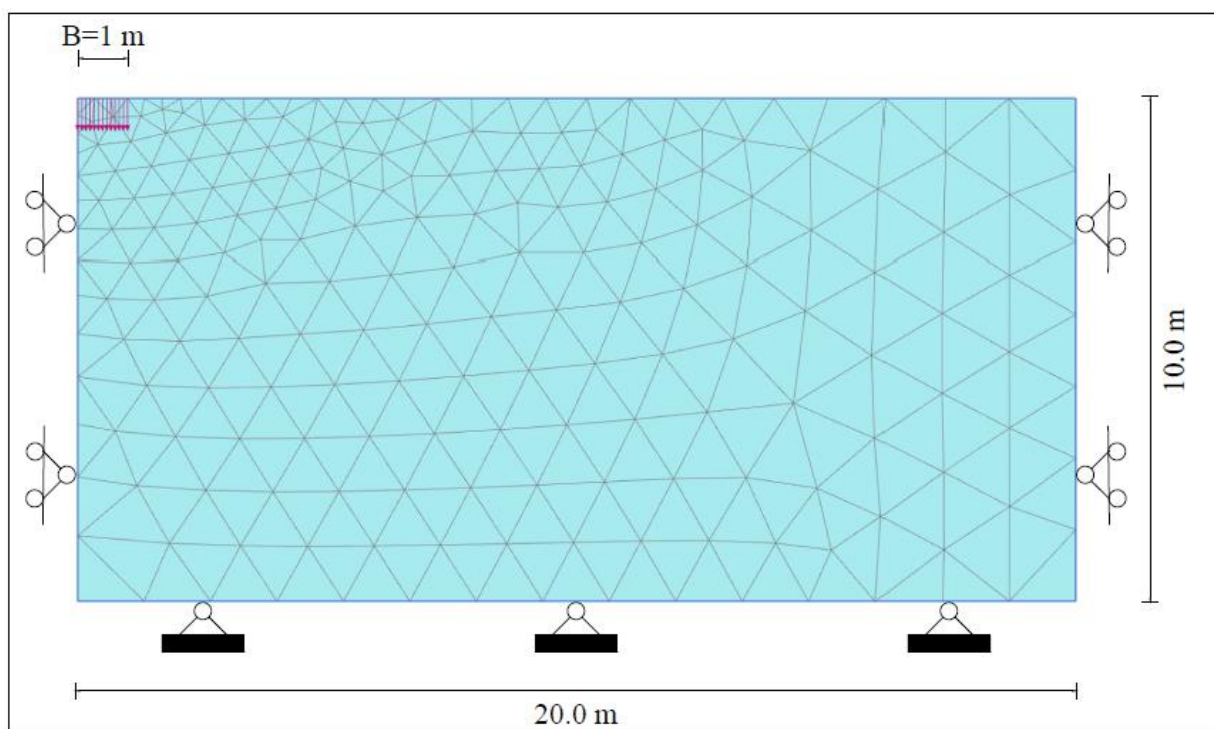
4.5.1 Нумерички модел

Нумерички модел је направљен у PLAXIS-у 2D и коришћено је раванско стање деформација. Димензије модела, мрежа коначних елемената и гранични услови су приказани на **Слика 4-15**. Коришћени су троугаони коначни елементи са 15 чворова. Због симетрије, разматрана је само половина модела. Усвојено је да се оптерећење на

тракасти темељ наноси веома брзо, тако да нема времена за дисипацију додатних порних притисака и у потпуности је занемарено струјање воде у порама тла. Због тога је коришћена недренирана еласто-пластична анализа типа А. На левој и десној ивици модела су спречена хоризонтална померања, на доњој ивици су спречена сва померања, а на горњој ивици су сва померања слободна. Нема потребе да се дефинишу хидраулички гранични услови јер је занемарено струјање воде у порама тла. Ширина модела је 20 m а дебљина слоја глине је 10 m. Ширина тракастог темеља у анализираној половини модела је 1.0 m. Тракасти темељ се налази на површини терена. Оптерећење на тракасти темељ је нането у облику задатих вертикалних померања (prescribed displacements) у износу од $u_{max} = 0.4$ m. Крут и хрпава темељ је симулиран тако што је свим тачкама тла испод темеља задато исто вертикално померање, док су релативна хоризонтална померања истих тачака спречена. На тај начин, није потребно моделирати сам темељ посебним коначним елементима.

Ниво подземне воде је задат на дубини од 1.0 m испод површине терена. Сматра се да је слој глине изнад нивоа подземне воде водом засићен и да су порни притисци негативни, односно да се јављају напони сукције. Као последица тога, на површини терена ефективни напони ће бити већи од нуле и глина ће имати смичућу чврстоћу. Уколико се не би узели напони сукције у обзир, на површини терена ефективни напони би били једнаки нули и глина не би имала никакву смичућу чврстоћу, односно не би могла да прими додатно оптерећење од тракастог темеља.

Разматране су две фазе. Прва фаза је иницијална фаза у којој се генеришу почетни напони и порни притисци у тлу. У другој фази су нанета задата вертикална померања на тракасти темељ. PLAXIS 2D наноси у чворовима мреже испод тла вертикалне силе све док се не достигне задата величина вертикалних померања. На крају фазе, PLAXIS 2D као резултат даје резултанту нанетих вертикалних сила у функцији величине нанетих померања.



Слика 4-15 Мрежа коначних елемената и гранични услови за модел тракастог темеља

За слој глине коришћени су параметри HASP и HASP-E модела за Newfield глину који су дати у **Табела 4-12**. У HASP-E моделу, "Brick" компонента није била активна, па параметри G_0^{ref} и $\gamma_{0,7}$ нису коришћени. За HASP модел коришћена је вредност параметра M од 1.2. Пошто тој вредности одговара угао унутрашњег трења од $\phi' = 30^\circ$, коефицијент притиска тла у стању мировања за нормално консолидовано тло је срачунат према изразу:

$$K_0^{NC} = 1 - \sin \phi' \quad 4-294$$

Коефицијент притиска тла у стању мировања за преконсолидовано тло се може одредити према изразу:

$$K_0^{OC} = K_0^{NC} \cdot \sqrt{OCR} \quad 4-295$$

За запреминску тежину тла изнад и испод нивоа подземне воде је усвојена вредност од $\gamma = 20 \text{ kN/m}^3$. За запреминску тежину воде коришћена је вредност од 10 kN/m^3 . За иницијални коефицијент порозности за цео слој глине је усвојена вредност од $e_0 = 0.85$. За ту вредност, просечна вредност параметра Γ за цео слој глине срачуната према изразу (3-48), на основу задатих степена преконсолидације и иницијалног стања напона, има вредност од 2.1. Преконсолидација слоја глине је задата преко параметра POP за цео слој глине. Коришћена је вредност од $POP = 100 \text{ kPa}$ (референтни модел), што би одговарало слоју дебљине тла од 5 m који се некада налазио изнад слоја глине.

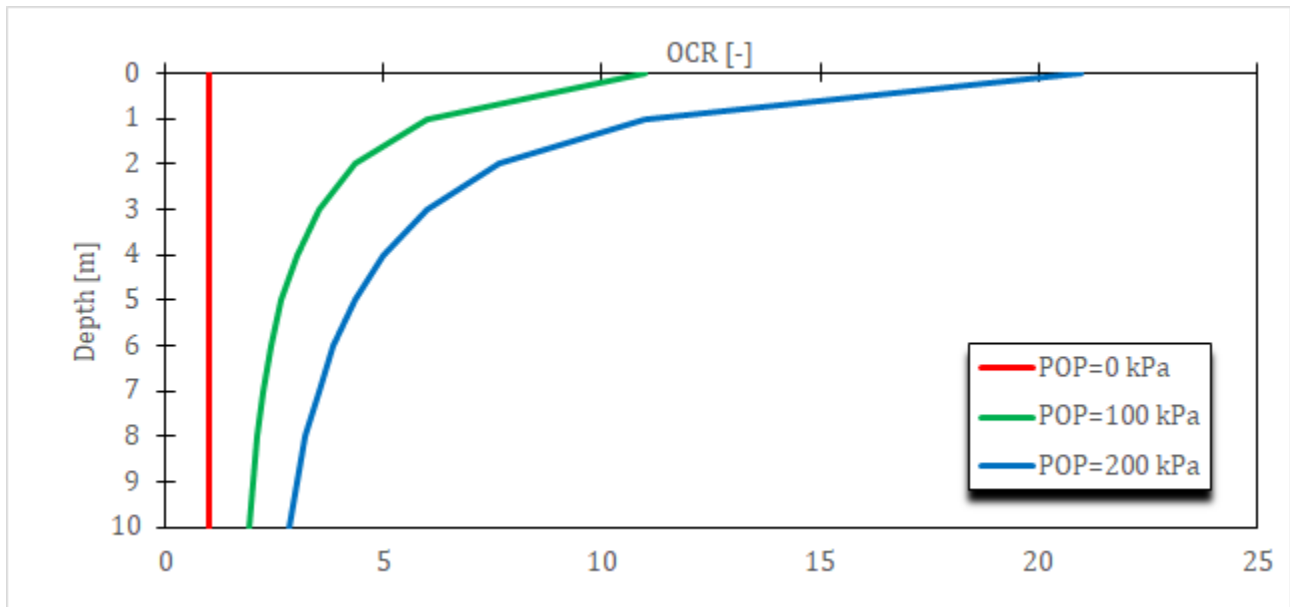
Како би се анализирао утицај иницијалних услова на робустност алгоритама, разматрано је више модела са различитим вредностима параметра POP . Поред вредности од $POP = 100 \text{ kPa}$ (референтна вредност), коришћене су још вредности од $POP = 0 \text{ kPa}$ (нормално консолидован слој глине) и $POP = 200 \text{ kPa}$ (јако преконсолидован слој глине). Промена степена преконсолидације слоја глине по дубини може се одредити из израза:

$$OCR = \frac{\sigma'_{vi} + POP}{\sigma'_{vi}} \quad 4-296$$

У претходном изразу σ'_{vi} је иницијални вертикални напон. Промена степена преконсолидације са дубином за све вредности параметра POP је дата на **Слика 4-16**. Коефицијент притиска тла у стању мировања за преконсолидовано тло је срачунат према изразу (4-296). С обзиром да се вредност OCR мења по дубини, за преконсолидовано тло се онда и K_0^{OC} мења са дубином. Међутим, да би се поједноставнио нумерички модел (нумерички моделиран само један слој глине), за цео слој глине је усвојена просечна вредност срачуната према изразу (4-296). Иницијални услови за све три разматране вредности POP дате су у **Табела 4-13**.

Табела 4-13 Иницијални услови за одређивање носивости тракастог темеља на Newfield глини

Модел	$POP[\text{kPa}]$	$K_0^{NC}[-]$	K_0^{OC}	$e_0[-]$	$\Gamma[-]$
M1 (референтни)	100		0.93		
M2	0	0.5	0.5	0.85	2.1
M3	200		1.2		



Слика 4-16 Промена степена преkonsolidације са порастом дубине у зависности од вредности параметра POP

За одређивање „тачног“ решења преко RKDP поступка са аутоматском субинкрементацијом коришћена је толеранција на локалну грешку од $STOL = 10^{-6}$, како симулације не би трајале неприхватљиво дуго. Мање вредности од 10^{-6} могу да се користе како би се додатно повећала тачност решења. Померања су нанета у 1000 инкремената. Даље смањивање вредности за $STOL$ од 10^{-6} и употреба већег броја инкремената од 1000 не би требало да утиче на опште закључке анализе ефикасности, тачности и робустности који су добијени са коришћеним вредностима.

Утицај толеранције за локалну грешку $STOL$ у модификованом Euler-овом поступку са аутоматском субинкрементацијом за оба конститутивна модела је анализиран на референтном моделу (модел M1), тако што су разматране следеће вредности за $STOL$: 10^{-4} , 10^{-3} , 10^{-2} , 10^{-1} и 1. За вредност $STOL = 1$ у експлицитним методама неће бити потребна субинкрементација.

Утицај величине инкремента на тачност и ефикасност поступка је анализиран тако што је вариран број инкремената n_{steps} приликом nanoшења вертикалних померања од $u_{max} = 0.4 m$. Разматран је следећи број инкремената: $n_{steps} = 40/80/200/400/1000$.

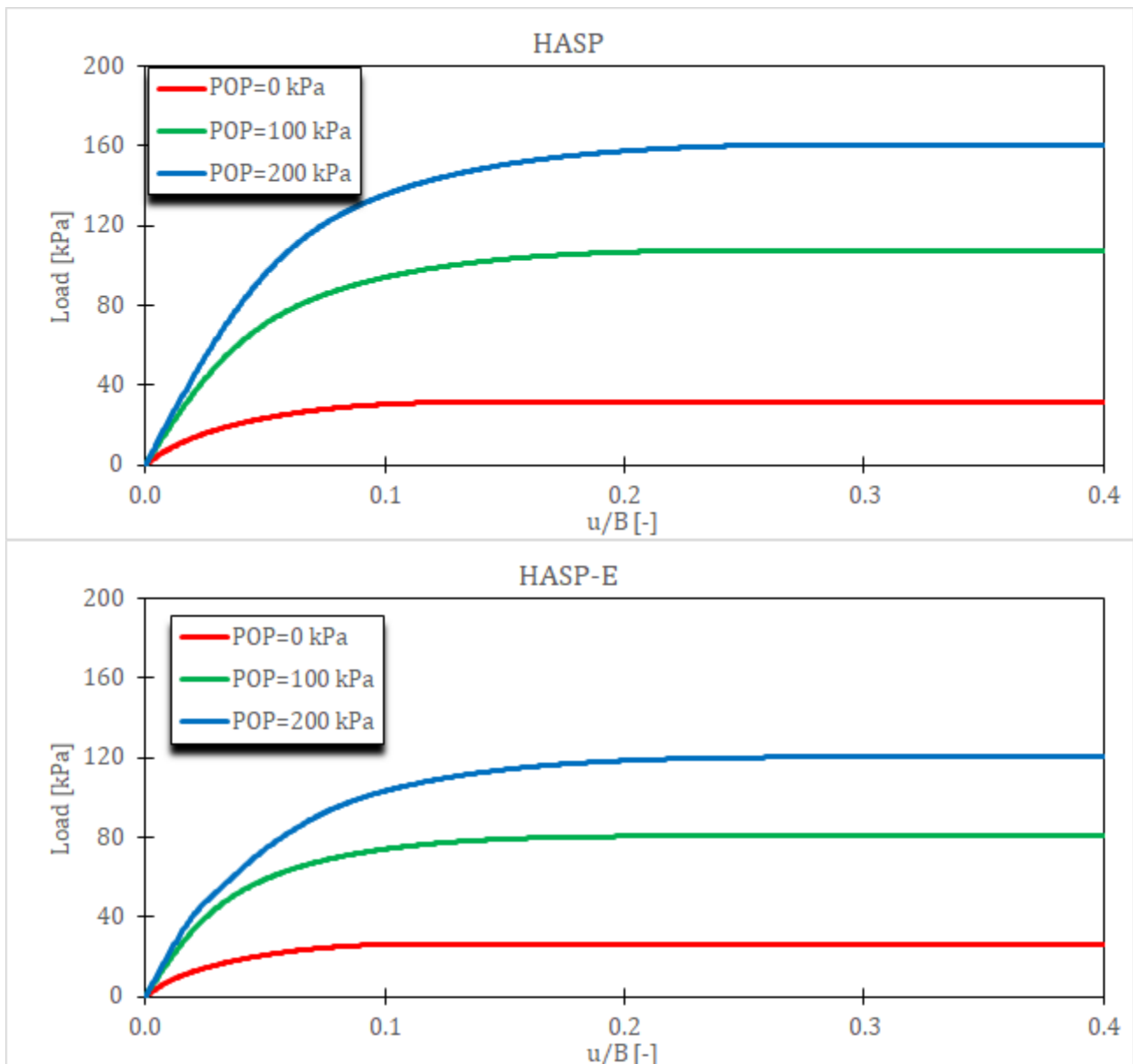
Све симулације су пуштане на рачунару са процесором Intel Core i7-12700H 4.7GHz са 32 GB RAM меморије. За сваку симулацију забележено је укупно процесорско време (CPU time) у секундама које је било потребно за спровођење симулације и ова величина представља меру утрошка рачунарског времена.

4.5.2 Реултати и дискусија

Тачност нумеричког решења може се квантификовати преко релативне грешке, E_Q , која се може срачунати према изразу (4-297). Са $Q_{exa}(u_k)$ је означена крива која одговара „тачном“ решењу и која описује зависност оптерећења Q које делује на тракасти темељ и слегања темеља u услед тог оптерећења. Са $Q(u_k)$ је означена иста крива која је добијена применом нумеричког поступка коришћењем n_{steps} инкремената.

$$E_Q = \sqrt{\sum_{k=1}^{n_{steps}} \frac{(Q_{exa}(u_k) - Q(u_k))^2}{(Q_{exa}(u_k))^2}}$$

Као што је претходно описано, „тачно“ решење је добијено применом RKDP поступка са аутоматском субинкрементацијом. Криве зависности оптерећења и слегања тракастог темеља за тачно решење (референтне криве) приказане су на **Слика 4-17**. Може се уочити да са порастом степена преконсолидације криве постају стрмије и носивост тракастог темеља расте. Такође се може уочити да HASP-E модел предвиђа мању носивост темеља него HASP модел.



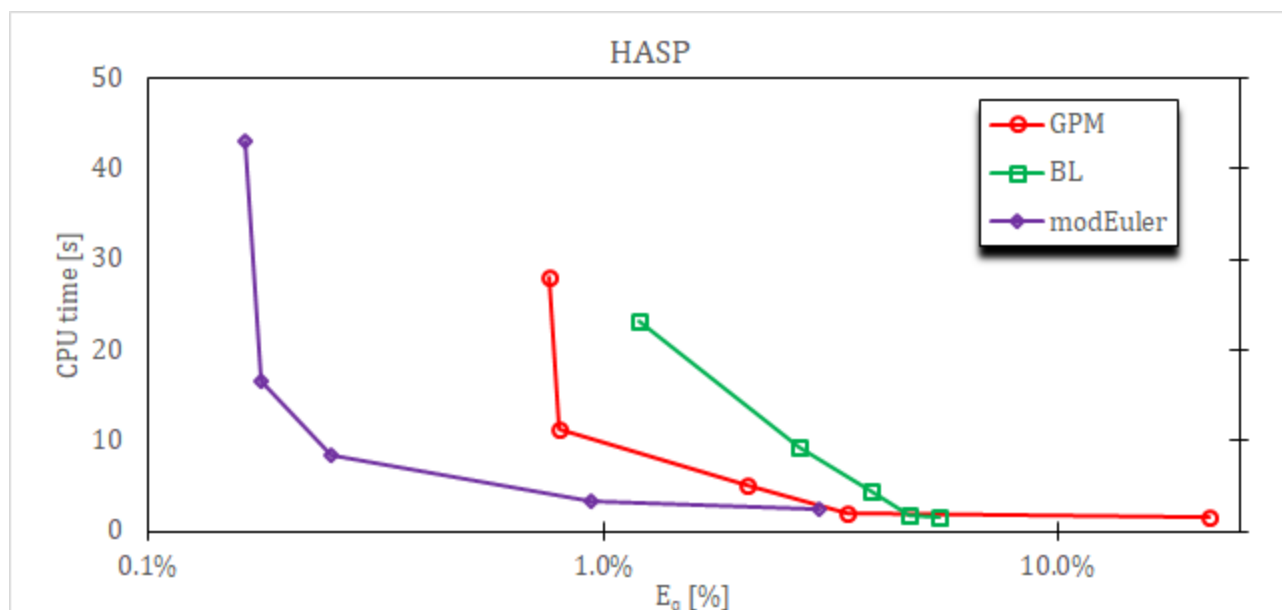
Слика 4-17 Референтне криве зависности оптерећења и слегања тракастог темеља у недренираним условима за HASP и HASP-E модел за различите иницијалне услове

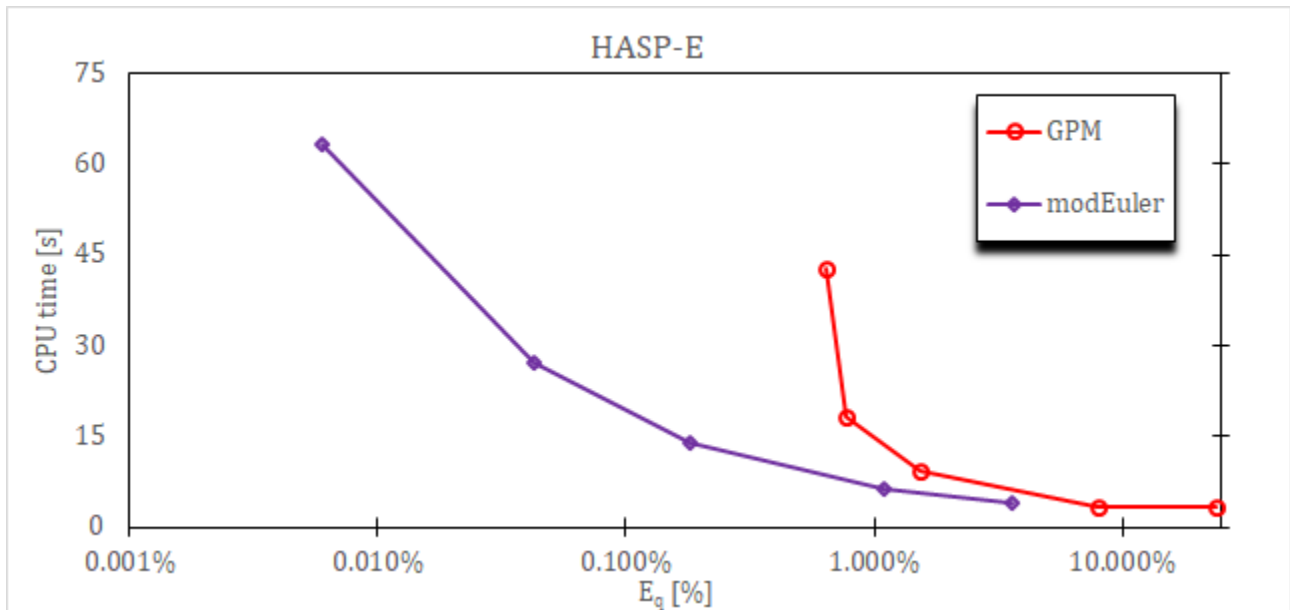
Разлика у носивостима тракастог темеља између HASP-E и HASP модела је последица тога што HASP-E узима у обзир утицај треће напонске инваријанте и користи адекватнији критеријум чврстоће тла него HASP. У **Табела 4-14** приказане су срачунате носивости тракастог темеља за “тачно” решење и потребан утрошак рачунарског времена. Може се уочити да са порастом степена преконсолидације расте и утрошак рачунарског времена, као и да је за исти степен преконсолидације потребан већи утрошак рачунарског времена код HASP-E него код HASP модела. Односно, да код конститутивних модела са неасоцијативним законом течења треба већи утрошак рачунарског времена него код модела са асоцијативним законом течења.

Табела 4-14 Носивост тракастог темеља у недренираним условима и потребан утрошак рачунарског времена за “тачно” решење (RKDP поступак са аутоматском субинкрементацијом, $STOL = 10^{-6}$ и $n_{steps} = 1000$)

Модел	POP = 0 kPa		POP = 100 kPa		POP = 200 kPa	
	CPU time [s]	p_{ult} [kPa]	CPU time [s]	p_{ult} [kPa]	CPU time [s]	p_{ult} [kPa]
HASP	92.156	31.55	271.844	107.62	266.813	160.66
HASP-E	123.375	26.33	348.203	80.70	499.391	120.37

На **Слика 4-18** дат је упоредни приказ релативне грешке E_Q и утрошка рачунарског времена за сваки алгоритам и оба конститутивна модела. Свака крива на **Слика 4-18** има пет тачака које одговарају коришћеном броју инкремената за наношење оптерећења (пет различитих бројева инкремената n_{steps} је коришћено). Тачке које су скроз десно одговарају броју инкремената од $n_{steps} = 40$, а тачке које су скроз лево одговарају броју инкремената од $n_{steps} = 1000$. Између су тачке за остали број инкремената. Што је крива за један алгоритам ближа доњем левом углу дијаграма, то је алгоритам ефикаснији и тачнији. На **Слика 4-18** су са GPM означени резултати за методу водећег параметра, са BL за алгоритам (Borja & Lee, 1990), а са modEuler модификовани Euler-ов поступак са аутоматском субинкрементацијом и $STOL = 10^{-4}$.

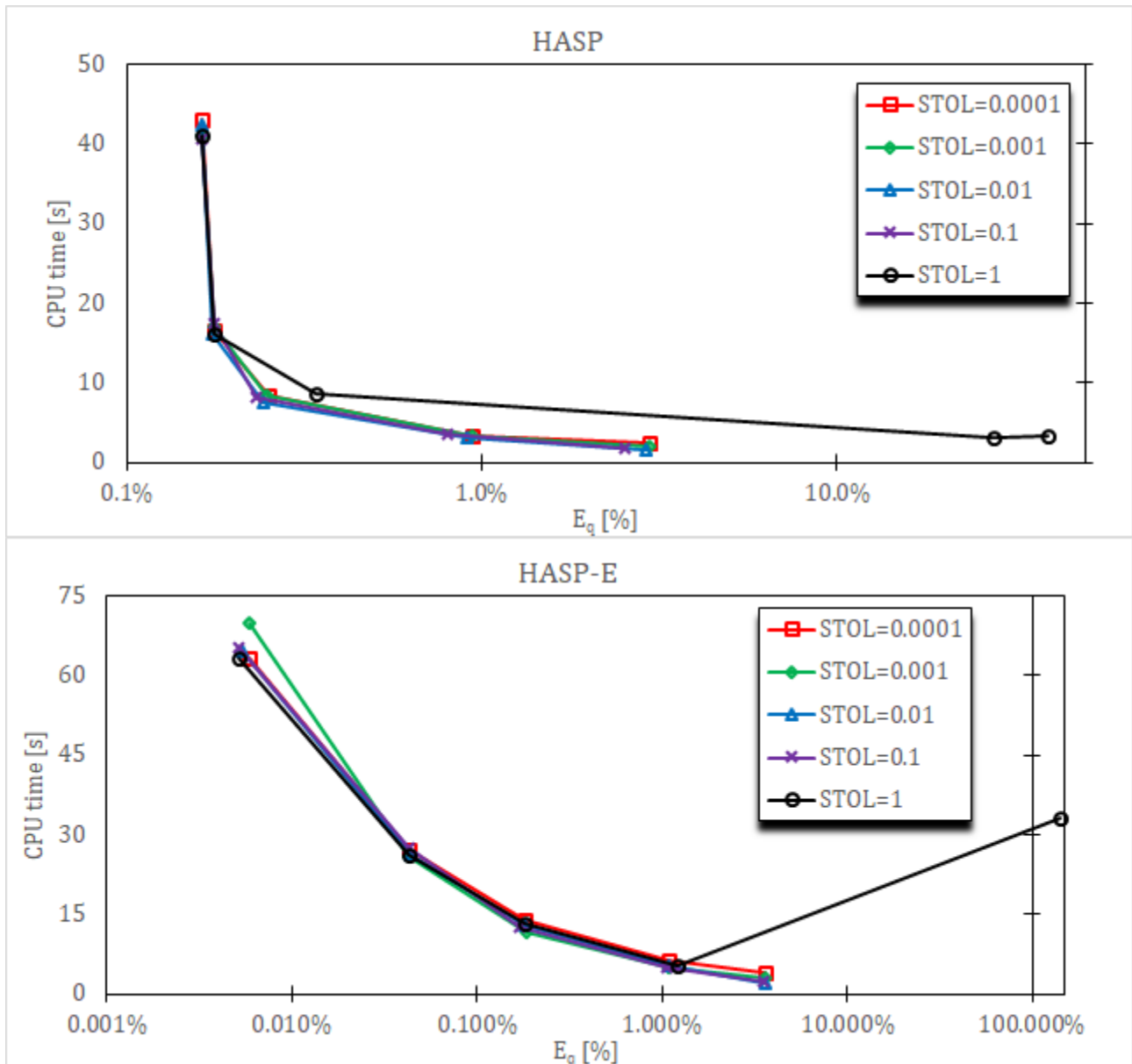




Слика 4-18 Тачност и ефикасност алгоритама за нумеричку интеграцију HASP и HASP-E модела за $POP = 100 \text{ kPa}$ и различит број инкремената

За HASP модел алгоритам (Vorja & Lee, 1990) захтева најмањи утрошак рачунарског времена, али је најмање тачан. Модификовани Euler-ов поступак са аутоматском субинкрементацијом је за оба модела најтачнији, али захтева већи утрошак рачунарског времена. Метода водећег параметра је за HASP модел тачнија од алгоритма (Vorja & Lee, 1990) уз скоро исти утрошак рачунарског времена. За оба модела, метода водећег параметра је мање тачна и са мање потребног утрошка рачунарског времена од експлицитне методе. За методу водећег параметра величина грешке је сличне вредности за оба модела. Са порастом броја коришћених инкремената релативна грешка опада уз пораст утрошка рачунарског времена за све моделе. За мали број инкремената, имплицитне методе имају скоро десет пута већу грешку него експлицитна метода са субинкрементацијом, због тога што експлицитне методе кроз субинкрементацију држе величину грешке у задатим границама. Код HASP модела за експлицитну и методу водећег параметра употреба већег броја инкремената од 1000 не би даље повећавала тачност решења.

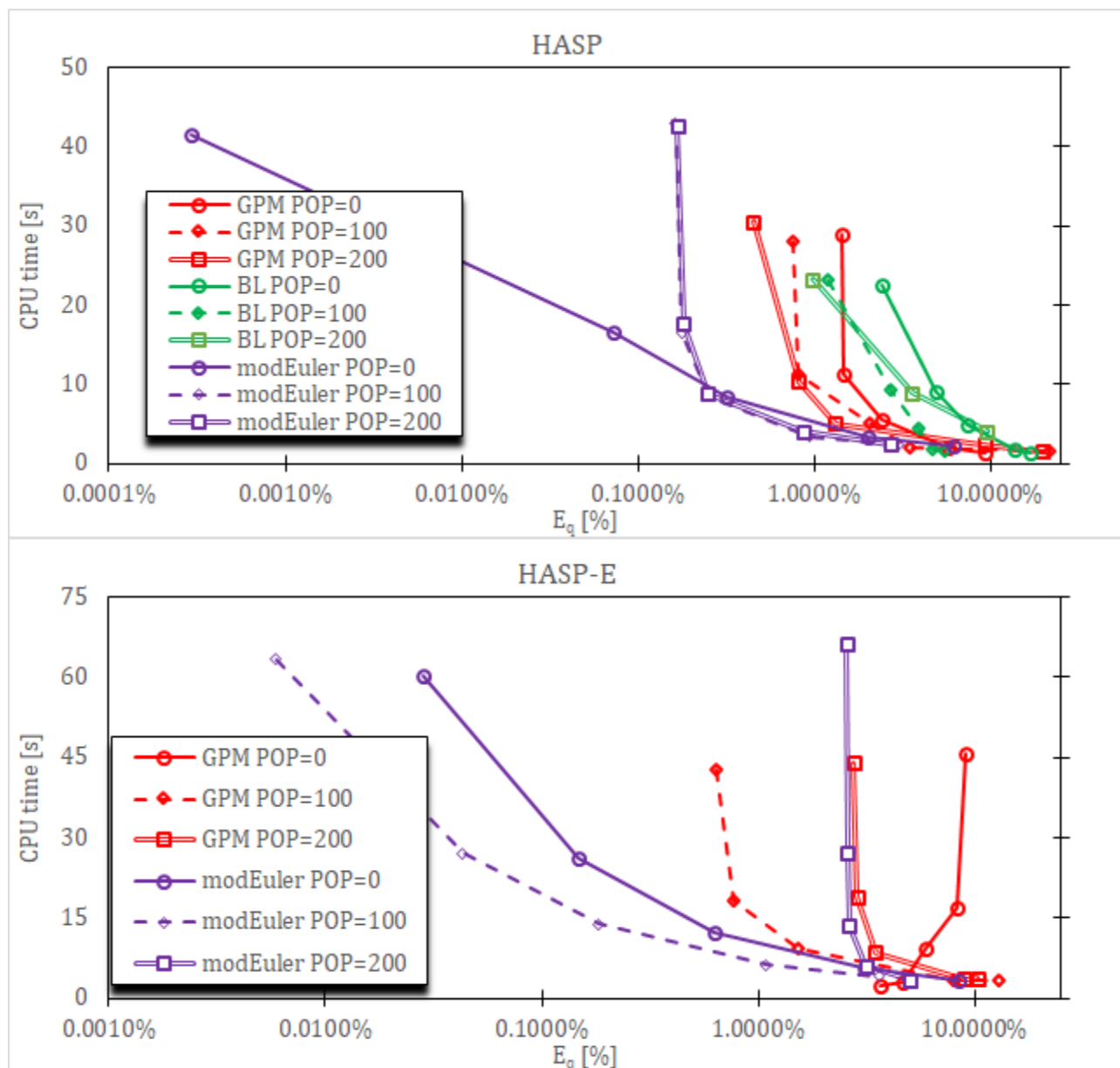
На **Слика 4-19** приказано је како величина толеранције на локалну грешку $STOL$ утиче на тачност и ефикасност модификованог Euler-овог поступка са аутоматском субинкрементацијом за оба конститутивна модела. За оба модела, за све вредности $STOL$ осим за $STOL = 1$, експлицитна метода даје решење сличне тачности уз исти утрошак рачунарског времена. Разлог за то би могао да буде да су инкременти у којима се наноси оптерећење већ довољно мали тако да није потребна субинкрементација и да су експлицитни алгоритми генерално тачнији од имплицитних алгоритама. За $STOL = 1$ (неће се вршити субинкрементација) и мали број инкремената, експлицитне методе дају решење које се доста разликује од тачног решења. Али ако се користи велики број инкремената, чак при вредности $STOL = 1$ експлицитне методе дају решење које је блиско тачном решењу. Експлицитне методе су ефикасније и тачније него имплицитне методе, јер и уз коришћење мањег броја инкремената, са адекватно одабраном вредности за $STOL$ дају решење задовољавајуће тачности уз прихватљив утрошак рачунарског времена.



Слика 4-19 Тачност и ефикасност модификованог Euler-овог поступка са аутоматском субинкрементацијом за $POP = 100 \text{ kPa}$ у зависности од величине $STOL$

На **Слика 4-20** приказано је како иницијални услови утичу на тачност и ефикасност алгоритама за оба модела. За модификовани Euler-ов поступак са аутоматском субинкрементацијом је коришћена вредност $STOL = 10^{-4}$. За алгоритам (Vorja & Lee, 1990) код HASP модела се за различите степене преконсолидације добијају сличне вредности релативне грешке и утрошка рачунарског времена. Код нумеричких симулација за јако преконсолидовано тло ($POP = 200 \text{ kPa}$) за мали број инкремената (40 и 80) код HASP модела са имплицитним алгоритмом (Vorja & Lee, 1990) није могла да се постигне конвергенција у глобалном инкрементално-итеративном поступку. Због тога, када се користи овај алгоритам за веће степене преконсолидације потребно је оптерећење нанети у што већем броју инкремената, односно инкременти морају да буду мањи него за друге методе нумеричке интеграције. Иницијални услови такође утичу на тачност и ефикасност методе водећег параметра и експлицитне методе. Утрошак рачунарског времена је сличан за различите степене преконсолидације, али степен

преконсолидације утиче на величину релативне грешке. Код експлицитне методе са порастом степена преконсолидације расте и релативна грешка, док код методе водећег параметра величина релативне грешке се смањује са порастом степена преконсолидације. Такође се може уочити да се релативне грешке за различите методе нумеричке интеграције за исти број инкремената највише разликују када је глина нормално консолидована, док се са порастом степена преконсолидације за исти број инкремената смањује разлика између релативних грешака различитих метода нумеричке интеграције.

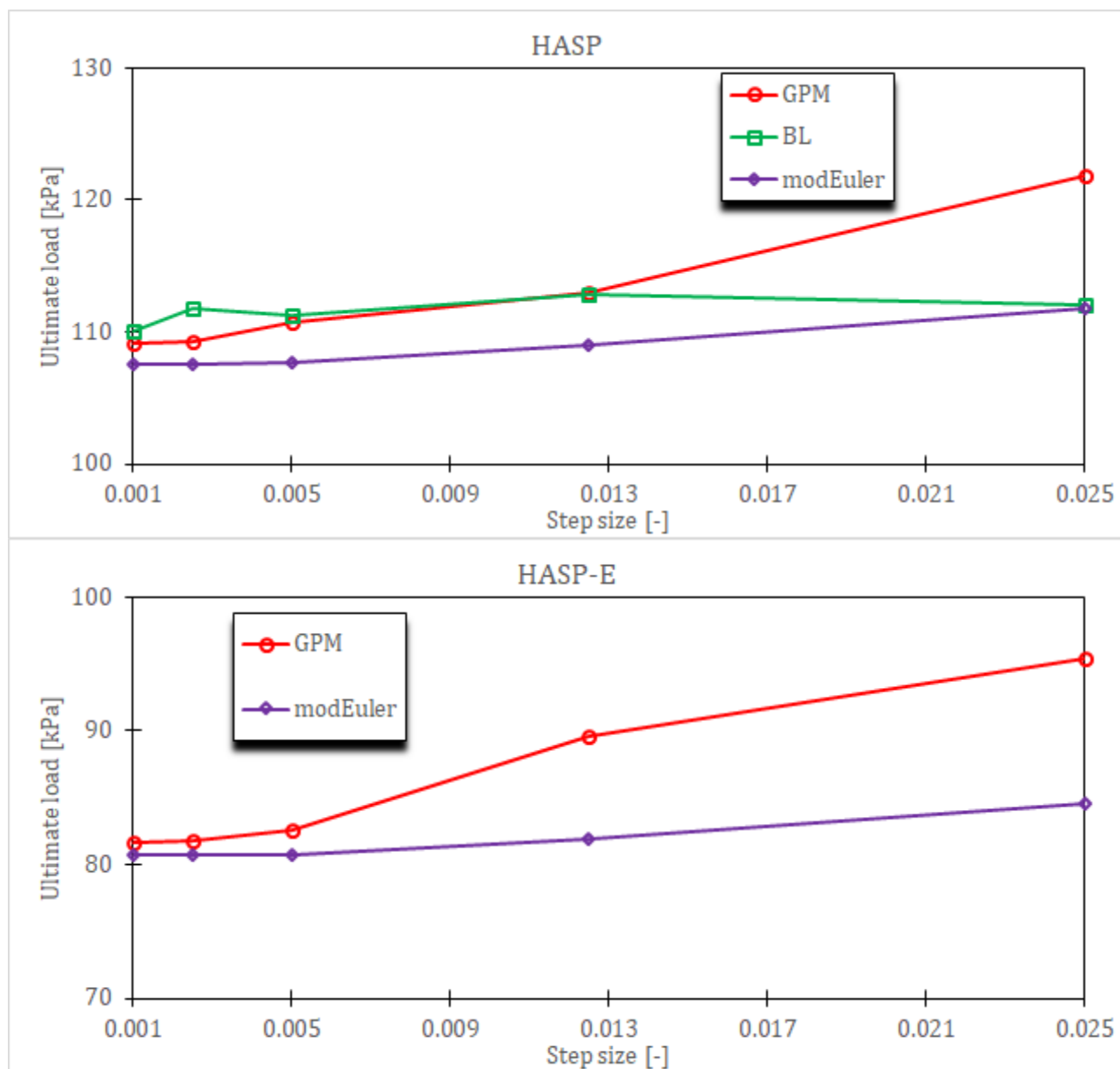


Слика 4-20 Тачност и ефикасност алгоритама за нумеричку интеграцију HASP и HASP-E модела у зависности од иницијалних услова за различит број инкремената

Код HASP-E модела иницијални услови имају великог утицаја на добијене резултате. За јако преконсолидовано тло ($POP = 200 \text{ kPa}$), експлицитна и имплицитна метода дају решења са сличном величином релативне грешке. Утрошак рачунараског времена за различите степене преконсолидације је сличан за исти број инкремената. Може се

уочити да се за обе методе нумеричке интеграције добијају мање релативне грешке за $POP = 100 \text{ kPa}$ и веће релативне грешке за $POP = 0 \text{ kPa}$, него за $POP = 200 \text{ kPa}$. Такође, за методу водећег параметра, за нормално консолидовану глине се уочава да релативна грешка расте са порастом броја инкремената, што је потпуно супротно понашање у односу на друге резултате. То може бити последица начина на који се неасоцијативни закон течења третира у имплицитним и експлицитним методама. Начин на који се врши нумеричка интеграција неасоцијативног закона течења у имплицитним и експлицитним методама има највећи утицај на резултате за нормално консолидовано тло.

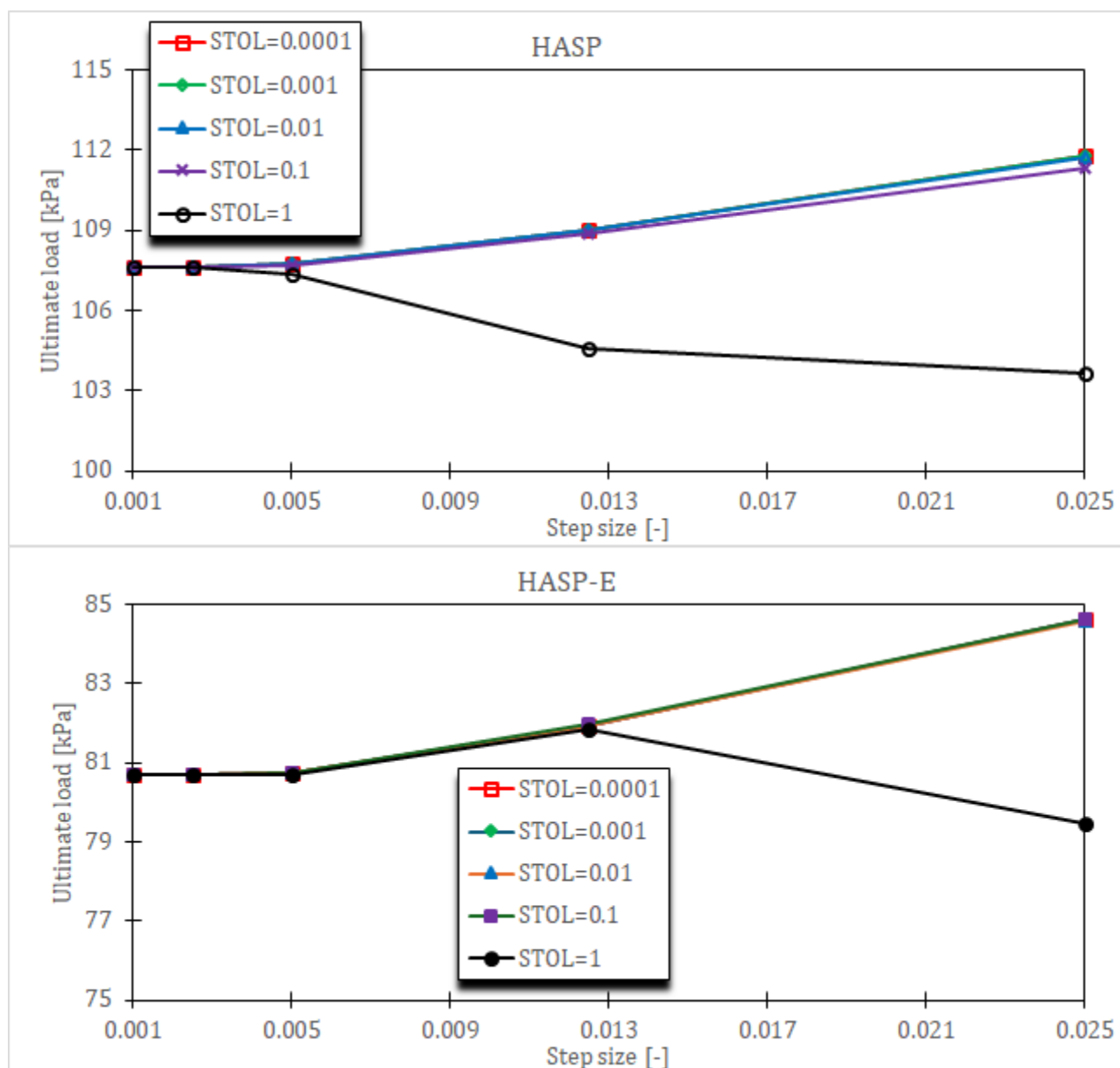
На **Слика 4-21** приказане су граничне носивости тракастог темеља које се добијају за оба модела коришћењем различитих алгоритама и величина инкремената за $POP = 100 \text{ kPa}$.



Слика 4-21 Гранична носивост тракастог темеља за $POP = 100 \text{ kPa}$ коју предвиђају HASP и HASP-E модел за различите алгоритме и величине инкремената

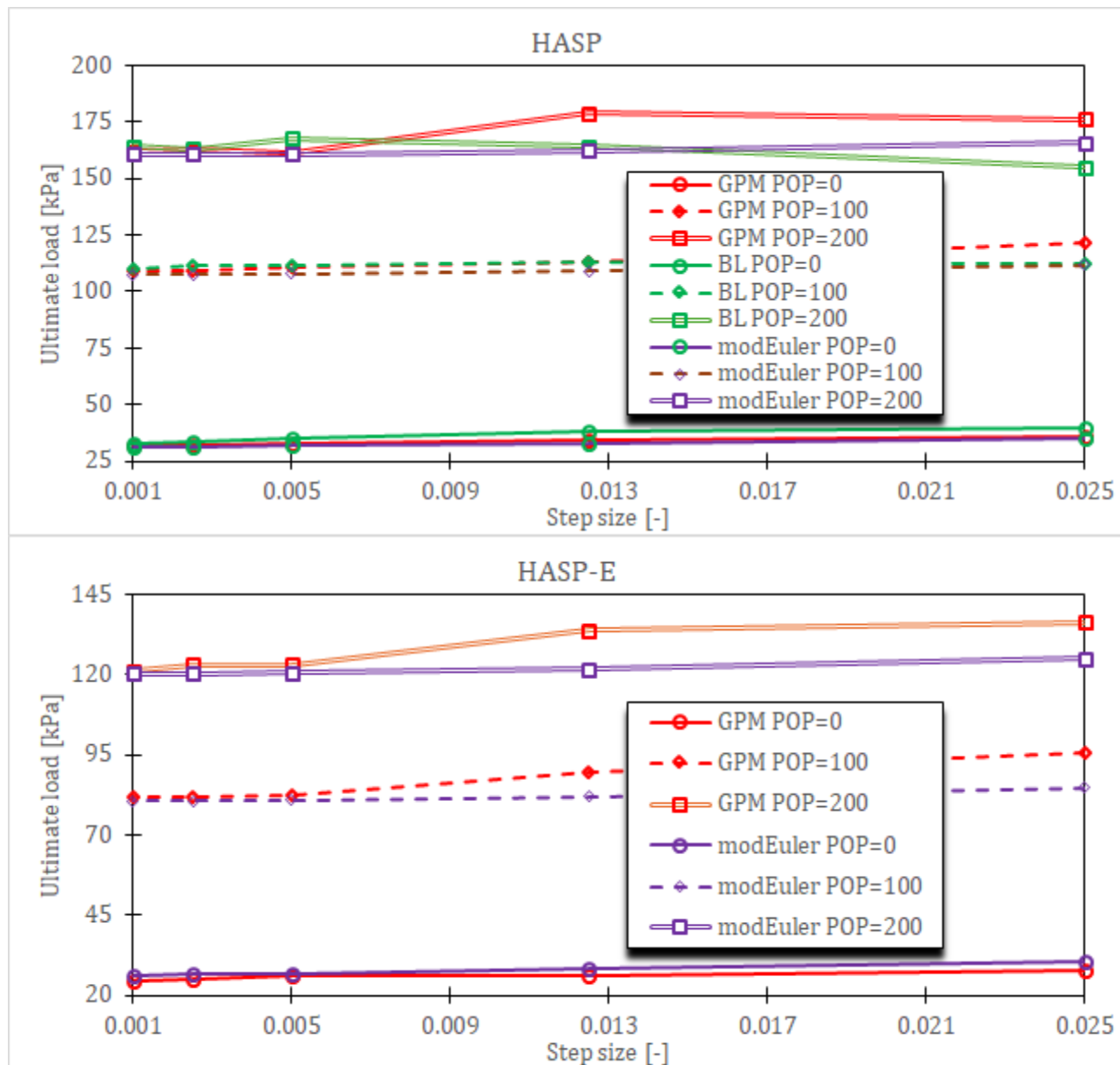
За модификовани Euler-ов поступак са аутоматском субинкрементацијом је коришћена вредност $STOL = 10^{-4}$. За HASP модел, имплицитни алгоритам (Borja & Lee, 1990) предвиђа сличну вредност носивости без обзира на величину инкремента, док метода водећег параметра и експлицитна метода предвиђају већу носивост за мали број инкремената него када се користи већи број инкремената. Исто важи и за HASP-E. За оба модела, сви алгоритми за велики број инкремената предвиђају приближно исту носивост тракастог темеља.

На **Слика 4-22** приказано је како величина $STOL$ код модификованог Euler-овог поступка са аутоматском субинкрементацијом утиче на срачунату граничну носивост темеља за $POP = 100 \text{ kPa}$. Може се уочити да се за вредности $STOL$ мање од 0.1 добијају за оба модела исте граничне носивости за исте величине инкремената.



Слика 4-22 Гранична носивост тракастог темеља за $POP = 100 \text{ kPa}$ за модификовани Euler-ов поступак са аутоматском субинкрементацијом у зависности од величине $STOL$ за различите величине инкремента

На **Слика 4-23** приказан је утицај иницијалних услова на срачунату граничну носивост темеља. За велики број инкремената (мала величина инкремента), све методе нумеричке интеграције предвиђају скоро исту граничну носивост. Са смањивањем броја инкремената (повећавањем величине инкремента), добијају се различите граничне носивости тракастог темеља.



Слика 4-23 Утицај иницијалних услова на срачунату граничну носивости за различите методе нумеричке интеграције и величине инкремената

Из претходних дијаграма може се закључити да алгоритам (Ворџа & Lee, 1990) захтева најмањи утрошак рачунарског времена, али даје решења мање тачности него друга два алгоритма уз проблеме са конвергенцијом глобалног инкрементално-итеративног поступка за јако преконсолидовано тло када се оптерећење наноси у малом броју инкремената. Метода водећег параметра захтева већи утрошак рачунарског времена него алгоритам (Ворџа & Lee, 1990), али даје решења веће тачности. Модификовани Euler-ов поступак са аутоматском субинкрементацијом и контролом грешке захтева

највећи утрошак рачунарског времена, али даје решење највеће тачности. У односу на утицај иницијалних услова и величину инкремента, експлицитни алгоритми су робустнији него имплицитни.

Поглавље 5 Валидација HASP-E модела

5.1 Увод

У овом поглављу извршена је валидација HASP-E модела поређењем публикованих резултата лабораторијских опита или теренских мерења са резултатима нумеричког модела. У потпоглављу 5.2 приказани су резултати нумеричке симулације цикличног консолидованог дренираног триаксијалног опита на Fujiomori глини и извршена је параметарска анализа са циљем да се утврди како и колико промене материјалних параметара конститутивног модела утичу на предвиђање HASP-E модела. У потпоглављу 5.3 приказани су резултати нумеричке симулације цикличног консолидованог недренираног триаксијалног опита на Newfield глини и спроведена је, такође, параметарска анализа. У потпоглављима 5.4 и 5.5 приказане су могућности HASP-E модела да предвиди понашање преконсолидоване глине услед изградње насипа.

5.2 CD циклични триаксијални опити на Fujiomori глини

Симулирана су три циклична CD триаксијална опита на Fujiomori глини. Детаљи нумеричког модела дати су у потпоглављу 4.4.1. Иницијални услови за нумеричке симулације су дати у **Табела 5-1**. Параметри HASP, HASP-E и MCC модела су преузети из (Nakai & Hinokio, 2004), док су параметри OC_CLAY модела одређени према изразима датим у потпоглављу 3.4. Параметри конститутивних модела су дати у **Табела 5-2**. Резултати опита су преузети из (Nakai & Hinokio, 2004) и (Moghadam, et al., 2022). Иницијални коефицијент порозности је одређен из израза (3-48) на основу познатих иницијалних услова. Пошто параметри који описују S-криву нису били на располагању, за вредност G_0^{ref} је усвојена вредност тако да однос модула смицања при веома малим деформацијама и при великим деформацијама буде десет пута ($G_0^{ref}/G_{ur}^{ref} = 10$). За вредност параметра $\gamma_{0,7}$ је усвојена вредност од $2.5 \cdot 10^{-4}$.

Табела 5-1 Иницијални услови за симулацију три CD циклична триаксијална опита на Fujiomori глини

Опит	$OCR[-]$	$p'_i [kPa]$	$e_o [-]$	$\Gamma [-]$
CD1	1	196	0.81	
CD2	1	392	0.746	2.23
CD3	1	196	0.81	

Симулирана су три опита на нормално консолидованој глини. У опитима CD1 и CD2 средњи ефективни напон је константан током целог опита, док је у опиту CD3 радијални напон константан током целог опита.

У опиту CD1, у првом циклусу напонски однос η је повећан до вредности од 0.75 (компресија), а онда је смањен до вредности од -0.6 (екстензија). У другом циклусу је

повећан до вредности од 1.2 (компресија), па смањен до вредности од -0.85 (екстензија). Након тога је напонски однос повећаван до лома у компресији.

У опиту CD2, у прва четири циклуса, напонски однос је повећаван до вредности од 1.2 (компресија) и смањиван до вредности од -0.85 (екстензија). У последњем, петом циклусу, напонски однос је повећан до лома у компресији.

У опиту CD3, напонски однос је прво повећан до 0.75 (компресија), па смањен до -0.6 (екстензија) и онда повећаван до лома у компресији.

Табела 5-2 Параметри HASP, HASP-E, OC_CLAY и MCC модела за симулацију три CD циклична триаксијална опита на Фујинотори глине

Модел		Параметри						
HASP и MCC	λ				M			
	[–]				[–]			
компресија екстензија	0.089	0.02			1.36	0.2		
					0.94			
HASP-E	λ	κ	M_c	M_e	μ	G_0^{ref}	$\gamma_{0,7}$	
	[–]	[–]	[–]	[–]	[–]	[kPa]	[–]	
	0.089	0.02	1.36	0.94	0.2	67875	0.00025	
OC_CLAY	Опит	E_{ur}^{ref}	μ	E_{oed}^{ref}	G_0^{ref}	$\gamma_{0,7}$	ϕ_{cs}	POP
		[kPa]	[–]	[kPa]	[kPa]	[–]	[°]	[kPa]
	CD1	16290	0.2	2033.71	67875	0.0025	33.67	0
	CD2	15714	0.2	1127.34	67875	0.0025	33.67	0
	CD3	16290	0.2	2033.71	67875	0.0025	33.67	0

Од метода нумеричких интеграција за HASP моделе коришћен је само модификовани Euler-ов поступак са аутоматском субинкрементацијом и $STOL = 10^{-4}$. Максимална величина инкремента оптерећења је 0.01.

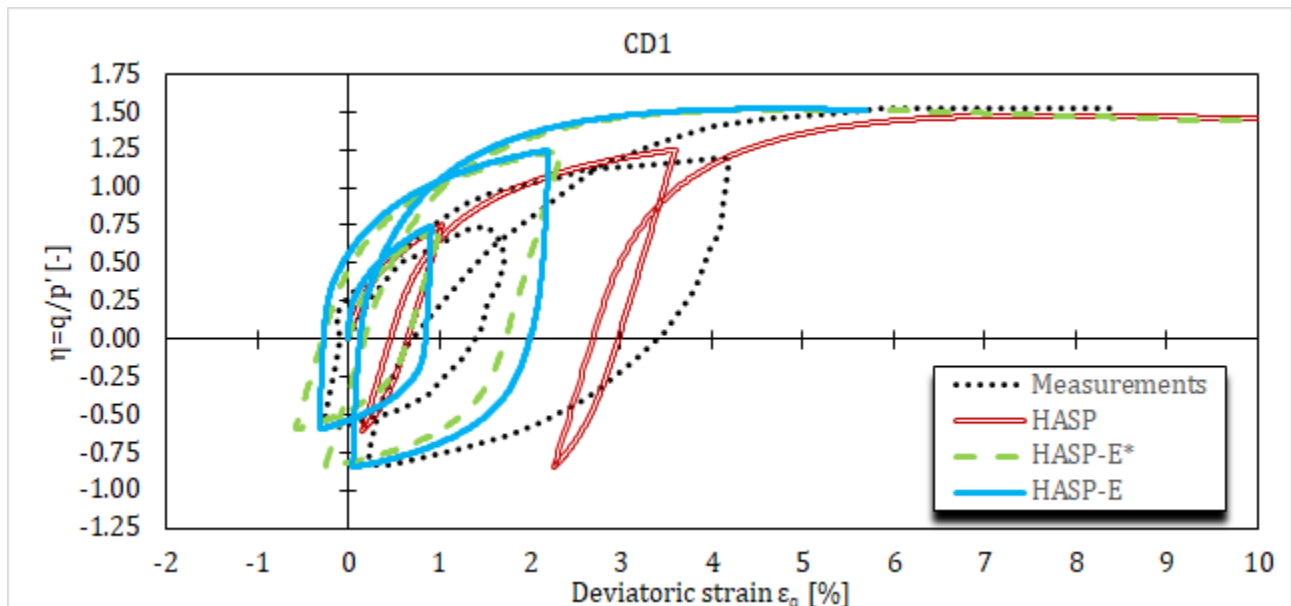
Прво су извршене нумеричке симулације сва три опита за иницијалне вредности материјалних параметара које су дате у **Табела 5-2** користећи HASP, HASP-E са неактивном “Brick” компонентом (HASP-E* на дијаграмима), HASP-E са активном “Brick” компонентом, OC_CLAY модел са активним SSO моделом и MCC модел. Циљ ове анализе је био да се испитају међусобне разлике ових модела у предвиђању понашања глине и колико се предвиђено понашање разликује од експерименталних резултата за сваки конститутивни модел.

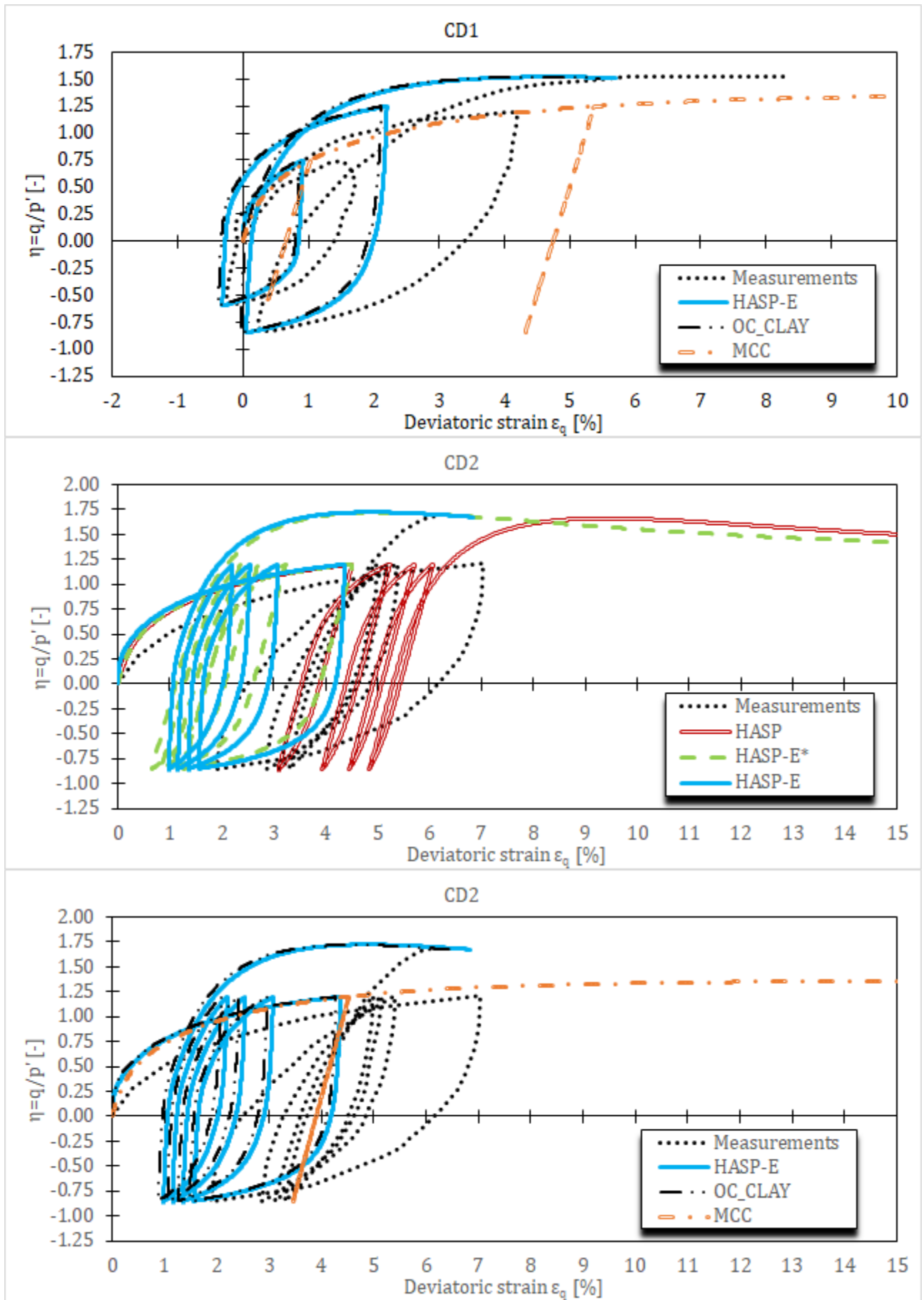
Након тога извршена је параметарска анализа са циљем да се испита како промена материјалних параметара HASP-E модела утиче на предвиђање понашања глине. У свакој анализи вредност само једног материјалног параметра је варирана, док су вредности осталих материјалних параметара биле једнаке иницијалним вредностима које су дате у **Табела 5-2**. Од материјалних параметара варирани су: λ , κ , G_0^{ref} и $\gamma_{0,7}$. Вредности параметара λ и κ су биране тако да напонско-деформацијске криве добијене из нумеричких симулација буду што приближније експерименталним кривама. Вредности G_0^{ref} су биране тако да однос G_0^{ref}/G_{ur}^{ref} има вредности 10/5/2.5. За параметар $\gamma_{0,7}$ су коришћене вредности од $2.5 \cdot 10^{-4}$ и $5.0 \cdot 10^{-4}$.

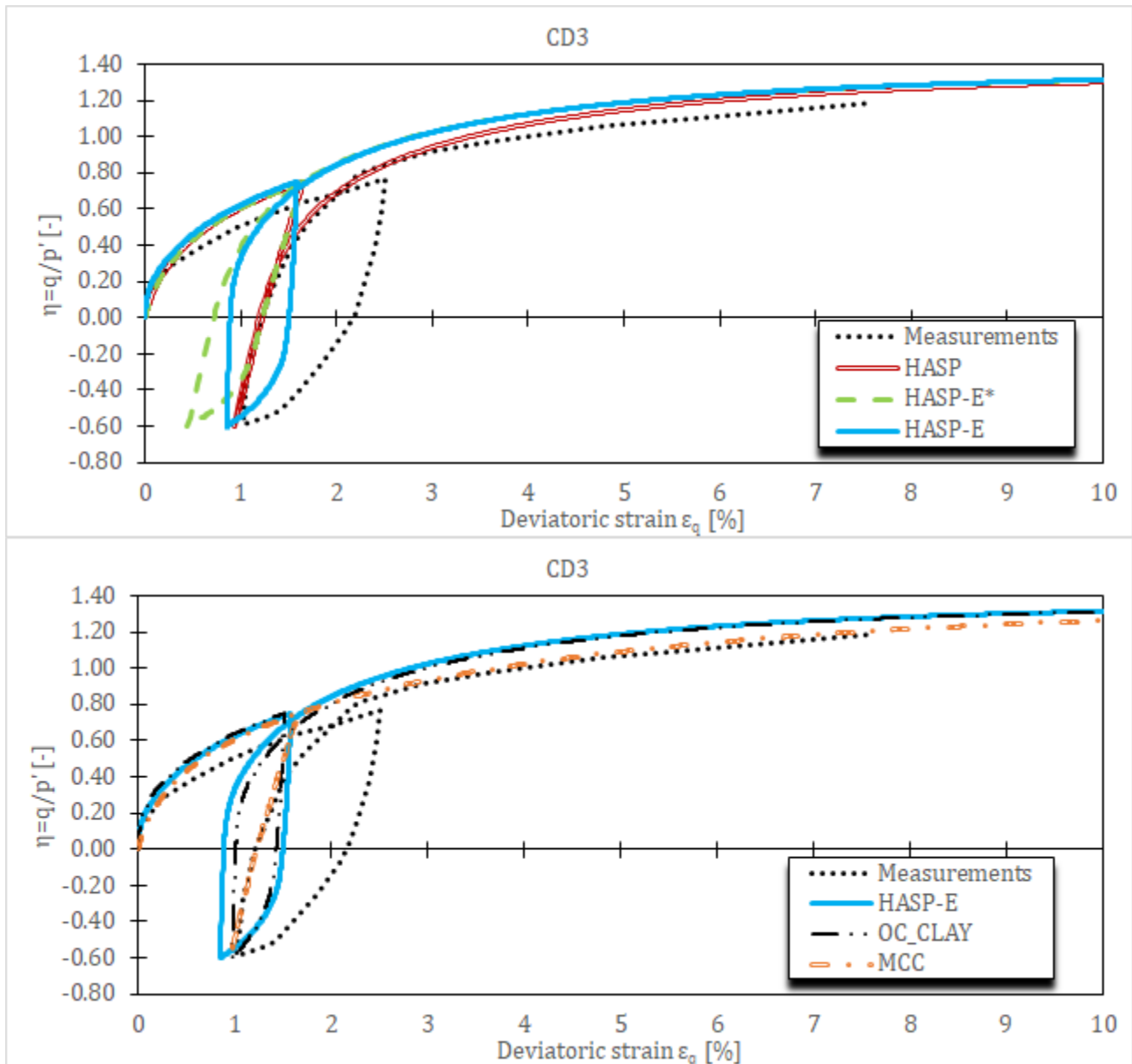
5.2.1 Резултати и дискусија

Резултати нумеричких симулација за све конститутивне моделе и иницијални сет материјалних параметара дати су на **Слика 5-1** и **Слика 5-2**. Из приказаних резултата може се уочити да предвиђања МСС модела веома одступају од експерименталних резултата. Иако је глина на почетку сваког опита нормално консолидована, након растеређивања и при поновном оптеређивању постаје преконсолидована и МСС модел не може да опише понашање глине при цикличном оптеређењу. Сви конститутивни модели дају исте напонско-деформацијске криве приликом оптеређивања у првом циклусу када је глина нормално консолидована. HASP модел не предвиђа адекватно понашање глине, али даје боља предвиђања него МСС модел, поготово током циклуса растеређења и поновног оптеређења. МСС модел предвиђа искључиво еластичне деформације током циклуса оптеређења и растеређења, док HASP поред еластичних предвиђа и пластичне деформације. Иако HASP модел може донекле да репродукује хистерезис који се јавља у једном циклусу оптеређења и растеређења, величина хистерезиса је мања и девијаторске деформације су веће него измерене.

Предвиђања понашања глине која су блиска понашању забележеном у експерименту се могу уочити код HASP-E модел без активне "Brick" компоненте. Разлог за то је што HASP-E модел предвиђа различиту чврстоћу тла у триаксијалној компресији и екстензији. Узимањем у обзир само утицај Lode-овог угла, HASP-E може реалније да опише понашање глине при цикличном оптеређењу у односу на HASP. Са активном "Brick" компонентом, предвиђања HASP-E модела се додатно побољшавају. OC_CLAY модел даје иста предвиђања као и HASP-E модел и као што је већ приказано у потпоглављима 4.4.4 и 4.4.5, разлике у предвиђањима понашања глине између ова два модела би могла да буду последица различитог облика S-криве. OC_CLAY и HASP-E у опиту CD1 и CD2 предвиђају мање запреминске и девијаторске деформације за иницијални сет параметара него што су измерене.







Слика 5-1 Напонско-деформацијске криве за CD цикличне триаксијалне опите на Фијипотори глини за различите конститутивне моделе и иницијални сет материјалних параметара

У опиту CD3 сви конститутивни модели предвиђају запреминске и девијаторске деформације које су блиске измереним. Боље поклапање предвиђања понашања глина са експерименталним подацима би могли да се постигну уколико би се извршила калибрација неких од материјалних параметара конститутивних модела.

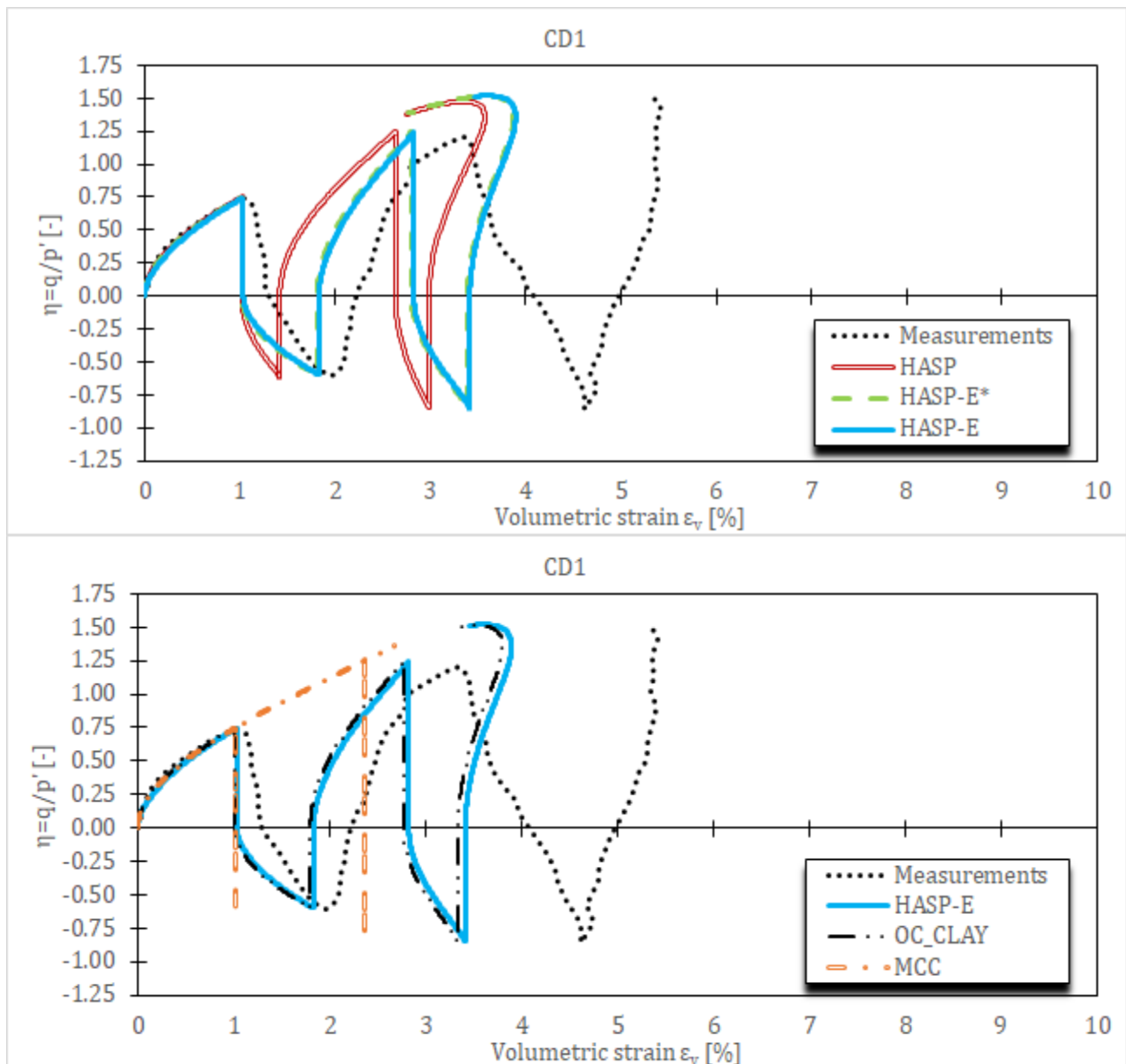
У наставку су дати резултати параметарске анализе. Све нумеричке симулације у параметарској анализи су спроведене само са HASP-E моделом са активном “Brick” компонентном.

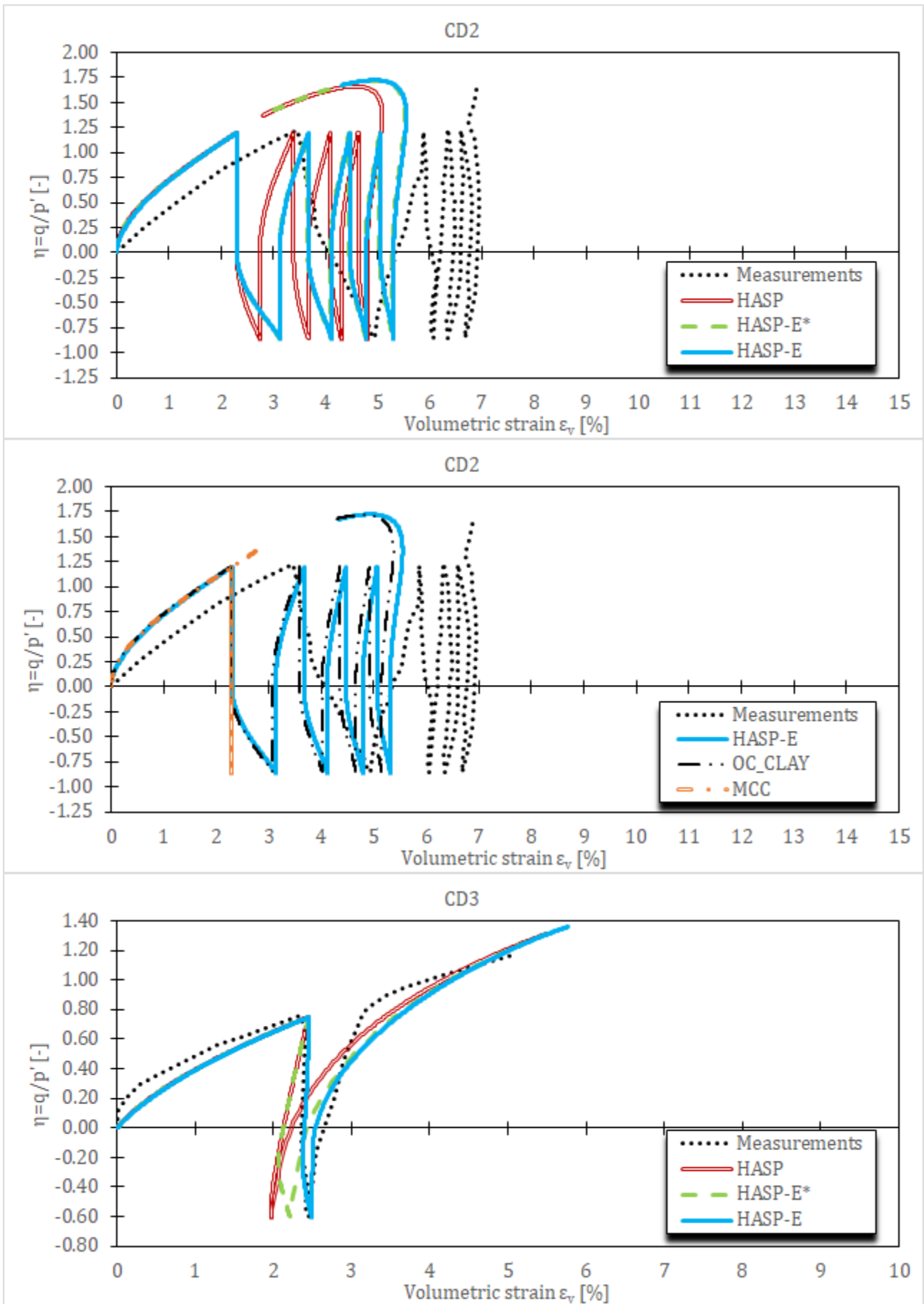
Прво је испитиван утицај параметра λ на предвиђање понашања глина. Варирана је вредност само параметра λ , док су остали параметри остали исти. Резултати нумеричких симулација дати су на **Слика 5-3** и **Слика 5-4**. Вредност параметра λ је

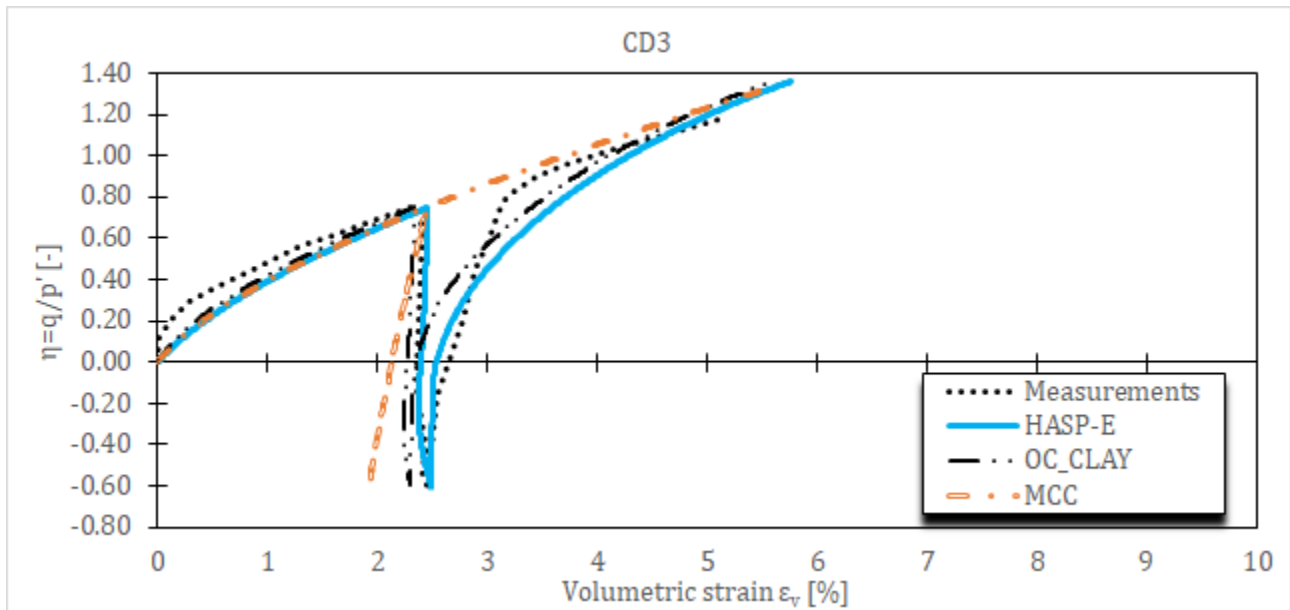
повећавана у односу на иницијалну вредност како би се резултати нумеричких симулација што више поклапали са експерименталним резултатима. Разматране су вредности за λ које су 1.5 и 2.0 пута веће од иницијалне вредности.

Из напонско-деформацијских кривих се може уочити да се за вредност $\lambda = 0.1335$ за опите CD2 и CD3 добијају резултати који се релативно добро поклапају са експерименталним резултатима, док се за опит CD1 боља поклапања добијају за вредност $\lambda = 0.178$. Такође, са повећавањем вредности λ величина хистерезиса се повећава и девијаторске деформације се повећавају.

Код опита CD3 најбоље поклапање величине запреминских деформација се добија за иницијалну вредност параметра λ . Код опита CD1 и CD2 за иницијалну вредност параметра λ се добијају мање запреминске деформације него измерене. Са порастом λ расту запреминске деформације. Код опита CD1 и CD2 се може уочити да у последњем циклусу, када се узорак оптерећује до лома, запреминске деформације прво расту, па онда опадају.





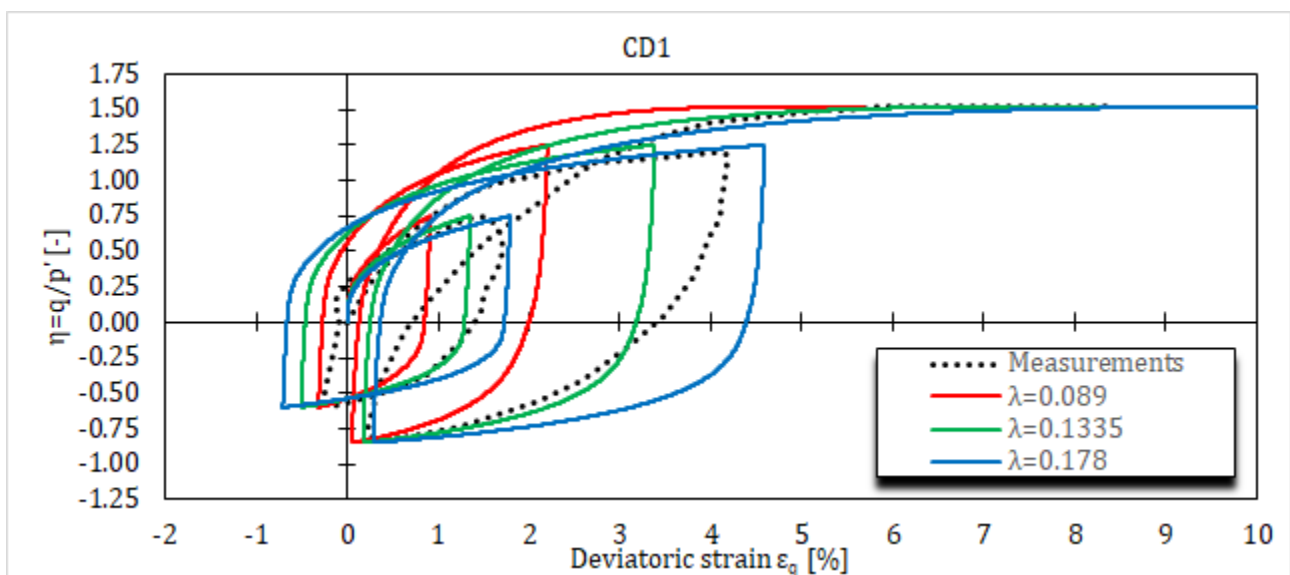


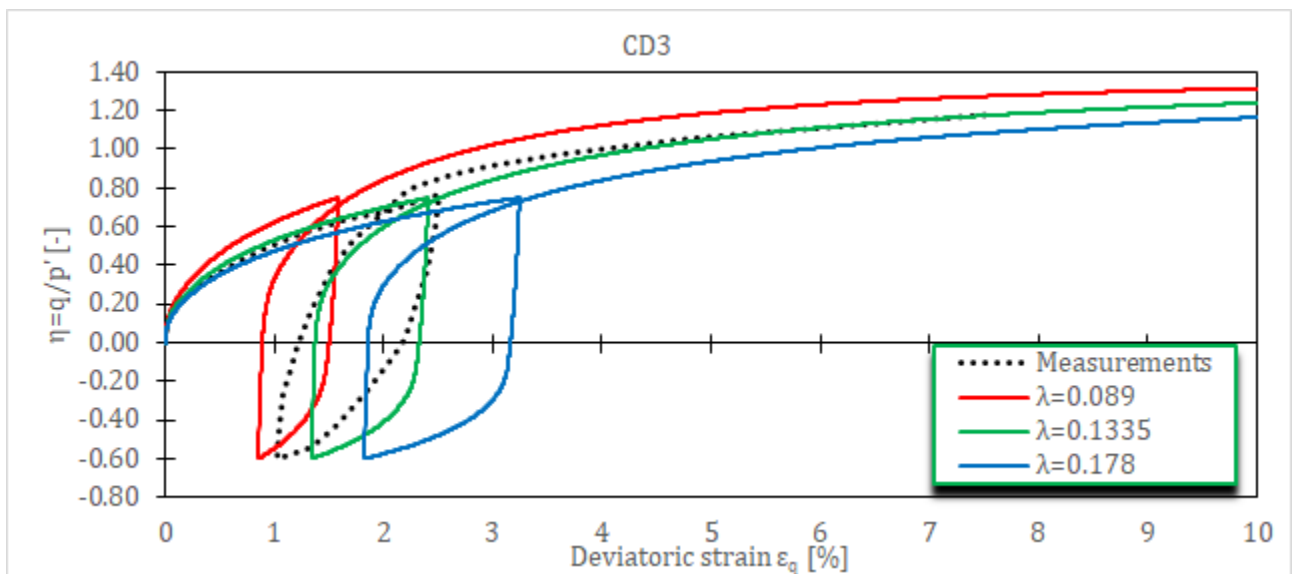
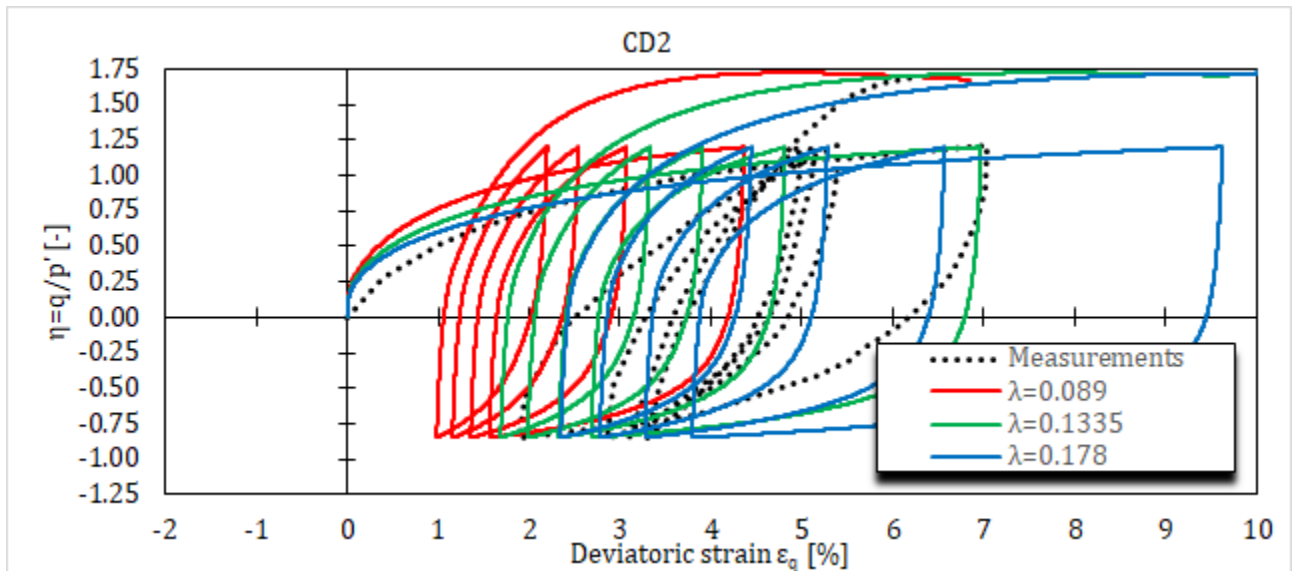
Слика 5-2 Запреминске деформације за CD цикличне триаксијалне опите на Fujiyomori глини за различите конститутивне modele и иницијални сет материјалних параметара

Утицај параметра k на предвиђање понашања глина дат је на **Слика 5-5** и **Слика 5-6**.

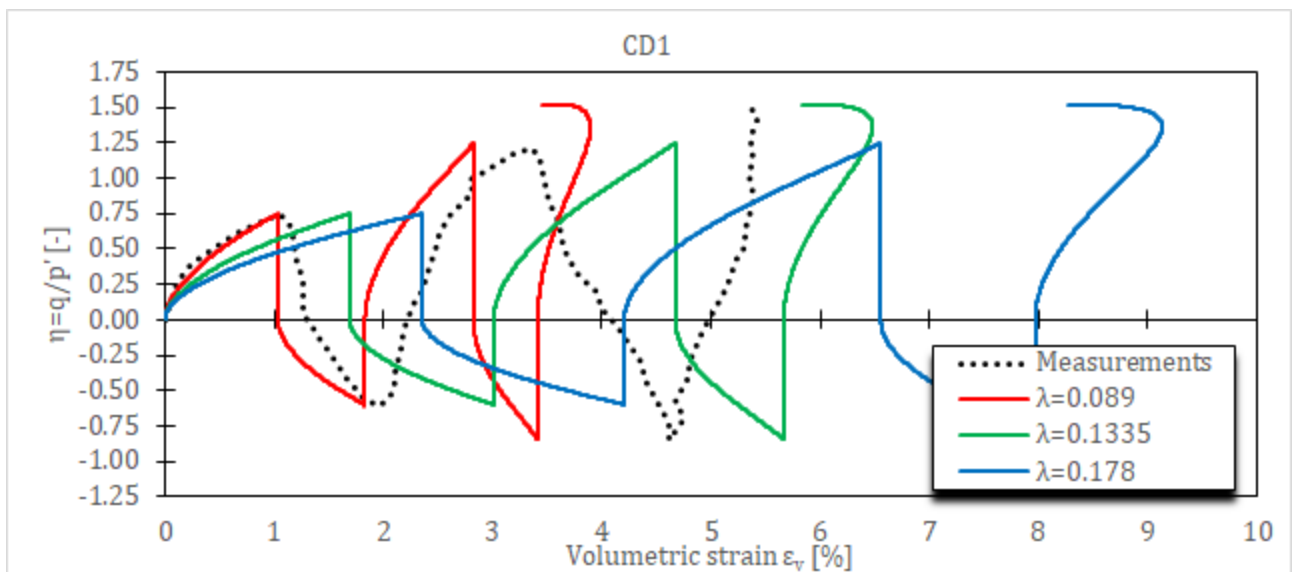
Из напонско-деформацијских кривих за опите CD1 и CD3 се може уочити да промена параметра k веома мало утиче на добијене резултате. Код опита CD2 величина параметра k утиче на величину хистерезиса. Што је мања вредност k , хистерезис је већи. А што је већи хистерезис, веће су и девијаторске деформације при првом циклусу оптерећења.

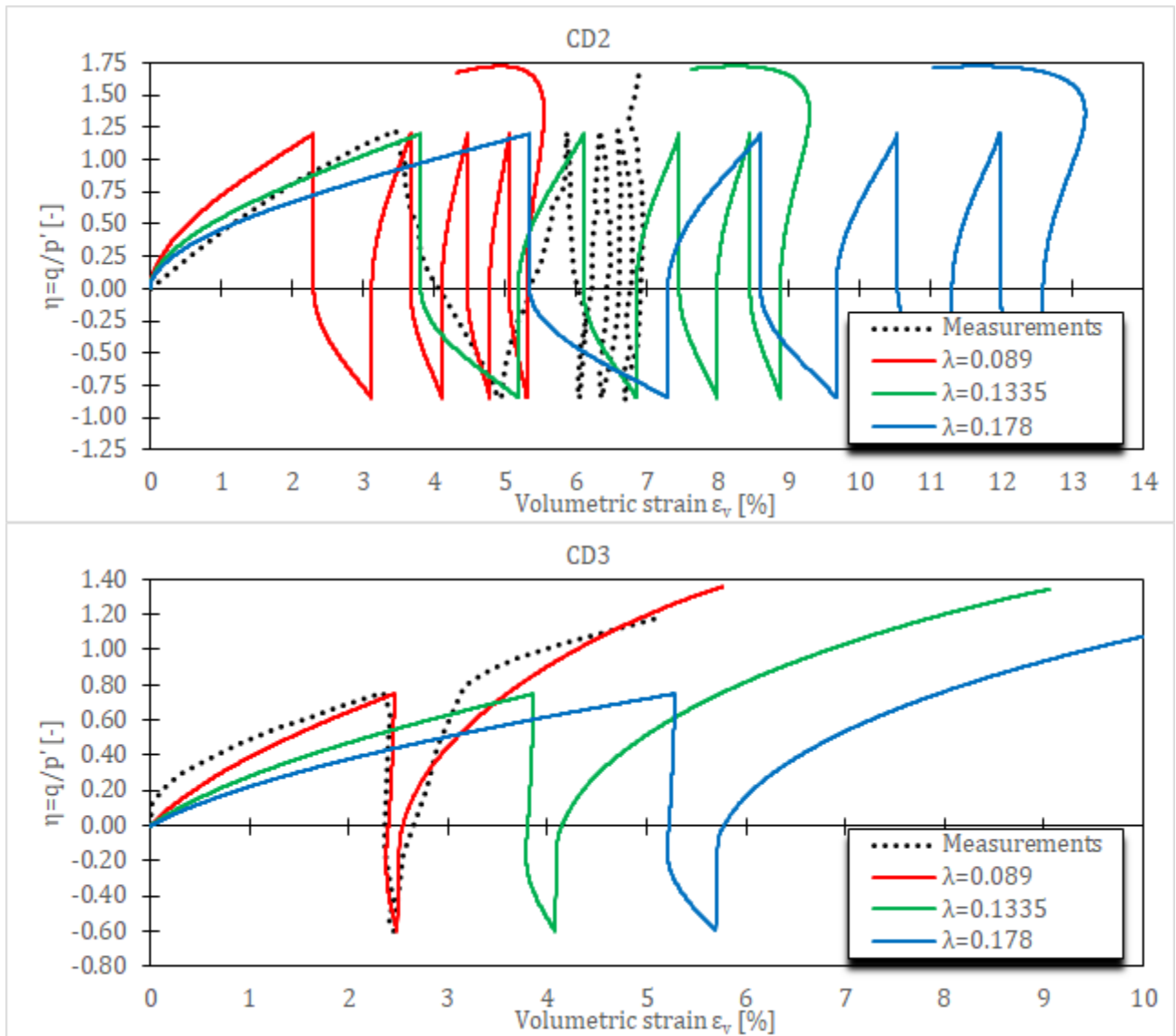
Код опита CD3 најбоље поклапање величине запреминских деформација се добија за иницијалну вредност параметра k . Код сва три опита са порастом вредности k , запреминске деформације опадају, док са смањењем вредности k , запреминске деформације расту. Може се извући закључак да што је већа разлика између параметара λ и k , модел предвиђа веће девијаторске и запреминске деформације.



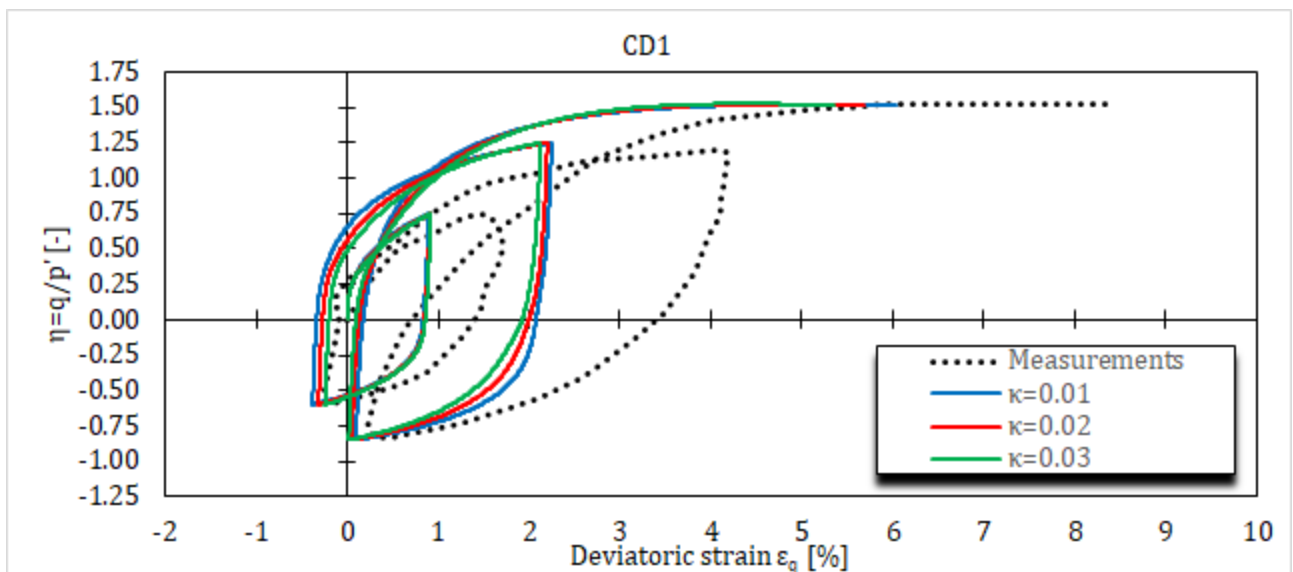


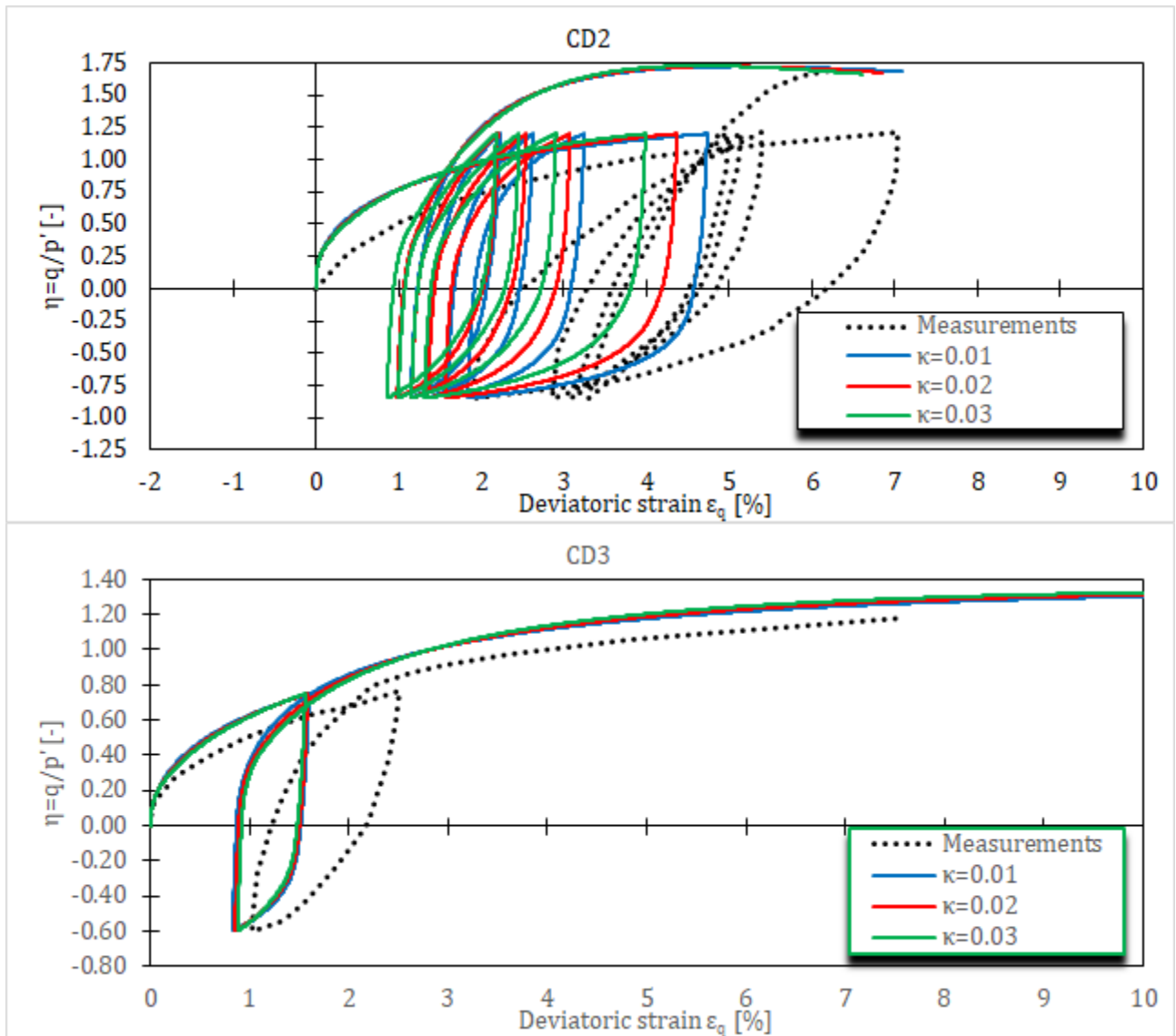
Слика 5-3 Напонско-деформацijske криве за CD цикличне триаксијалне опите на Фујипотоги глини за HASP-E модел и различите вредности параметра λ



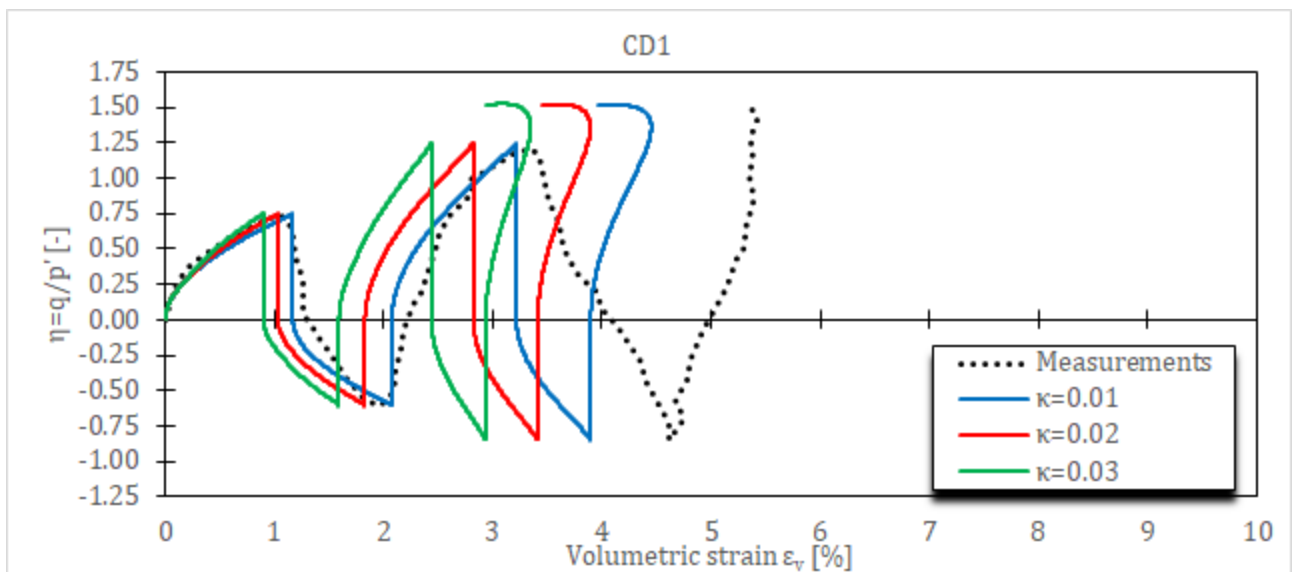


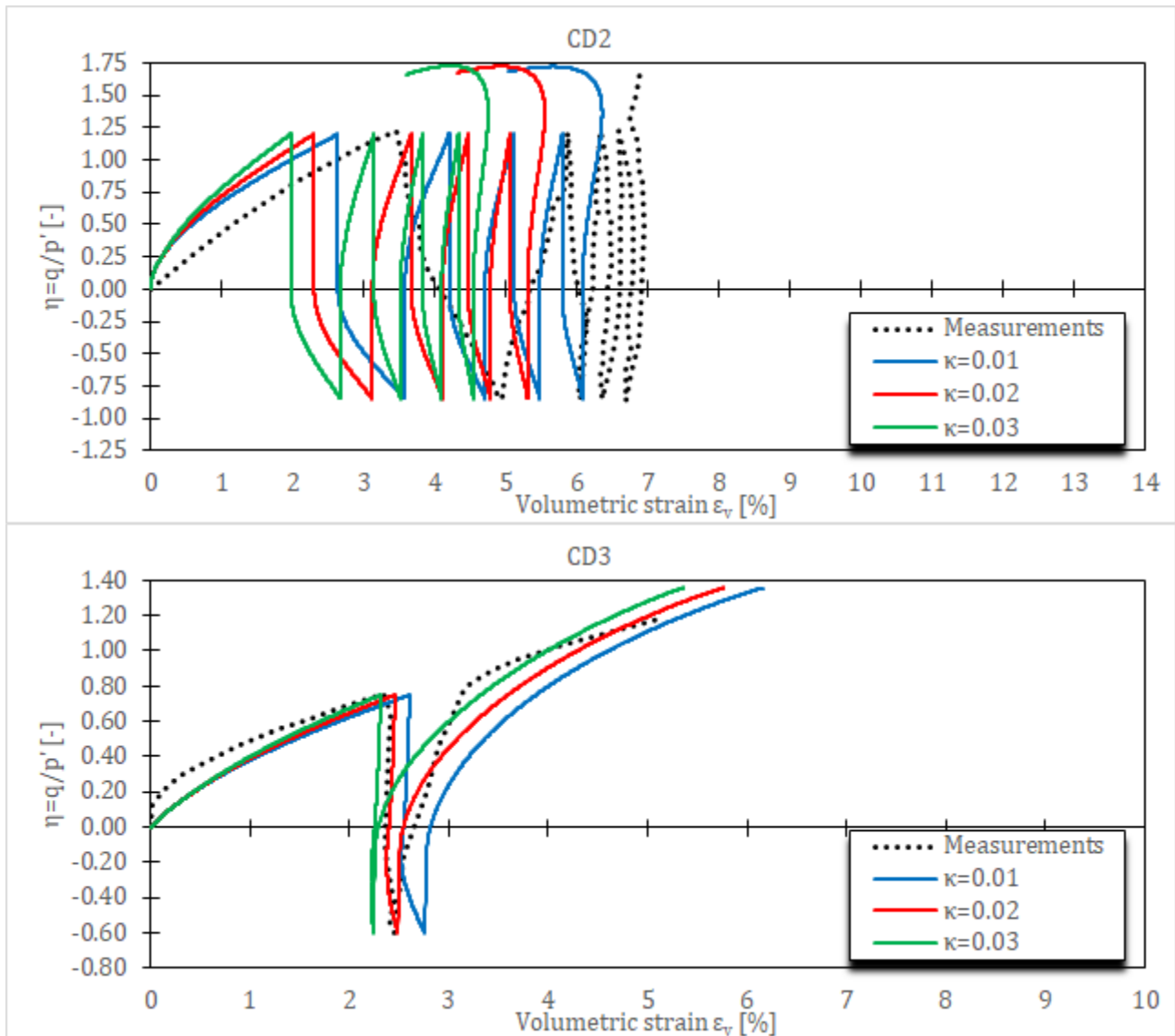
Слика 5-4 Запреминске деформације за CD цикличне триаксијалне опите на Фујитомори глини за HASP-E модел и различите вредности параметра λ





Слика 5-5 Напонско-деформацијске криве за CD цикличне триаксијалне опите на Фујипотоги глини за HASP-E модел и различите вредности параметра κ





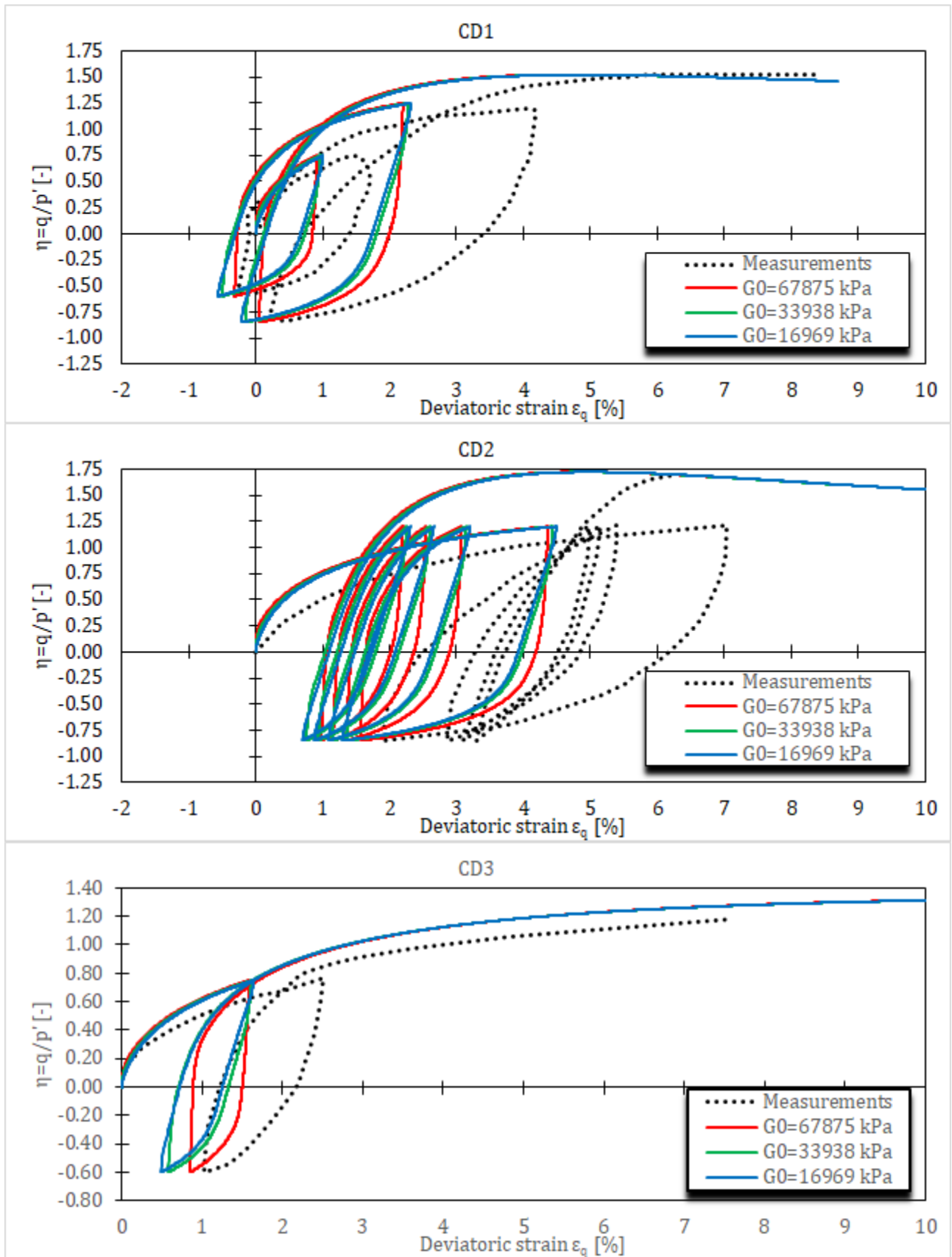
Слика 5-6 Запреминске деформације за CD цикличне триаксијалне опите на Fujiyomori глини за HASP-E модел и различите вредности параметра κ

Утицај величине G_0^{ref} приказан је на **Слика 5-7** и **Слика 5-8**.

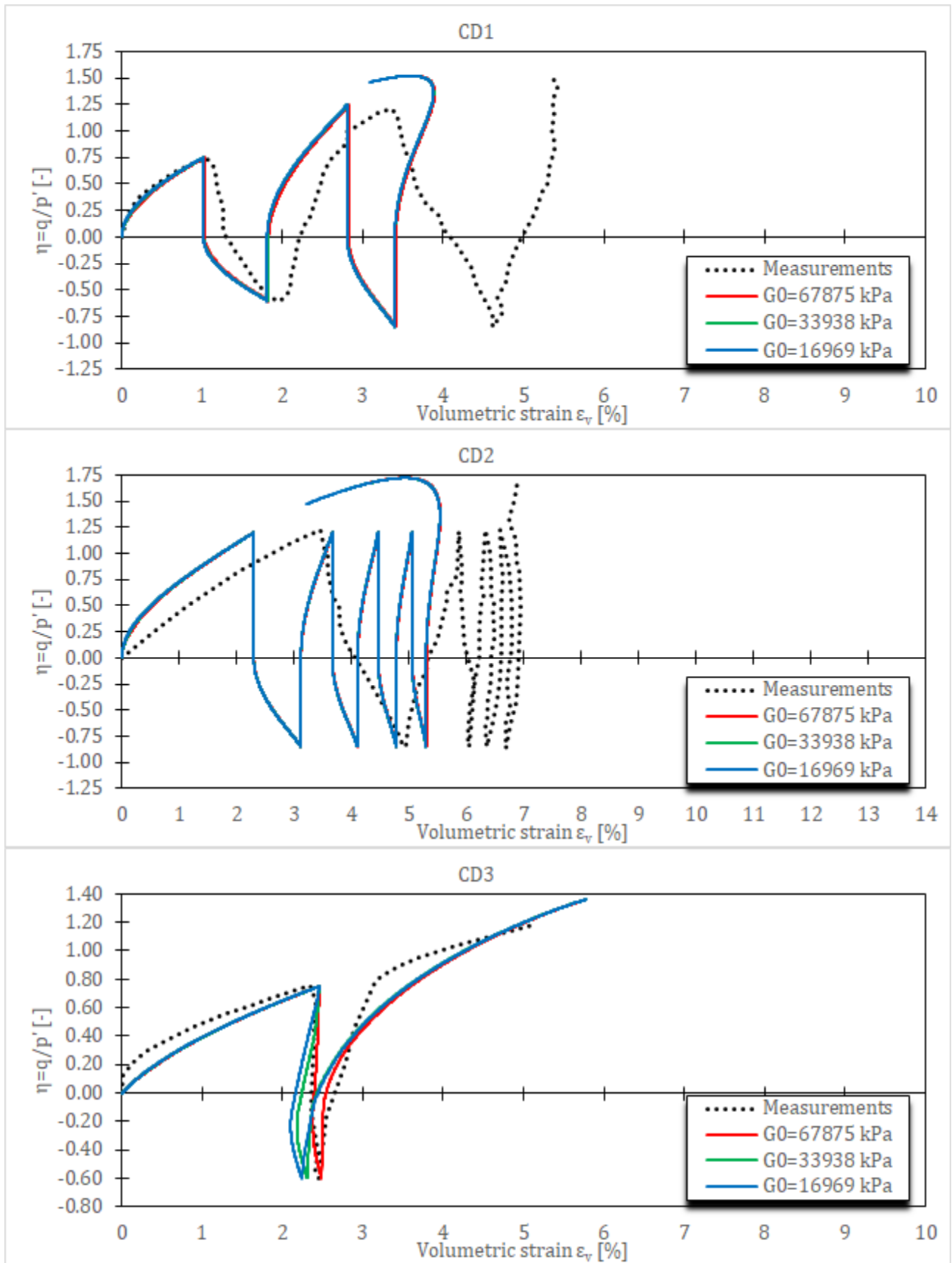
Из напонско-деформацијских кривих се може уочити да што је вредност G_0^{ref} већа, стрмији је нагиб кривих након растеређења и поновног оптерећивања. Величина хистерезиса се веома мало мења са променом G_0^{ref} . Величина запреминских деформација практично не зависи од величине G_0^{ref} , сем у опиту CD3 где се могу уочити мале разлике приликом растерећивања. Са смањењем вредности G_0^{ref} , долази до смањивања запреминских деформација приликом растерећивања.

Утицај величине параметра $\gamma_{0,7}$ приказан је на **Слика 5-9** и **Слика 5-10**.

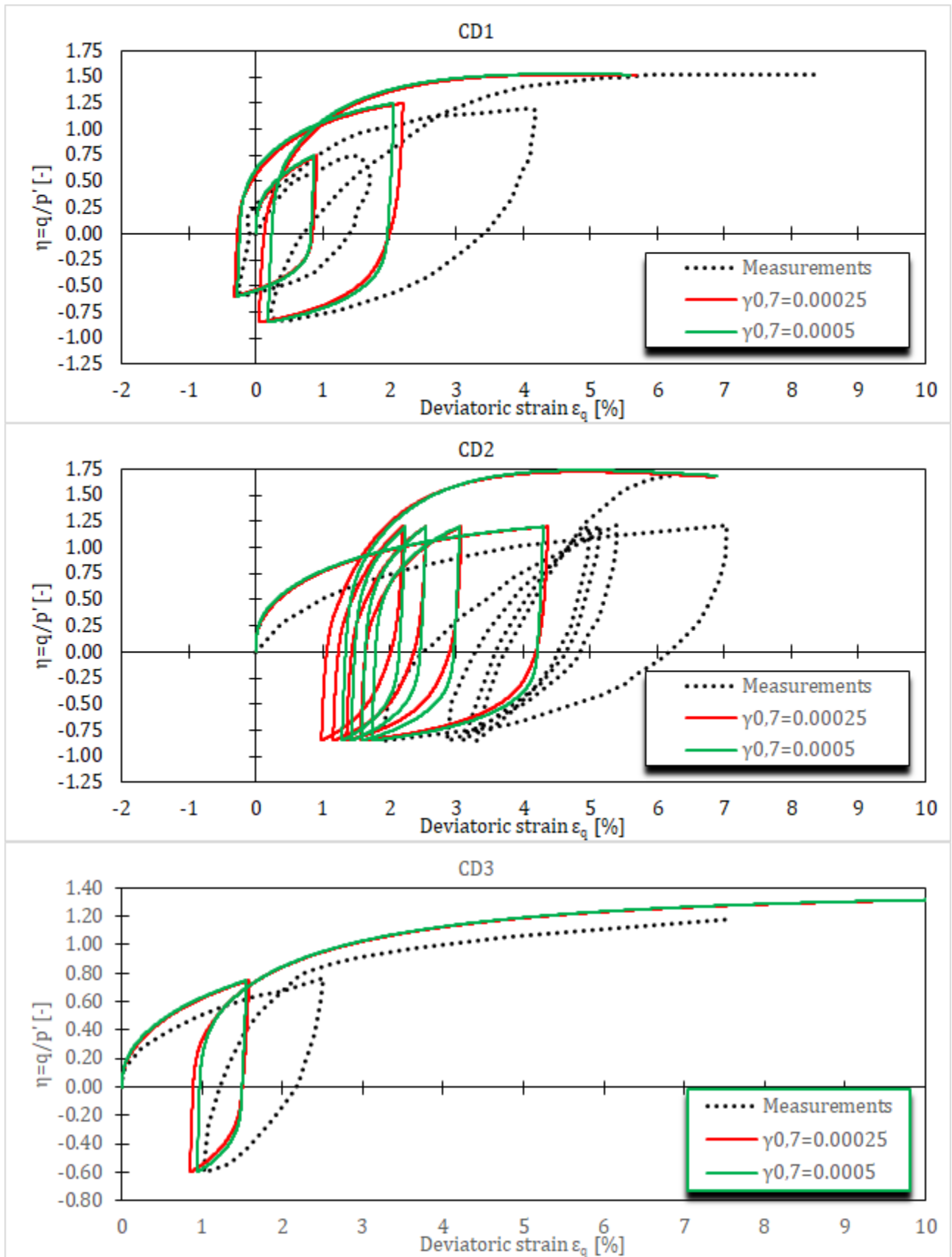
Из напонско-деформацијских кривих се може уочити да са порастом $\gamma_{0,7}$ криве постају стрмије након растеређења и поновног оптерећивања и да се добијају веће запреминске деформације.



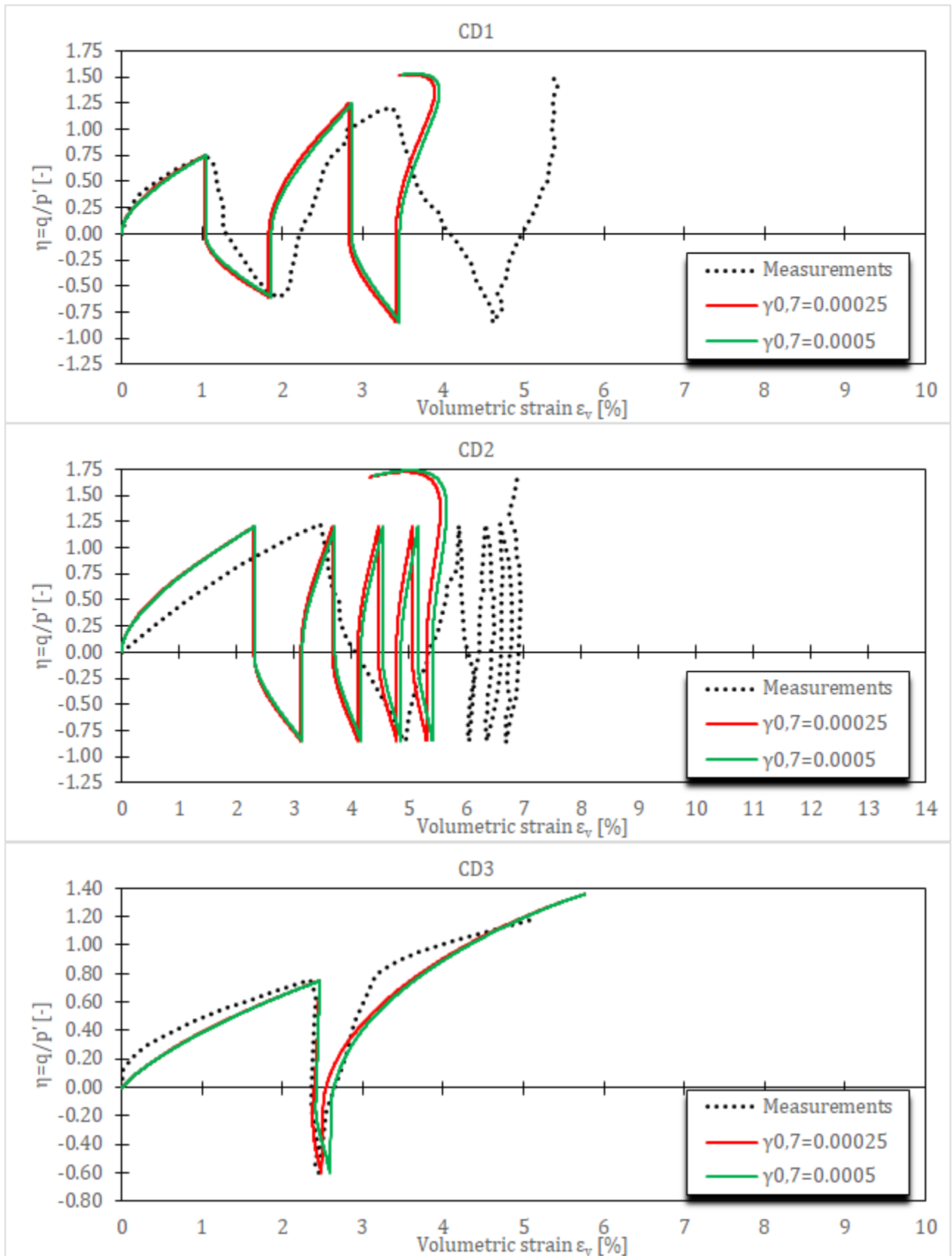
Слика 5-7 Напонско-деформацијске криве за CD цикличне триаксијалне опите на Фујјотори глини за HASP-E модел и различите вредности параметра G_0^{ref}



Слика 5-8 Запреминске деформације за CD цикличне триаксијалне опите на Фуџипотри глини за HASP-E модел и различите вредности параметра G_0^{ref}



Слика 5-9 Напонско-деформацијске криве за CD цикличне триаксијалне опите на Фуџијотори глини за HASP-E модел и различите вредности параметра $\gamma_{0,7}$

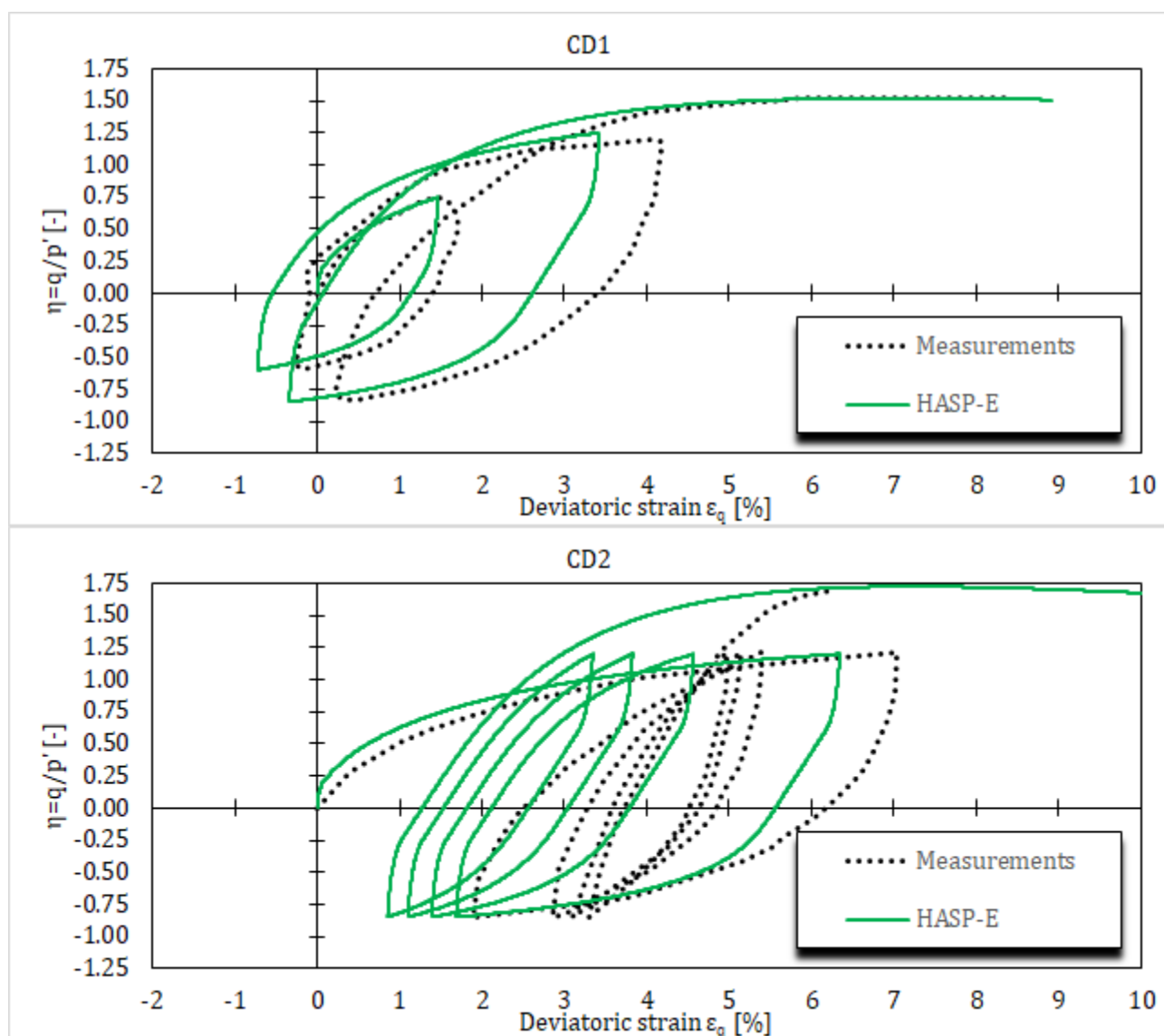


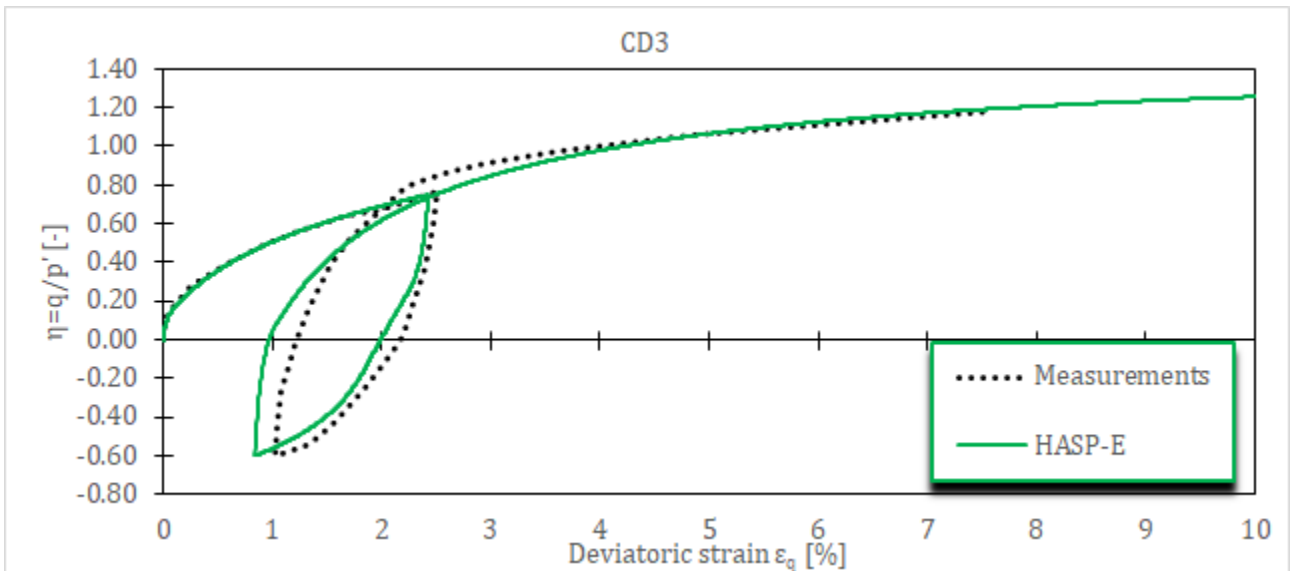
Слика 5-10 Запреминске деформације за CD цикличне триаксијалне опите на Фуџијотори глини за HASP-E модел и различите вредности параметра $\gamma_{0,7}$

На основу резултата параметарске анализе, са циљем да се резултати нумеричких симулација за HASP-E модел што више поклапају се мерењима, измењене су вредности параметара λ , κ и G_0^{ref} у односу на вредности дате у **Табела 5-2**. Односно, у одређеној мери је извршена калибрација ових параметара HASP-E модела. Усвојени сет материјалних параметара дат је у **Табела 5-3**. Иницијални услови (вредност иницијалног коефицијента порозности) дати у **Табела 5-1** су остали непромењени. Резултати нумеричких симулација за HASP-E модел и усвојени сет материјалних параметара дати су на **Слика 5-11** и **Слика 5-12**.

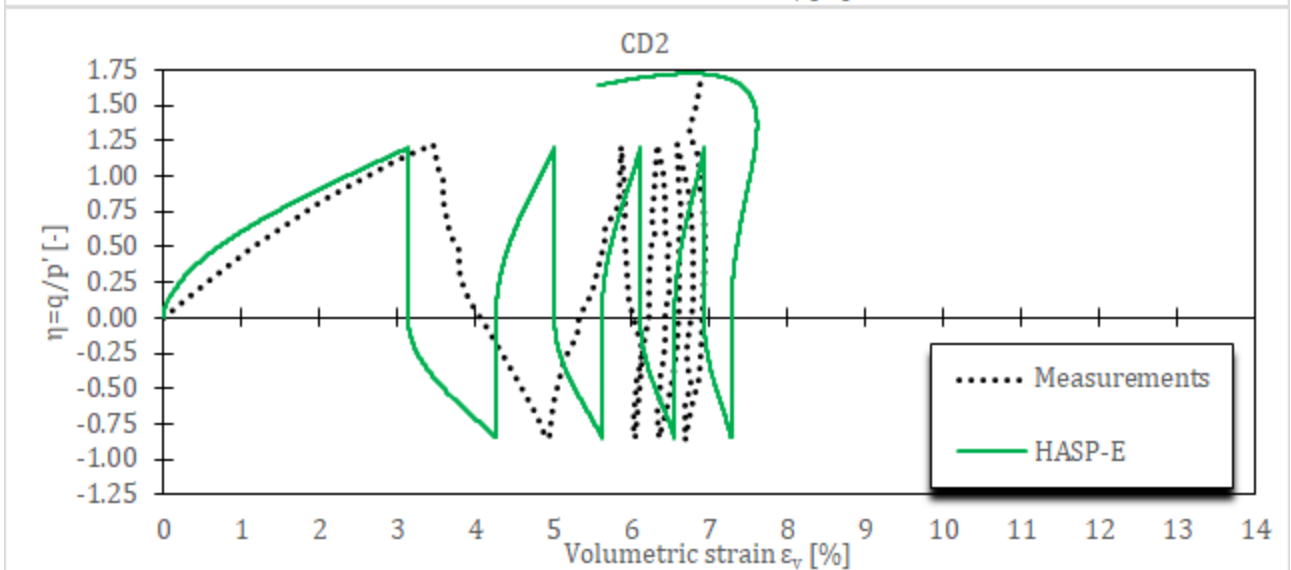
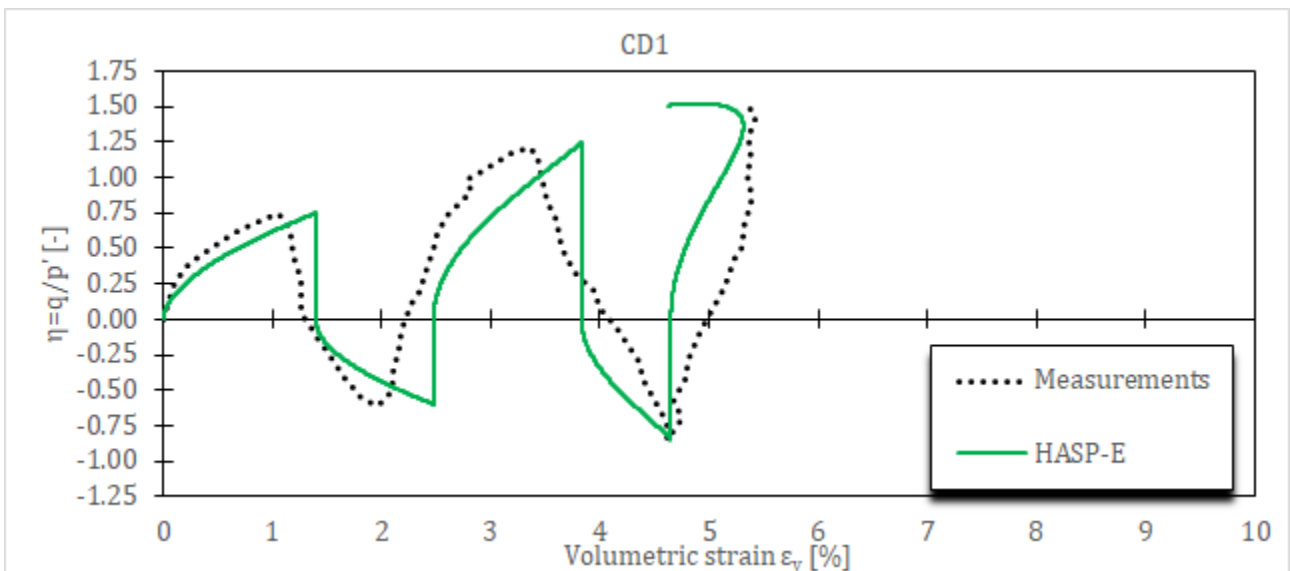
Табела 5-3 Усвојени сет параметара HASP-E модела за симулацију три CD циклична триаксијална опита на Фујипотори глини

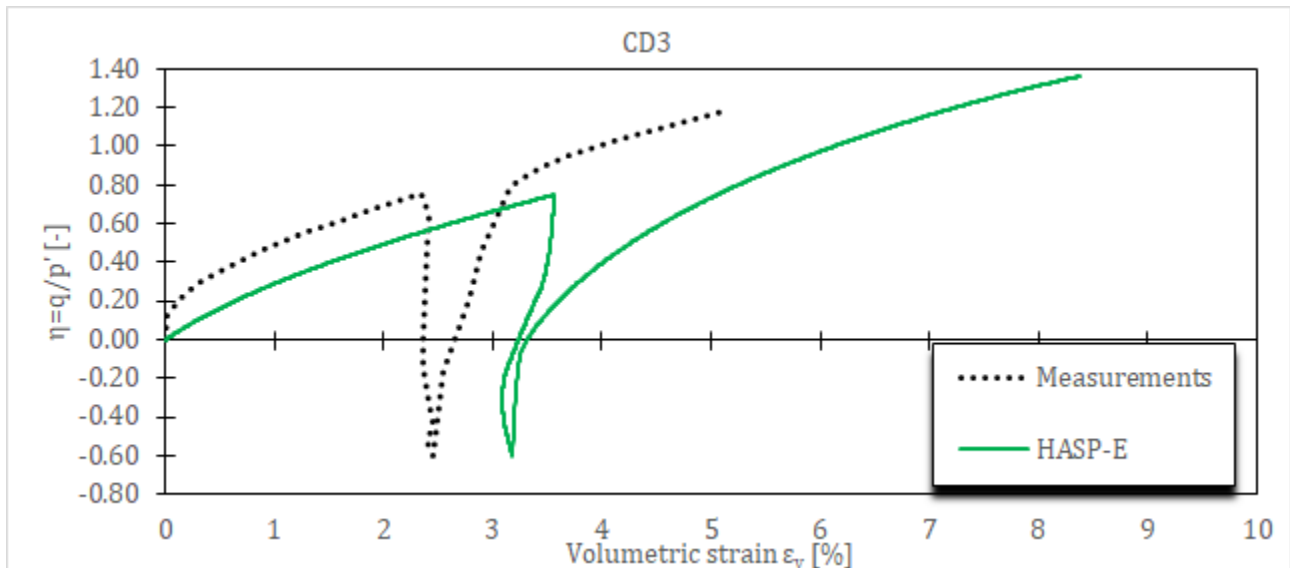
Модел	Параметри						
	λ [-]	κ [-]	M_c [-]	M_e [-]	μ [-]	G_0^{ref} [kPa]	$\gamma_{0,7}$ [-]
HASP-E	0.1335	0.04	1.36	0.94	0.2	33937.5	0.00025





Слика 5-11 Напонско-деформацијске криве за CD цикличне триаксијалне опите на Фујипотоги глини за HASP-E модел и усвојени сет материјалних параметара





Слика 5-12 Запреминске деформације за CD цикличне триаксијалне опите на Fujiyonomori глини за HASP-E модел и усвојени сет материјалних параметара

5.3 CU циклични триаксијални опити на Newfield глини

Симулирана су два CU циклична триаксијална опита на Newfield глини. Детаљи нумеричког модела дати су у потпоглављу 4.4.1. Иницијални услови за нумеричке симулације су дати у Табела 5-4. Параметри HASP, HASP-E и MCC модела су преузети из (Moghadam, et al., 2022), док су параметри OC_CLAY модела одређени према изразима датим у потпоглављу 3.4. Параметри конститутивних модела су дати у Табела 5-5. Резултати опита се могу наћи у (Sangrey, et al., 1969). Иницијални коефицијент порозности је одређен из израза (3-48) на основу познатих иницијалних услова. Пошто параметри који описују S-криву нису били на располагању, за вредност G_0^{ref} је усвојена вредност тако да однос модула смицања при веома малим деформацијама и при великим деформацијама буде десет пута. За вредност параметра $\gamma_{0,7}$ је усвојена вредност од $2.5 \cdot 10^{-4}$.

Табела 5-4 Иницијални услови за симулацију два CU циклична триаксијална опита на Newfield глини

Опит	OCR[-]	p'_i [kPa]	e_o [-]	Γ [-]
CU1 и CU2	1	393	0.71	2.1

Симулирана су два опита на нормално консолидованој глини. У оба опита иницијални средњи ефективни напон је исти и радијални напони су константни током целог опита. У опиту CU1 узорак глине је био изложен циклусима триаксијалне компресије (укупно 10 циклуса) са амплитудом девијаторског напона од $q_{cyc} = 325 \text{ kPa}$. У опиту CU2 узорак глине је био изложен циклусима триаксијалне компресије (укупно 9 циклуса): осам циклуса са амплитудом девијаторског напона од $q_{cyc} = 300 \text{ kPa}$ и један циклус са амплитудом девијаторског напона од $q_{cyc} = 325 \text{ kPa}$.

Табела 5-5 Параметри HASP, HASP-E, OC_CLAY и MCC модела за симулацију два CU циклична триаксијална опита на Newfield глини

Модел	Параметри							
HASP и MCC	λ [–]		κ [–]		M [–]		μ [–]	
компресија	0.07		0.035		1.2		0.2	
екстензија					0.8			
HASP-E	λ [–]	κ [–]	M_c [–]	M_e [–]	μ [–]	G_0^{ref} [kPa]	$\gamma_{0,7}$ [–]	
	0.07	0.035	1.2	0.8	0.2	36643	0.00025	
OC_CLAY	Опит	E_{ur}^{ref} [kPa]	μ [–]	E_{oed}^{ref} [kPa]	G_0^{ref} [kPa]	$\gamma_{0,7}$ [–]	ϕ_{cs} [°]	POP [kPa]
	CU1 и CU2	8794.29	0.2	2442.86	36643	0.00025	30	0

Од метода нумеричких интеграција за HASP моделе коришћен је само модификовани Euler-ов поступак са аутоматском субинкрементацијом и $STOL = 10^{-4}$. Максимална величина инкремента оптерећења је 0.01.

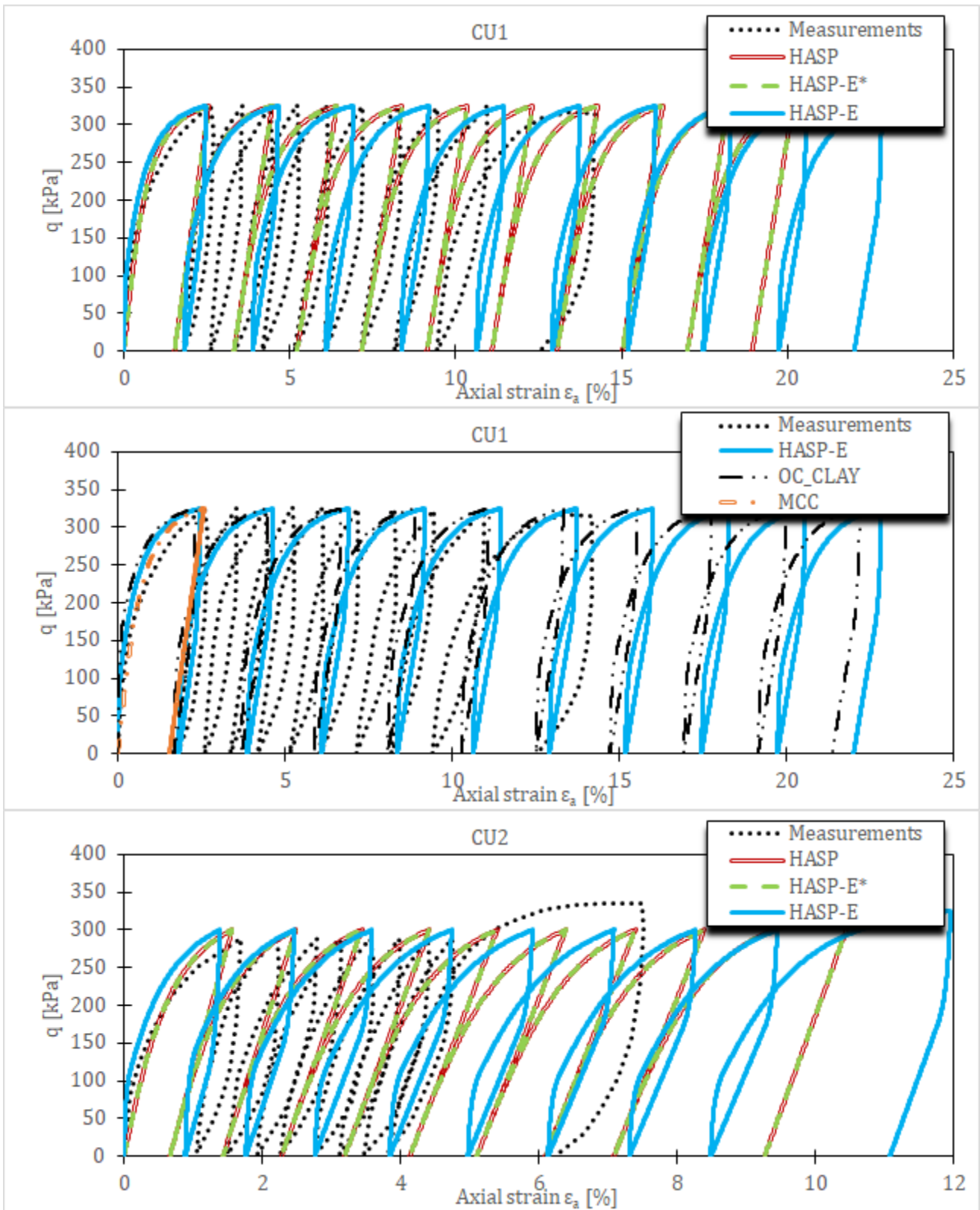
Као у претходном потпоглављу, прво су извршене нумеричке симулације оба опита за иницијалне вредности материјалних параметара датих у **Табела 5-5** користећи HASP, HASP-E са неактивном “Brick” компонентом (HASP-E* на дијаграмима), HASP-E са активном “Brick” компонентом, OC_CLAY модел са активним SSO моделом и MCC модел. Након тога извршена је параметарска анализа за HASP-E модел на исти начин као у претходном потпоглављу.

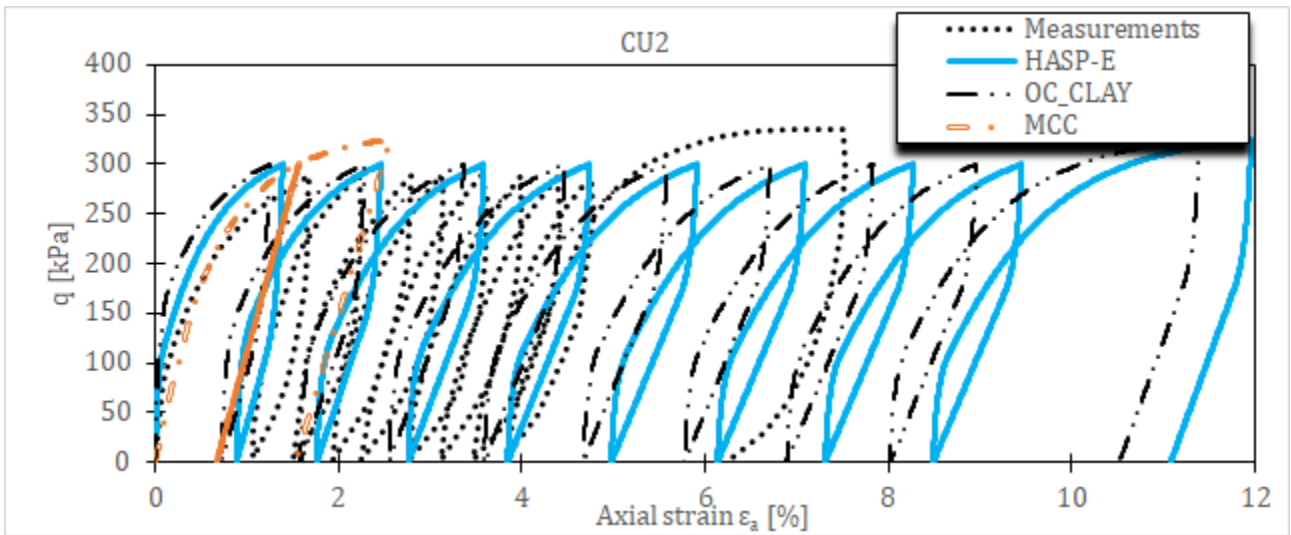
5.3.1 Резултати и дискусија

Резултати нумеричких симулација за све конститутивне моделе и иницијални сет материјалних параметара дати су на **Слика 5-13** и **Слика 5-14**. Из приказаних резултата може се уочити да MCC модел не може да предвиди понашање глине при цикличном оптерећењу у недренираним условима. HASP и HASP-E модел без активне “Brick” компоненте дају идентично предвиђање понашања глине због тога што је узорак изложен само циклусима триаксијалне компресије. Ова два модела боље описују понашање глине него MCC модел, али не предвиђају хистерезисно понашање.

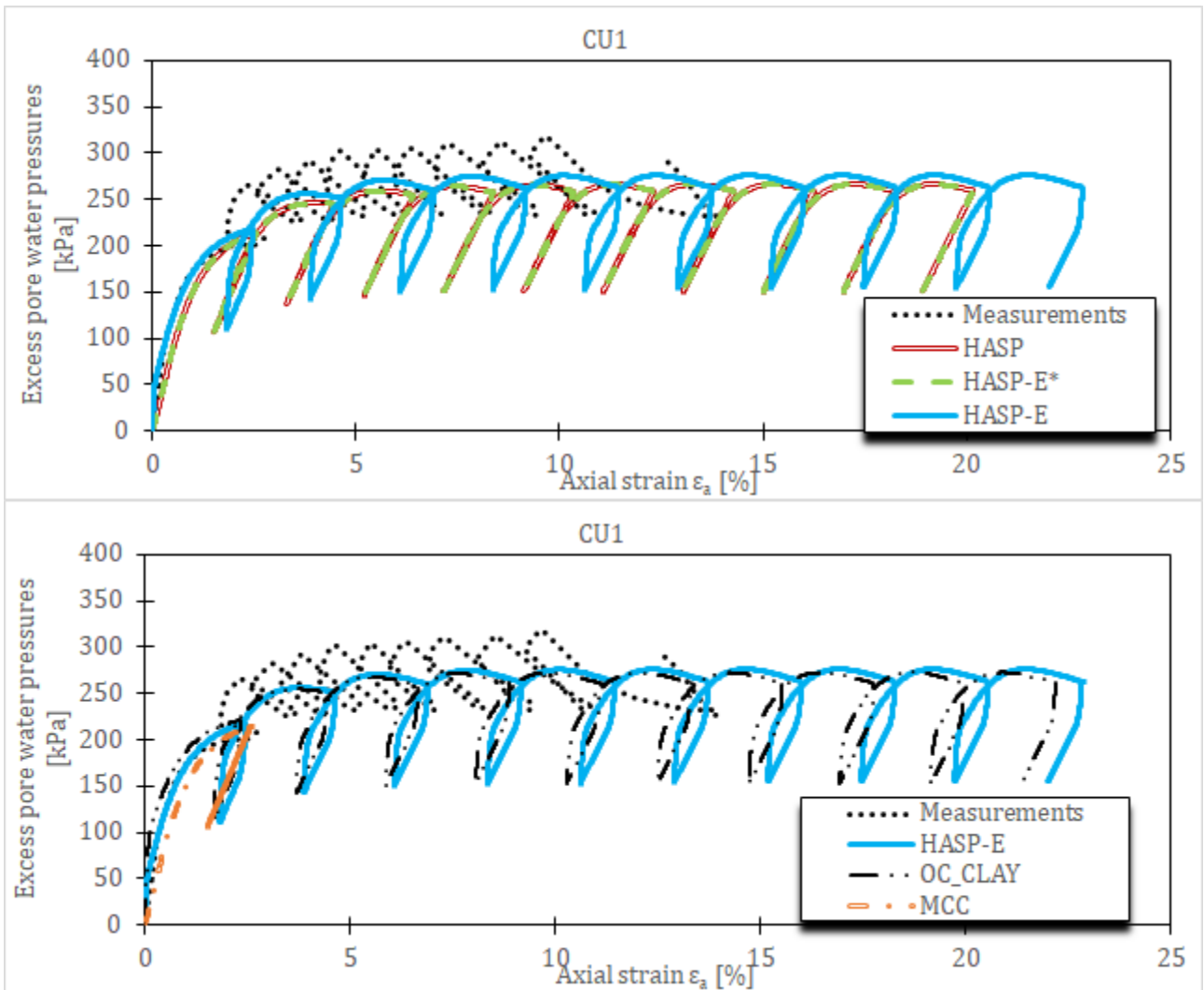
OC_CLAY модел са активним SSO моделом и HASP-E модел са активном “Brick” компонентом могу да опишу понашање глине при цикличном оптерећењу у недренираним условима. Оба модела предвиђају хистерезисно понашање током једног циклуса оптерећења и растерећења. Мале разлике у предвиђањима понашања глине између HASP-E и OC_CLAY модела могу да се објасне разликом у облику S-кривих које користе ова два модела. За оба опита HASP-E и OC_CLAY модел за иницијални сет материјалних параметара предвиђају веће аксијалне деформације него што су експериментално забележене. Величина хистерезиса коју оба модела предвиђају је слична експерименталним подацима. Оба модела предвиђају нешто мање додатне порне притиске него што су измерени.

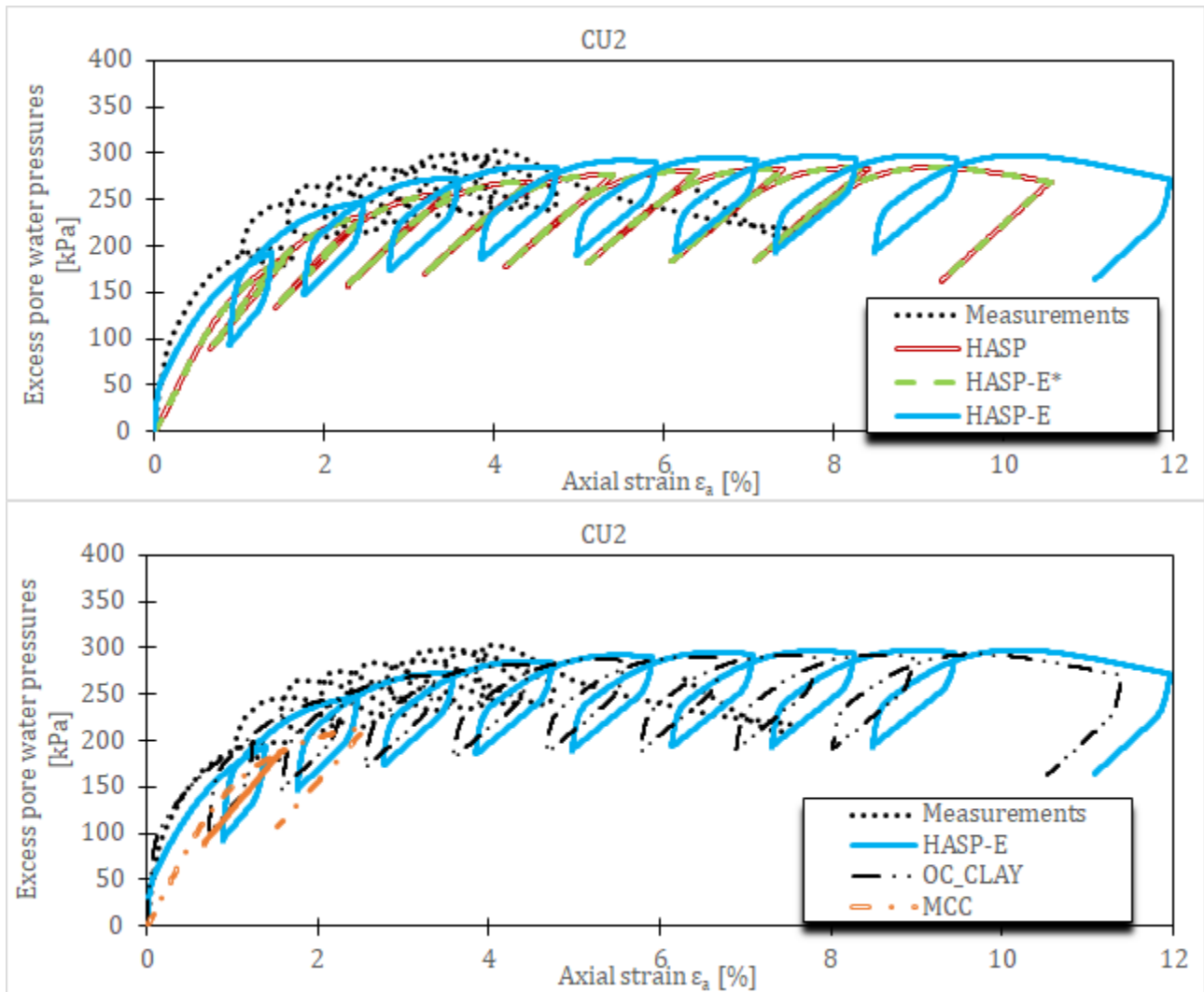
У наставку су дати резултати параметарске анализе.





Слика 5-13 Напонско-деформацијске криве за CU цикличне триаксијалне опите на Newfield глини за различите конститутивне моделе и иницијални сет материјалних параметара





Слика 5-14 Додатни порни притисци за CU цикличне триаксијалне опите на Newfield глини за различите конститутивне modele и иницијални сет материјалних параметара

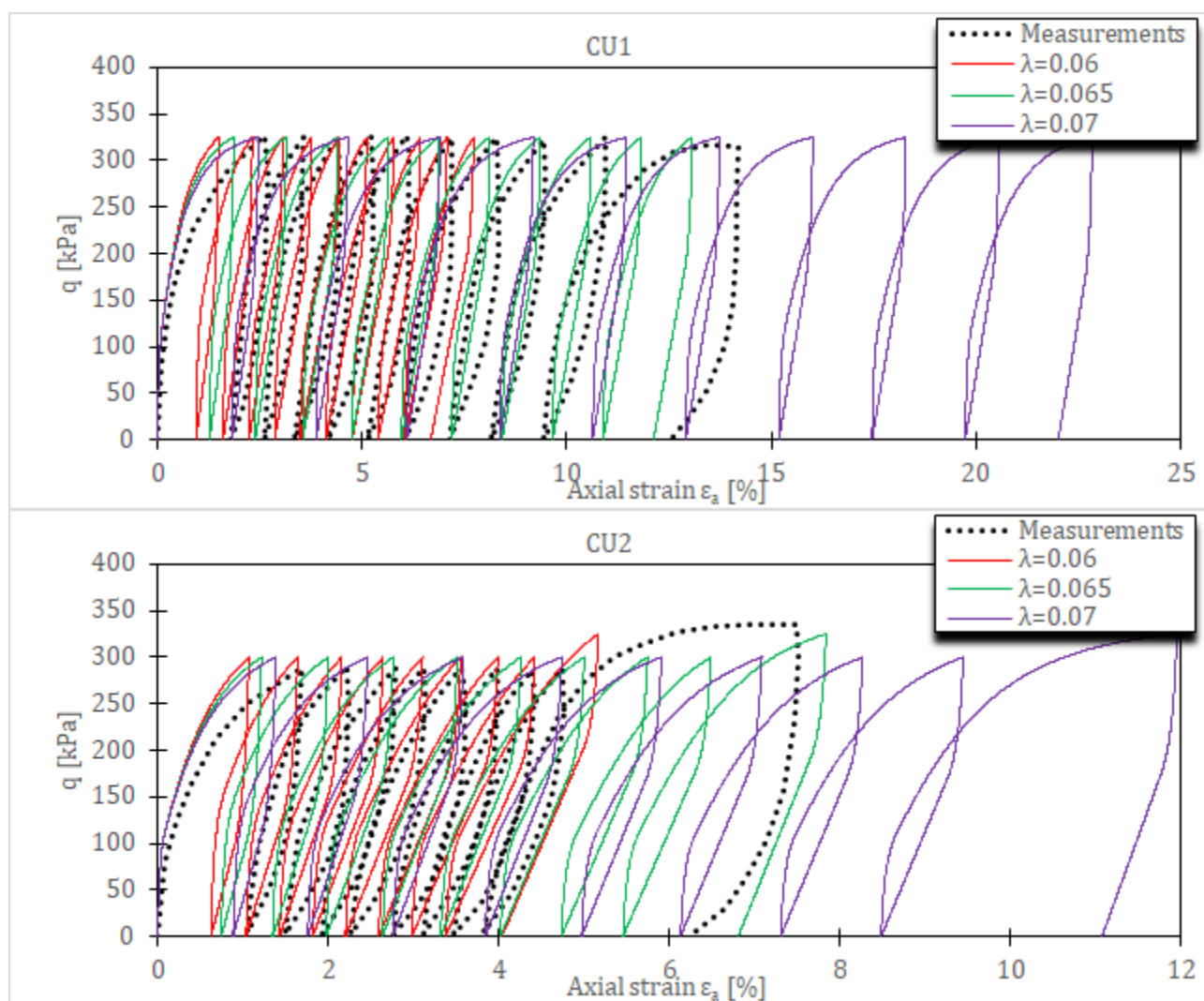
Утицај величине λ је дат на **Слика 5-15** и **Слика 5-16**. Пошто се за иницијалну вредност параметра λ добијају веће аксијалне деформације него што су експериментално забележене, у параметарској анализи је коришћена вредност параметра λ мања од вредности дате у **Табела 5-5**. Из приказаних напонско-деформацијских кривих може се уочити да мала промена параметра λ значајно утиче на величину аксијалних деформација које се добијају. Величина хистерезиса је иста без обзира на вредност параметра λ . Са смањивањем вредности λ повећавају се вредности додатних порних притисака у сваком циклусу.

Утицај величине k је дат на **Слика 5-17** и **Слика 5-18**. Са вредности k већом од иницијалне вредности дате у **Табела 5-5** добијају се предвиђања понашања глине која су ближа експериментално забележеним. Са порастом вредности k добијају се мање аксијалне деформације и већи додатни порни притисци.

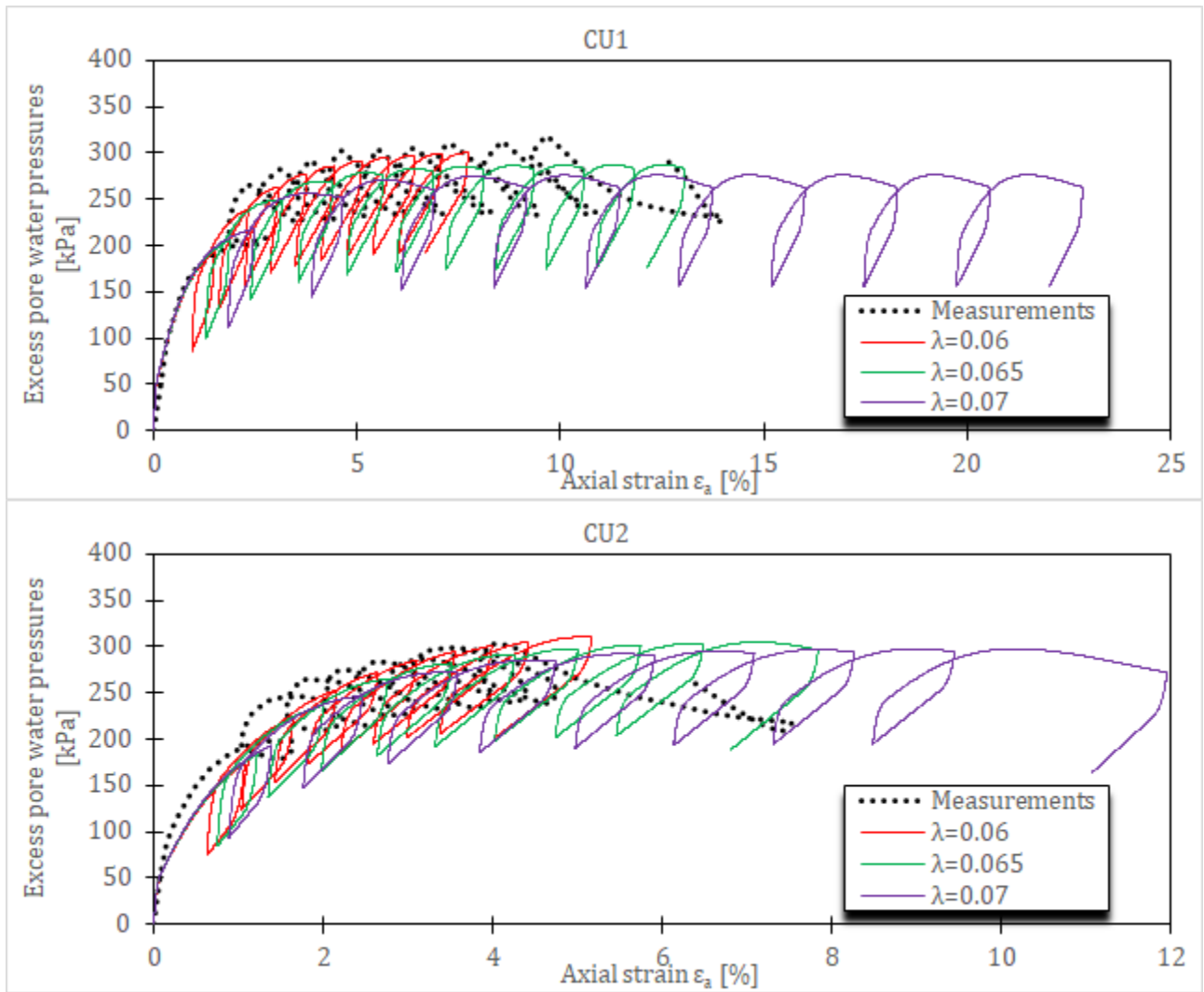
Утицај величине G_0^{ref} је дат на **Слика 5-19** и **Слика 5-20**. Са вредностима G_0^{ref} мањим од иницијалне вредности дате у **Табела 5-5** добијају се мање аксијалне деформације и

мања величина хистерезиса него што је експериментално забележено. Са смањивањем вредности G_0^{ref} смањују се и максималне вредности додатних порних притисака. Са смањивањем вредности G_0^{ref} , блажи је нагиб напонско-деформацијских кривих приликом растеређивања и поновног оптеређивања.

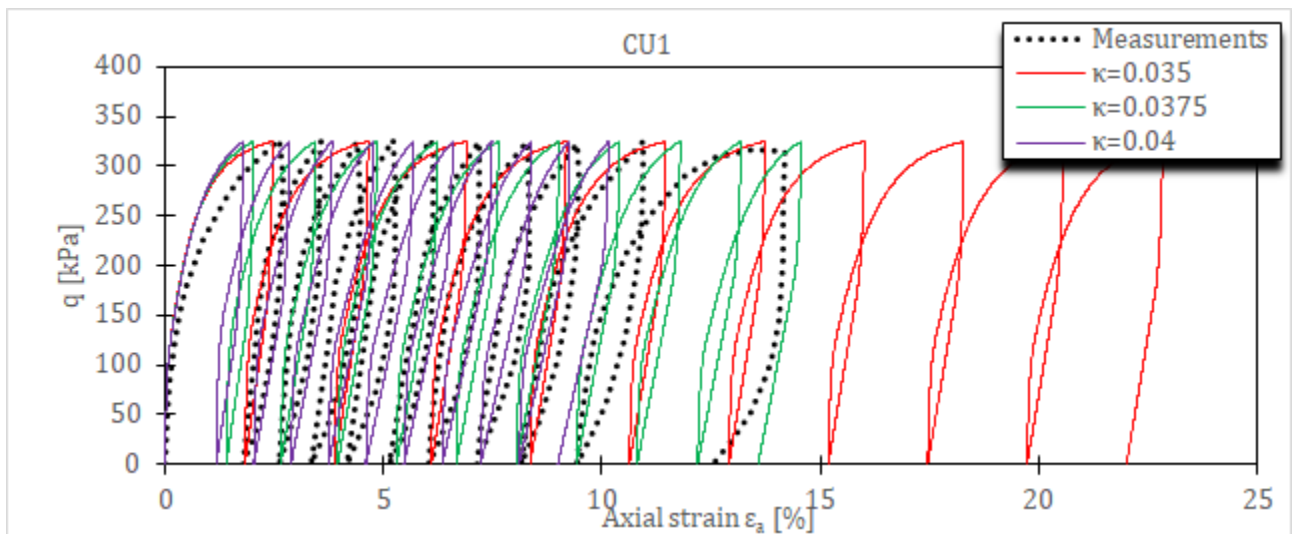
Утицај величине $\gamma_{0,7}$ је дат на **Слика 5-21** и **Слика 5-22**. Са повећањем вредности $\gamma_{0,7}$ смањује се величина хистерезиса, напонско-деформацијске криве су стрмије приликом растеређивања и поновног оптеређивања, аксијалне деформације су веће него што су измерене и повећавају се максималне вредности додатних порних притисака. За веће вредности $\gamma_{0,7}$, деградација крутости тла је спорија и разлика између λ и κ (израз 3-96) током једног циклуса дуже остаје већа него за мање вредности $\gamma_{0,7}$, што као последицу има веће аксијалне деформације за веће вредности $\gamma_{0,7}$.

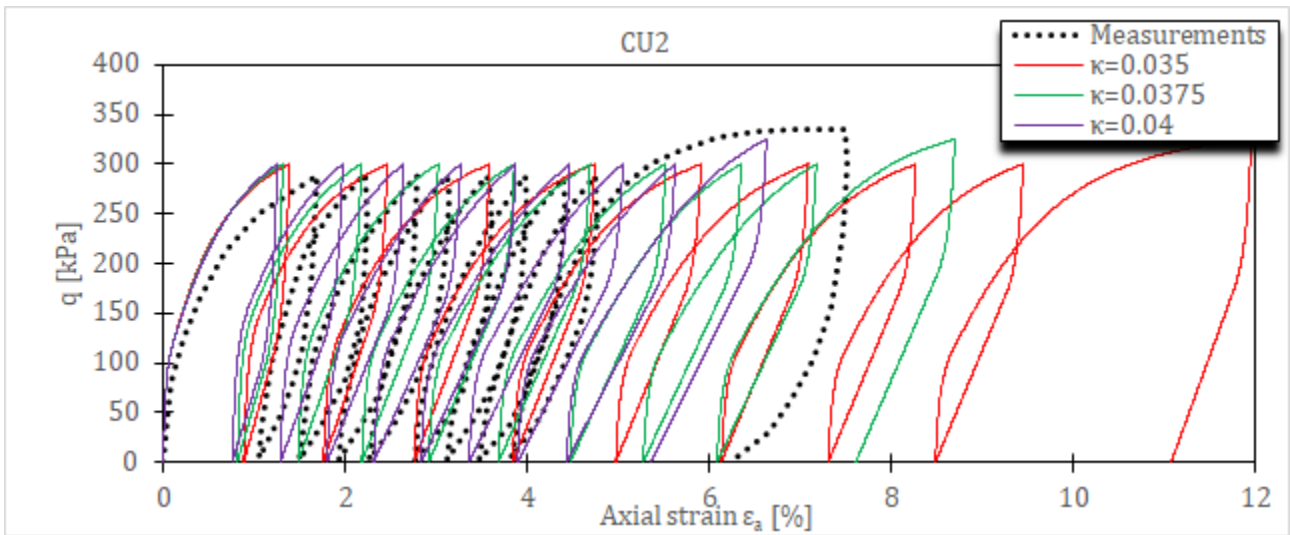


Слика 5-15 Напонско-деформацијске криве за CU цикличне триаксијалне опите на Newfield глини за HASP-E модел и различите вредности параметра λ

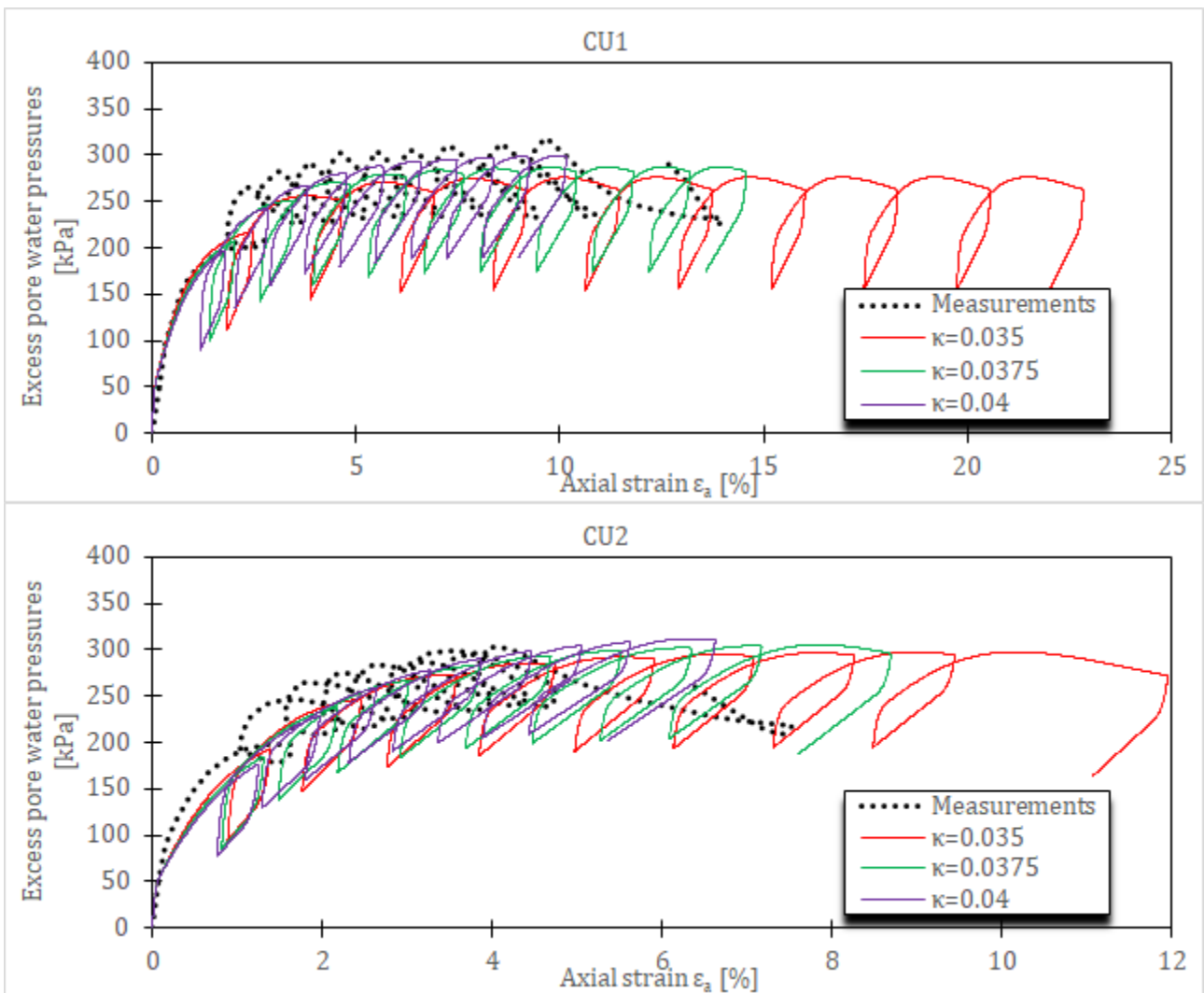


Слика 5-16 Додатни порни притисци за CU цикличне триаксијалне опите на Newfield глини за HASP-E модел и различите вредности параметра λ

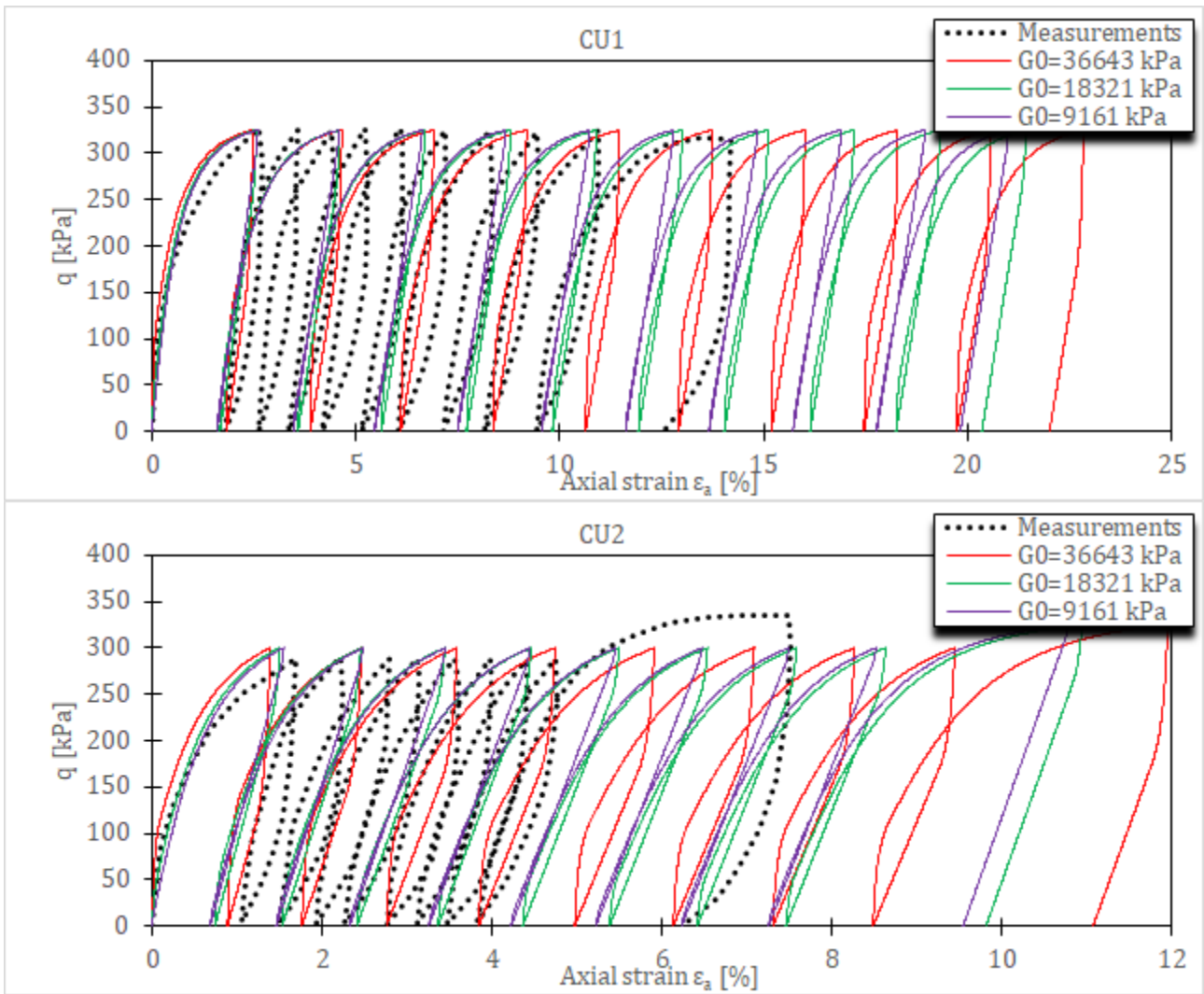




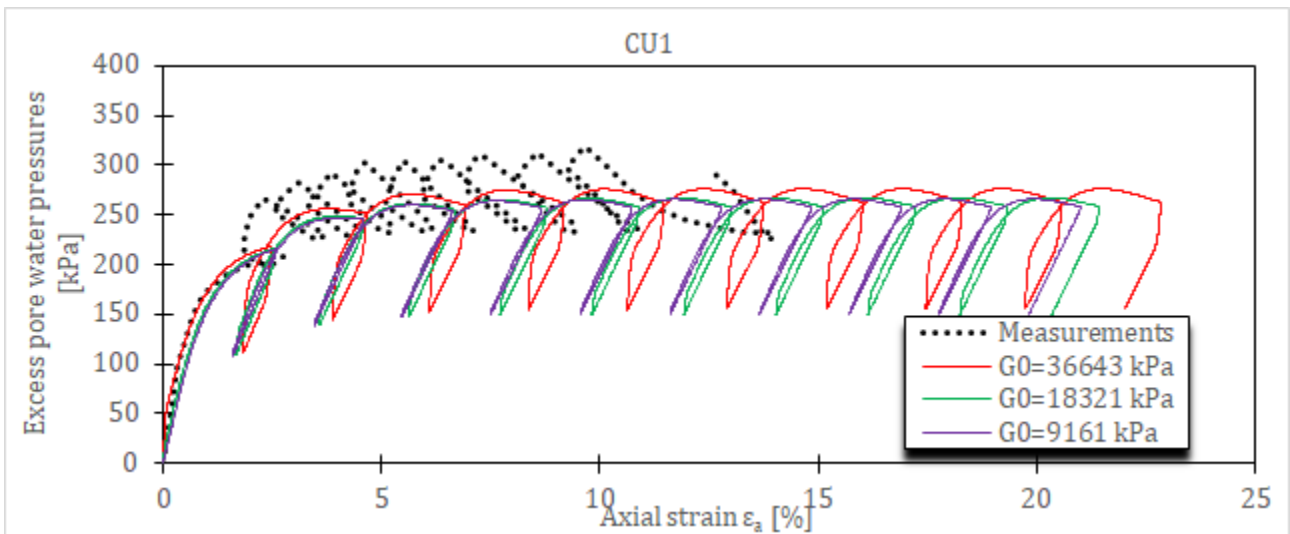
Слика 5-17 Напонско-деформацијске криве за CU цикличне триаксијалне опите на Newfield глини за HASP-E модел и различите вредности параметра κ

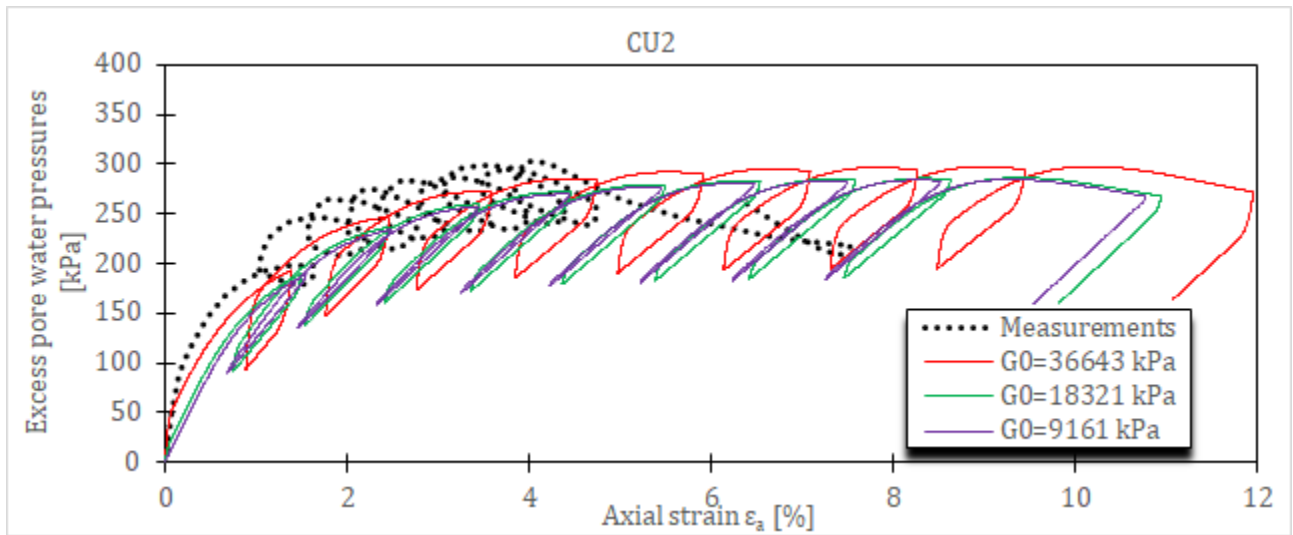


Слика 5-18 Додатни порни притисци за CU цикличне триаксијалне опите на Newfield глини за HASP-E модел и различите вредности параметра κ

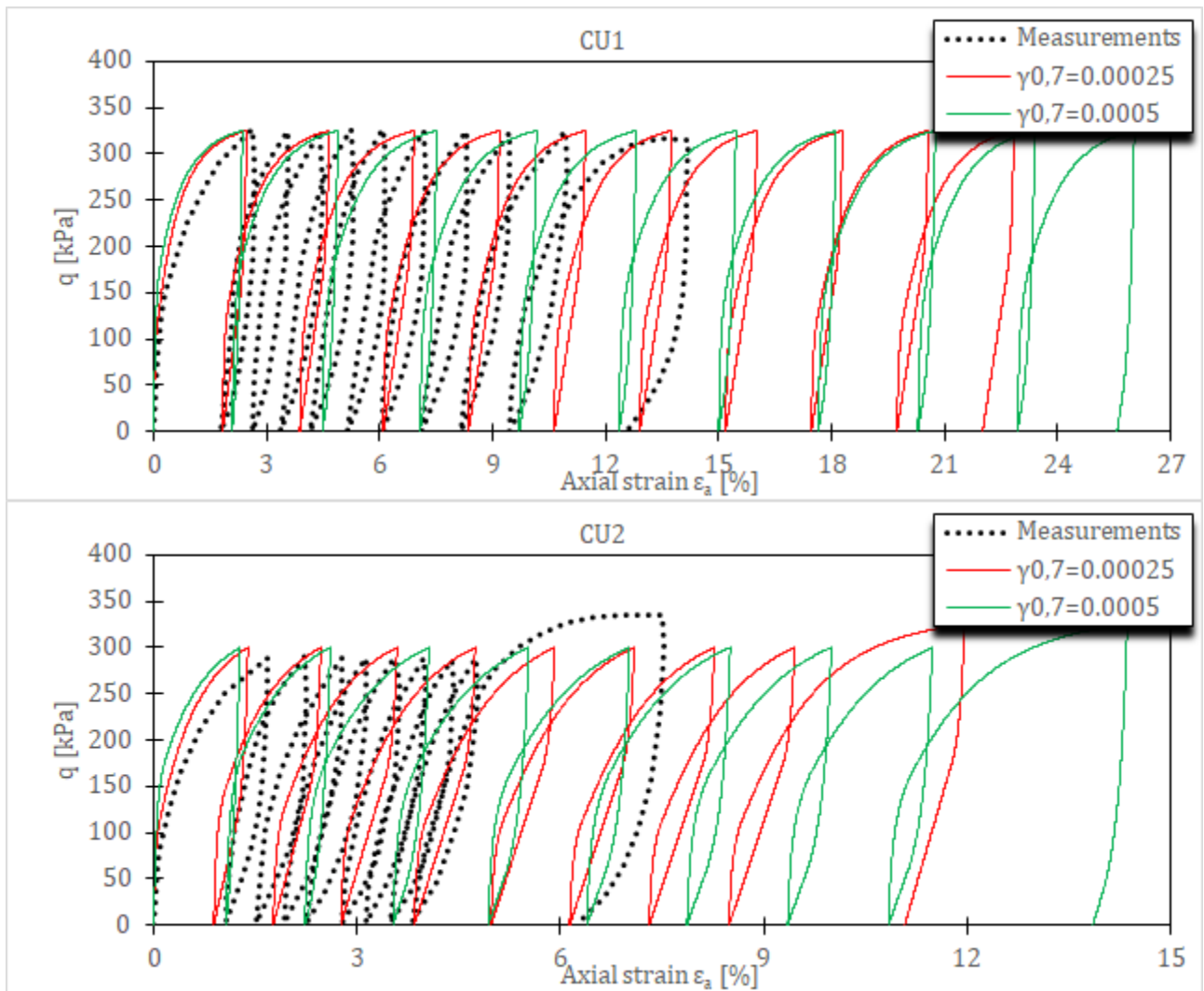


Слика 5-19 Напонско-деформацијске криве за CU цикличне триаксијалне опите на Newfield глини за HASP-E модел и различите вредности параметра G_0^{ref}

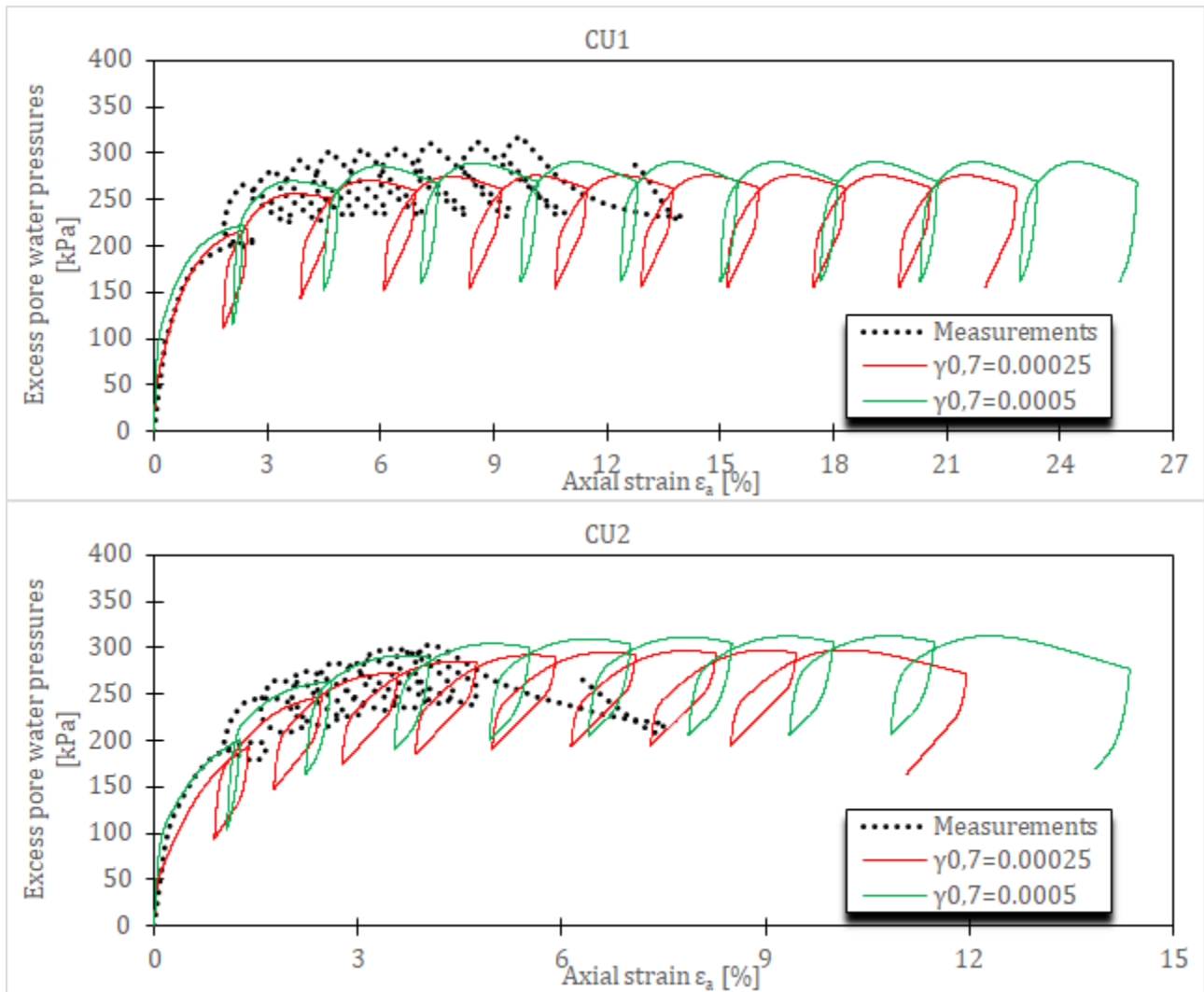




Слика 5-20 Додатни порни притисци за CU цикличне триаксијалне опите на Newfield глини за HASP-E модел и различите вредности параметра G_0^{ref}



Слика 5-21 Напонско-деформацијске криве за CU цикличне триаксијалне опите на Newfield глини за HASP-E модел и различите вредности параметра $\gamma_{0,7}$



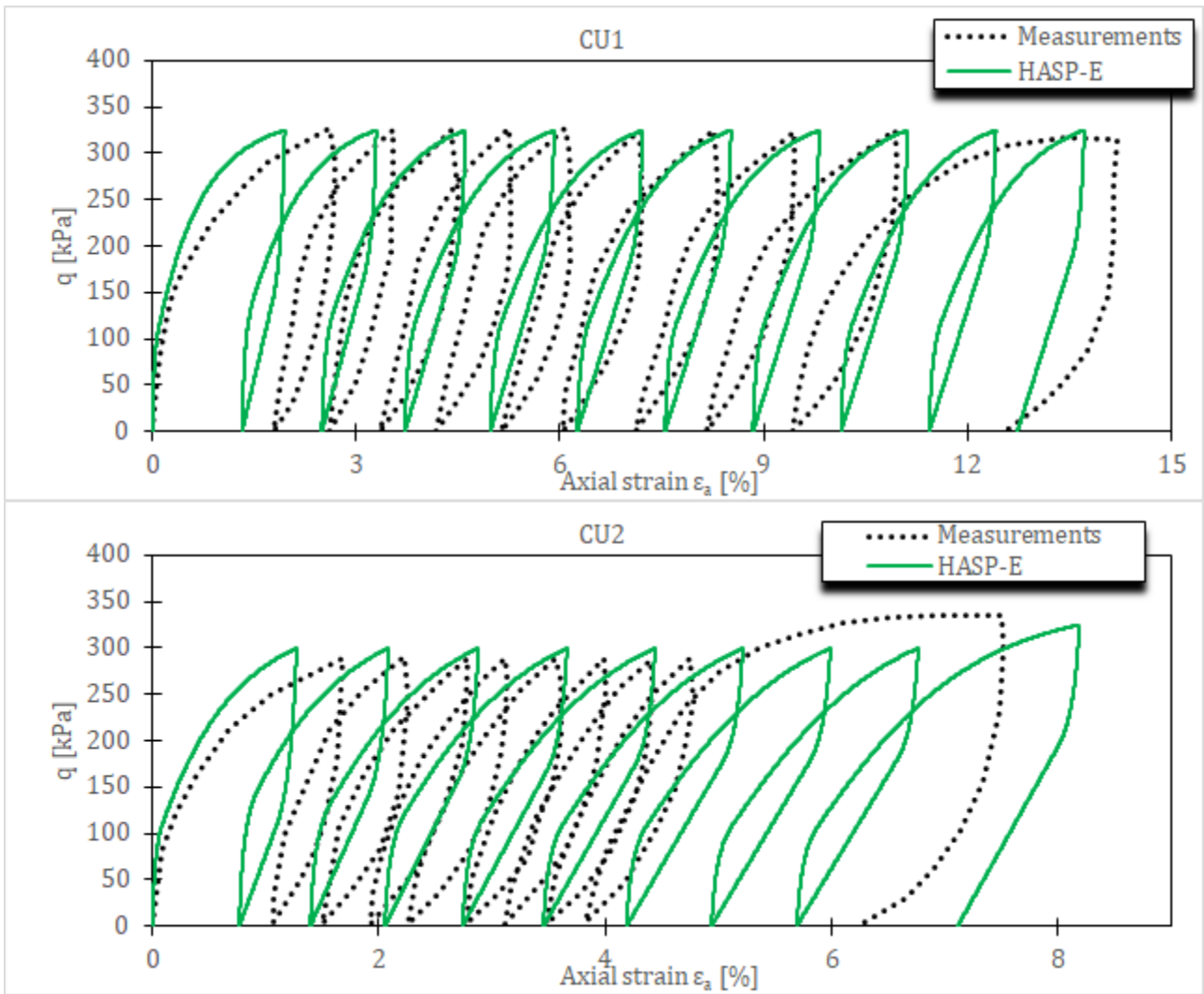
Слика 5-22 Додатни порни притисци за CU цикличне триаксијалне опите на Newfield глини за HASP-E модел и различите вредности параметра $\gamma_{0,7}$

На основу резултата параметарске анализе, са циљем да се резултати нумеричких симулација за HASP-E модел што више поклапају се мерењима, извршена је калибрација вредности параметара λ , κ и G_0^{ref} HASP-E модела. Усвојени сет материјалних параметара дат је у **Табела 5-6**. Иницијални услови (вредност иницијалног коефицијента порозности) дати у **Табела 5-4** су остали непромењени.

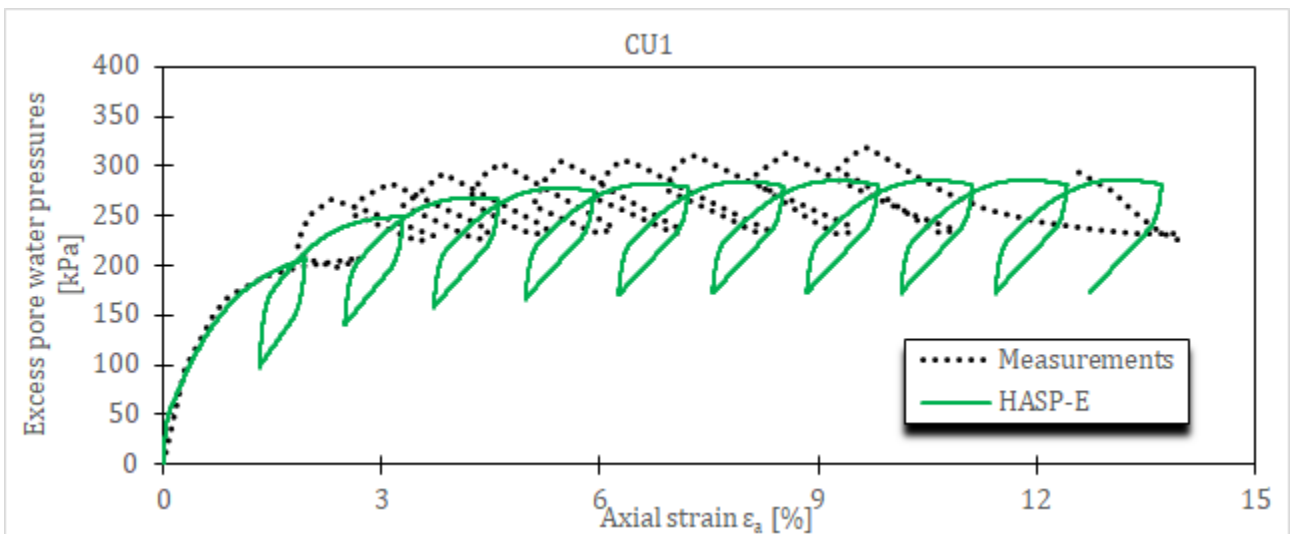
Резултати нумеричких симулација за HASP-E модел и усвојени сет материјалних параметара дати су на **Слика 5-23** и **Слика 5-24**.

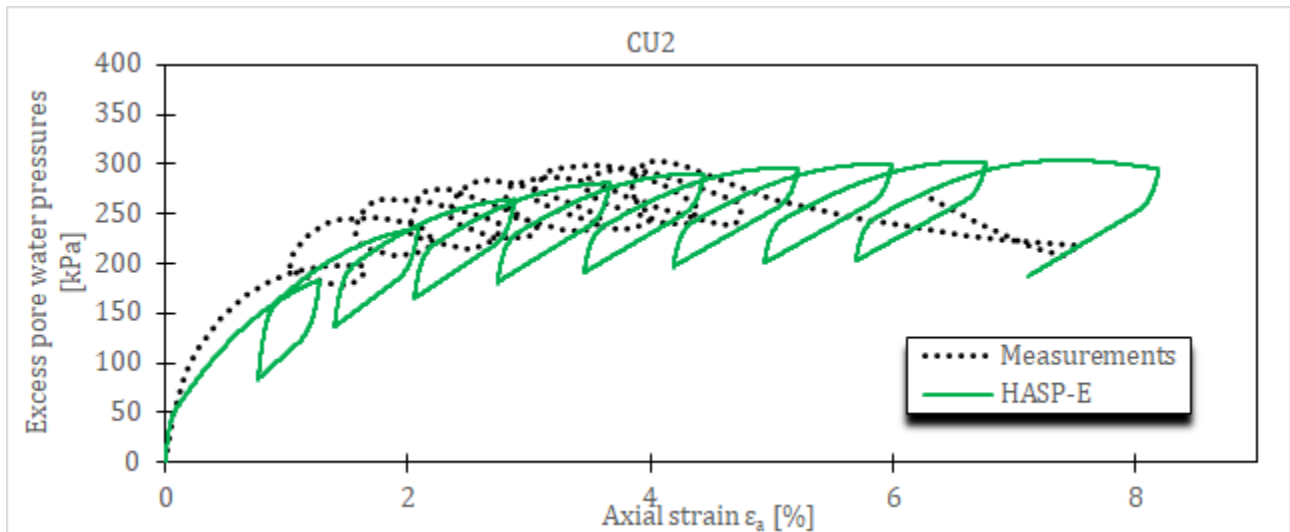
Табела 5-6 Усвојени сет параметара HASP-E модела за симулацију два CU циклична триаксијална опита на Newfield глини

Модел	Параметри						
	λ [-]	κ [-]	M_c [-]	M_e [-]	μ [-]	G_0^{ref} [kPa]	$\gamma_{0,7}$ [-]
HASP-E	0.0675	0.03625	1.2	0.8	0.2	35379	0.00025



Слика 5-23 Напонско-деформацијске криве за CU цикличне триаксијалне опите на Newfield глини за HASP-E модел и усвојени сет материјалних параметара





Слика 5-24 Додатни порни притисци за CU цикличне триаксијалне опите на Newfield глине за HASP-E модел и усвојени сет материјалних параметара

На основу приказаних резултата нумеричке симулације цикличних триаксијалних опита у овом и претходном потпоглављу, могу се извести следећи општи закључци о могућностима HASP-E модела:

- HASP-E модел може да се користи за немонотона оптерећења, као што је растерећење тла и циклично оптерећење,

- HASP-E модел може да опише хистерезисно понашање при растерећењу и поновном оптерећењу,

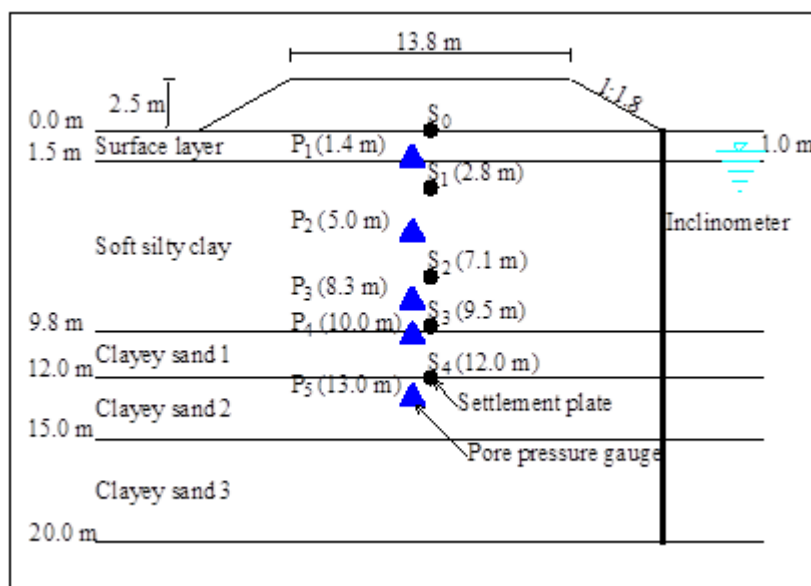
- “Brick” компонента HASP-E модела може на адекватан начин да опише величину хистерезиса,

- HASP-E модел је најосетљивији на промену параметара деформабилности λ и κ , у дренираним цикличним опитима, HASP-E модел квалитативно и уз адекватну калибрацију материјалних параметара и квантитативно добро описује понашање тла,

- у недренираним цикличним опитима, HASP-E модел квалитативно и уз калибрацију материјалних параметара и квантитативно добро описује величину деформација и додатних порних притисака.

5.4 Предвиђање понашања слоја глине услед изградње насипа – Saga Highway Embankment

У овом потпоглављу приказани су резултати нумеричке симулације понашања глине у PLAXIS-у 2D услед изградње насипа на примеру преузетом из (Chai, et al., 2013) – Saga Highway Embankment. Насип за потребе изградње аутопута је изграђен на слоју меке Ariake глине у близини града Saga у Јапану.



Слика 5-25 Попречни пресек кроз насип са положајем мерних инструмената – адаптирано из (Chai, et al, 2013)

Како би се одредили материјални параметри тла потребни за пројектовање, коришћени су мерни инструменти уграђени у насип и темељно тло. Мерена су померања темељног тла (вертикална и хоризонтална) и промена порних притисака у тлу током изградње насипа. Мерења су настављена још три године након завршетка изградње насипа. Спроведена су веома детаљна теренска и лабораторијска испитивања на локацији насипа.

Темељно тло је изграђено из више различитих слојева. На површини терена налази се слој јако преконсолидоване глине у дебљини од 1.5 m. Испод површинског слоја налази се слој меке прашинасте глине дебљине око 8 m. Глиновити пескови се налазе испод меке прашинасте глине. Ниво подземне воде се налази на дубини од 1 m испод површине терена. Усвојен профил терена са положајем мерних инструмената дат је на **Слика 5-25**.

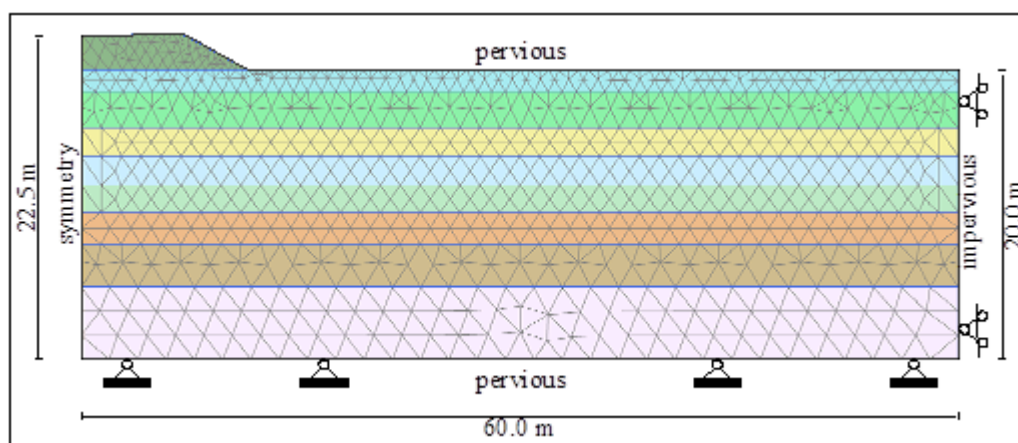
Висина насипа је 2.5 m и изграђен је за 50 дана. Насип је у основи дугачак 46.8 m и широк 21.8 m. Нагиб косина насипа је 1: 1.8 (вертикално:хоризонтално). У круни, насип је дугачак 37.8 m и широк 13.8 m. Насип је изграђен од распаднутог гранита.

5.4.1 Нумерички модел

За нумеричке симулације коришћено је раванско стање деформација. Коришћена мрежа коначних елемената са граничним условима је приказана на **Слика 5-26**. Због симетрије, само једна половина насипа је разматрана у нумеричком моделу. Како би се што више смањио утицај граничних услова на резултате нумеричке симулације, у нумеричком моделу је разматран домен тла ширине 60 m и дебљине 22.5 m. Коришћено је укупно 1620 коначних елемената. Коришћена је спрегнута напонско-деформацијска и филтрациона анализа (Fully coupled flow-deformation analysis).

Хоризонтална померања на левој и десној ивици нумеричког модела су спречена, док су вертикална слободна. На доњој ивици нумеричког модела, вертикална и хоризонтална померања су спречена, а на горњој ивици, вертикална и хоризонтална померања су

слободна. Од хидрауличких граничних услова, дренарање воде је дозвољено на горњој (површина терена) и доњој ивици (прашинасти пескови) нумеричког модела. Дренарање је спречено на левој и десној ивици нумеричког модела. Хидраулички гранични услови су преузети из (Chai, et al., 2013).



Слика 5-26 Мрежа коначних елемената и гранични услови за Saga Highway Embankment

Прва фаза у нумеричкој симулацији је иницијална фаза у којој се генеришу почетни напони и порни притисци у тлу. Друга фаза је изградња насипа до пуне висине у трајању од 50 дана. Након друге фазе разматрано је још четири фазе од којих свака траје годину дана како би се одредило консолидационо понашање глине након изградње насипа. Укупан број дана у свих пет фаза је 1511. Минимални број инкремената у свакој од фаза прорачуна је 1000.

Слој глине изнад нивоа подземне воде је водом засићен и порни притисци су негативни (напони сукције). Онда су ефективни напони на површини терена већи од нуле и на тај се обезбеђује да површински слој глине има смичућу чврстоћу.

5.4.2 Параметри конститутивних модела

У спроведеним нумеричким симулацијама, за насип и слојеве песка коришћен је Mohr-Coulomb-ов (MC) конститутивни модел, а за слојеве глине коришћени су HASP, HASP-E или MCC модел. Параметри свих конститутивних модела су преузети без измена из (Chai, et al., 2013). Параметри конститутивних модела су дати у Табела 5-7. HASP и MCC модели користе вредност параметра M_c за параметар M . Вредност параметра M_e није била дата у (Chai, et al., 2013), па је усвојено да је $\phi'_E = \phi'_C$ и M_e је срачунато према изразу (3-25). Код HASP-E модела, "Brick" компонента није била активна.

Иницијални услови за нумеричке симулације су дати у Табела 5-8. Иницијална вредност коефицијента порозности и коефицијент притиска тла у стању мировања за нормално консолидовано тло су преузети из (Chai, et al., 2013). Преконсолидација слојева глине је задата преко параметра POP . Вредност параметра POP за сваки слој глине је одређена преко израза (4-296) тако да се добије вредност OCR које су дате у (Chai, et al., 2013). Промена OCR по дубини за усвојене вредности POP и вредности OCR из (Chai, et al., 2013) приказани су на Слика 5-27. Коефицијент притиска тла у стању мировања за преконсолидоване слојеве глине је одређен преко израза (4-295). Пошто

се вредност OCR мења са дубином, за један слој глине је усвојена просечна вредност K_0^{OC} срачуната према изразу (4-295).

Табела 5-7 Параметри конститутивних модела за нумеричку симулацију понашања глине услед изградње насипа – *Saga Highway Embankment*

Дубина [m]	Слој тла	Параметри				
		λ [-]	κ [-]	M_c [-]	M_e [-]	μ [-]
0.0-1.5	Површински слој глине	0.25	0.025	1.6	1.044	0.15
1.5-4.0	Мека прашинаста глина	1.07	0.107	1.6	1.044	0.15
4.0-6.0		1.19	0.119			
6.0-8.0		0.84	0.084			
8.0-9.8		0.66	0.066			
9.8-12.0	Глиновити пескови	МС параметри: $E' = 7500 \text{ kPa}$, $c' = 20 \text{ kPa}$, $\phi' = 35^\circ$, $\psi = 5^\circ$, $\mu = 0.3$				
12.0-15.0		МС параметри: $E' = 20000 \text{ kPa}$, $c' = 20 \text{ kPa}$, $\phi' = 35^\circ$, $\psi = 5^\circ$, $\mu = 0.3$				
15.0-20.0		МС параметри: $E' = 37500 \text{ kPa}$, $c' = 20 \text{ kPa}$, $\phi' = 35^\circ$, $\psi = 5^\circ$, $\mu = 0.3$				
	Насип	МС параметри: $E' = 5000 \text{ kPa}$, $c' = 20 \text{ kPa}$, $\phi' = 35^\circ$, $\psi = 0^\circ$, $\mu = 0.3$				

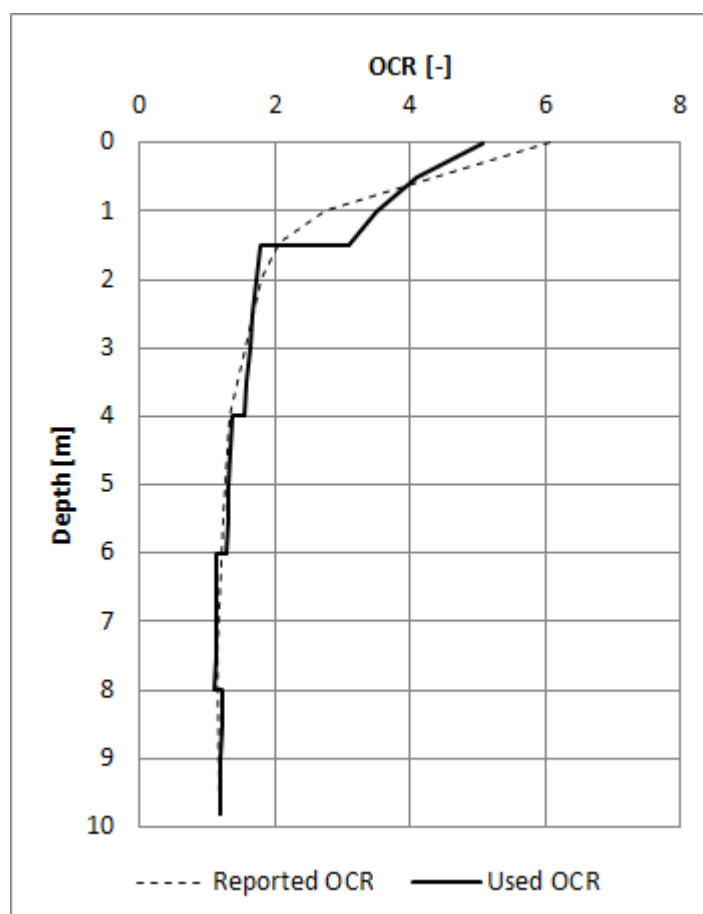
Табела 5-8 Иницијални услови за нумеричку симулацију понашања глине услед изградње насипа – *Saga Highway Embankment*

Дубина [m]	Слој тла	Параметри			
		e_0 [-]	POP [kPa]	K_0^{NC} [-]	K_0^{OC} [-]
0.0-1.5	Површински слој глине	1.5	40	0.51	0.9837
1.5-4.0	Мека прашинаста глина	3.71	15	0.51	0.6428
4.0-6.0		2.88	10		0.5745
6.0-8.0		2.72	5		0.5306
8.0-9.8		1.91	10		0.5485
9.8-12.0	Глиновити пескови	0.80	0	0.43	
12.0-15.0		0.80	0		
15.0-20.0		0.70	0		
	Насип	0.80	0	0.43	

Остали параметри који су потребни за формирање нумеричког модела су дати у **Табела 5-9**. Вредности запреминске тежине и водопропустљивости тла у вертикалном (k_v) и хоризонталном (k_h) правцу су преузети из (Chai, et al., 2013). Природна глиновита тла су анизотропна у погледу водопропустљивости. Водопропустљивост у хоризонталном правцу је већа него у вертикалном правцу. Према (Chai & Miura, 1999), усвојен је однос k_h/k_v од 1.5 за све слојеве глине. Водопропустљивост тла се мења у зависности од промене коефицијента порозности према изразу:

$$k = k_0 \cdot 10^{-(e_0 - e)/C_k}$$

5-1



Слика 5-27 Промена степена преконсолидације са порастом дубине у зависности од вредности параметра POP за Saga Highway Embankment

Табела 5-9 Остали параметри нумеричког модела потребни за нумеричку симулацију понашања глине услед изградње насипа – Saga Highway Embankment

Дубина [m]	Слој тла	Параметри			
		Запреминска тежина [kN/m ³]	k_v [10 ⁻⁴ $\frac{m}{day}$]	k_h [10 ⁻⁴ $\frac{m}{day}$]	c_k [-]
0.0-1.5	Површински слој глине	16.0	6	9.1	0.6
1.5-4.0	Мека прашинаста глина	13.1	10.4	15.6	1.48
4.0-6.0		19.9	17.3	25.9	1.15
6.0-8.0		14.1	16.5	24.7	1.09
8.0-9.8		15.7	7	10.6	0.76
9.8-12.0	Глиновити пескови	18.0	2500	2500	0.32
12.0-15.0		18.0	2500	2500	0.32
15.0-20.0		19.0	2500	2500	0.32
Насип		19.0	Веома велика		-

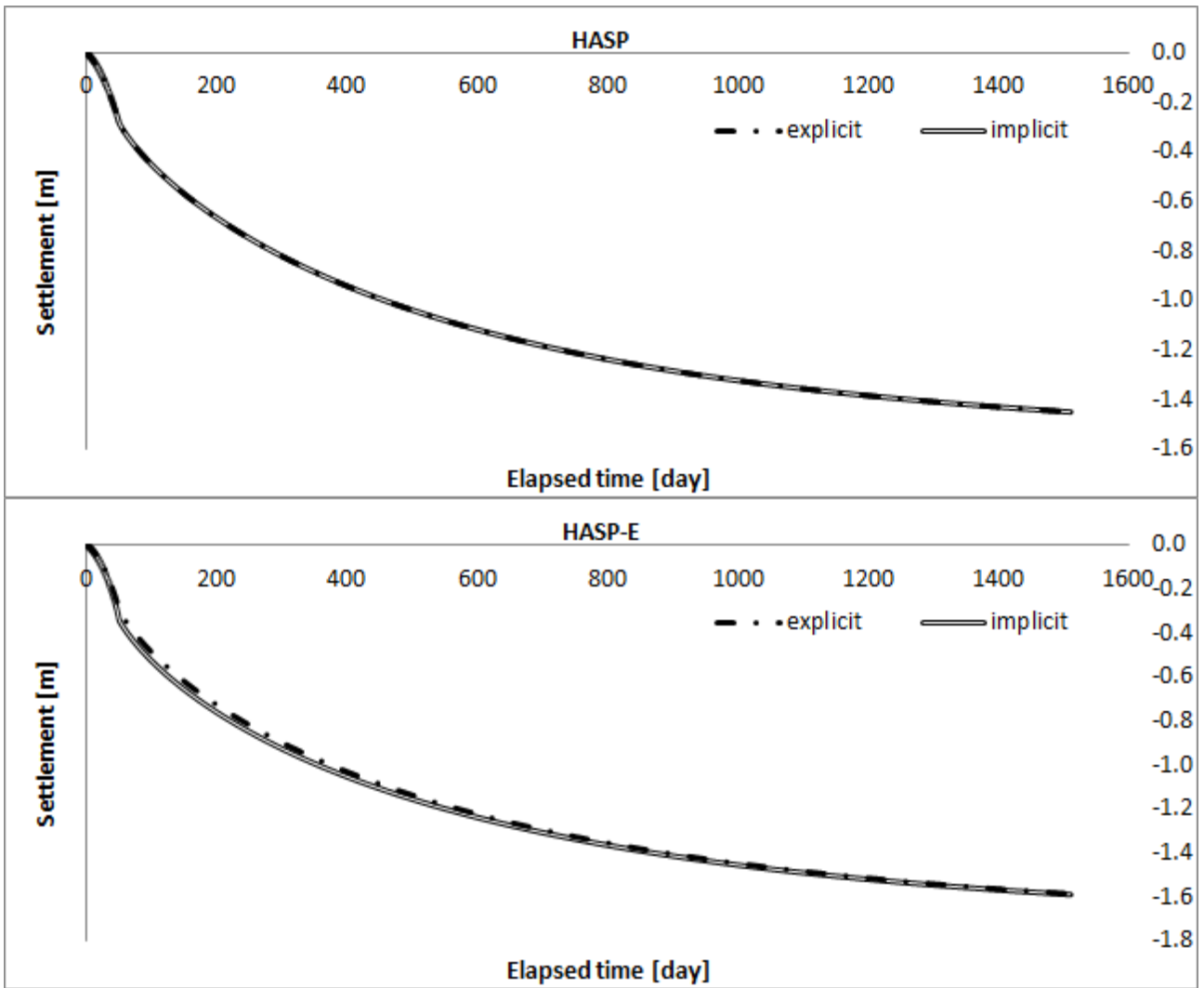
У претходном изразу, k_0 је иницијална вредност водопропустљивости тла (у вертикалном или хоризонталном правцу) и дата је у **Табела 5-9**. Са e_0 је означена иницијална вредност коефицијента порозности, са e је означена тренутна вредност коефицијента порозности, а C_k је константа. Према препорукама у (Tavenas, et al., 1986), за C_k је усвојена вредност од $0.4e_0$. С обзиром да је крутост слојева песка знатно већа него слојева глине, коефицијент порозности слојева песка ће се веома мало променити током симулације у односу на иницијалну вредност, па ће зато водопропустљивост слојева песка остати непромењена током прорачуна.

5.4.3 Резултати и дискусија

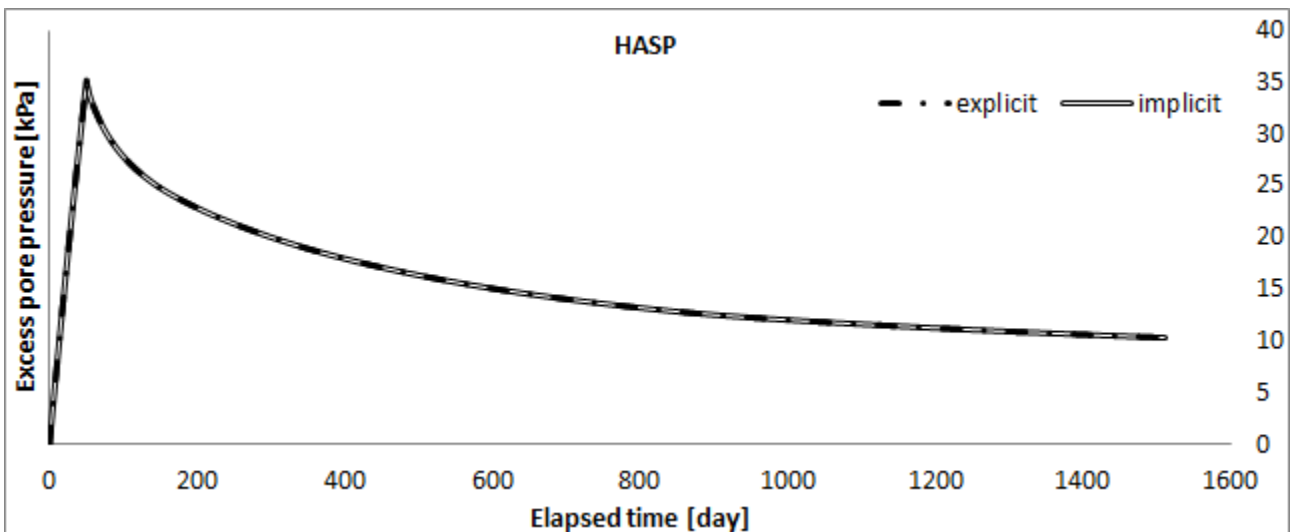
Како би се проверило да ли на добијене резултате утиче метода интеграције конститутивних релација HASP и HASP-E модела, спроведене су нумеричке симулације користећи имплицитни алгоритам водећег параметра и експлицитни алгоритам модификованог Euler-овог поступка са аутоматском субинкрементацијом и контролом грешке са $STOL = 10^{-4}$. Број и величина коришћених инкремената је иста за обе методе нумеричке интеграције. Резултати ове анализе дати су на **Слика 5-28** и **Слика 5-29**. Из приложених дијаграма може се уочити да избор методе нумеричке интеграције не утиче на добијене резултате. У наставку су за HASP и HASP-E резултати приказани само за модификовани Euler-ов поступак са аутоматском субинкрементацијом и контролом грешке.

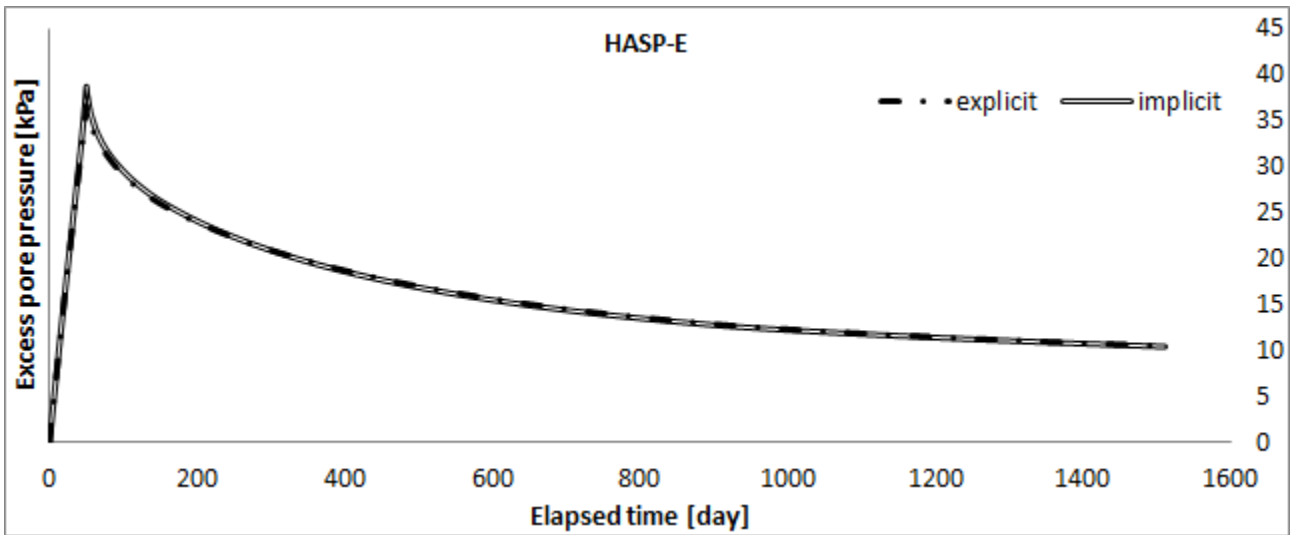
Предвиђања HASP и HASP-E модела упоређена су са резултатима предвиђања MCC модела и са осмотреним подацима. Приказани су резултати за слегање тачака S_0 (површина терена) и S_1 (2.8 m испод површине терена) на **Слика 5-30**. Резултати за слегања других тачака нису анализирани због проблема са мерним инструментима (Chai, et al., 2013). Из приказаних резултата може се уочити да се за све конститутивне моделе, слегања све мање повећавају са временом. Највећи прираштај слегања је током и непосредно након изградње насипа, а отприлике три-четири године након завршетка изградње, крива слегања се полако заравњује, односно већи део примарне консолидације се завршио. Према резултатима мерења, до завршетка примарне консолидације долази отприлике две године након завршетка изградње насипа. Према резултатима нумеричких симулација, примарна консолидација се још увек није завршила четири године након завршетка изградње насипа. Разлог зашто се у нумеричким симулацијама добија да примарна консолидација траје дуже него што је измерено може бити у томе да је водопропустљивост тла већа од вредности које су коришћене у нумеричкој анализи.

Током изградње насипа, сви конститутивни модели предвиђају сличне вредности, слегања које су блиске измереним. Након завршетка изградње насипа, MCC модел предвиђа слегања која су мања од измерених, поготово у прве две године након завршетка изградње насипа. Оба HASP модела предвиђају слегања која су ближа измереним вредностима. Деформације које предвиђају HASP модели су генерално већа од предвиђања MCC модела, што је и очекивано, с обзиром да се код HASP модела еласто-пластичне деформације јављају одмах са почетком оптерећивања, док се код MCC модела јављају само еластичне деформације за напонска стања унутар површи течења (граничне површи за HASP моделе). Слегања HASP-E модела су већа него слегања која предвиђа HASP модел, што је последица тога што HASP-E узима у обзир утицај треће напонске инваријанте и реалнији услов лома него HASP модел.

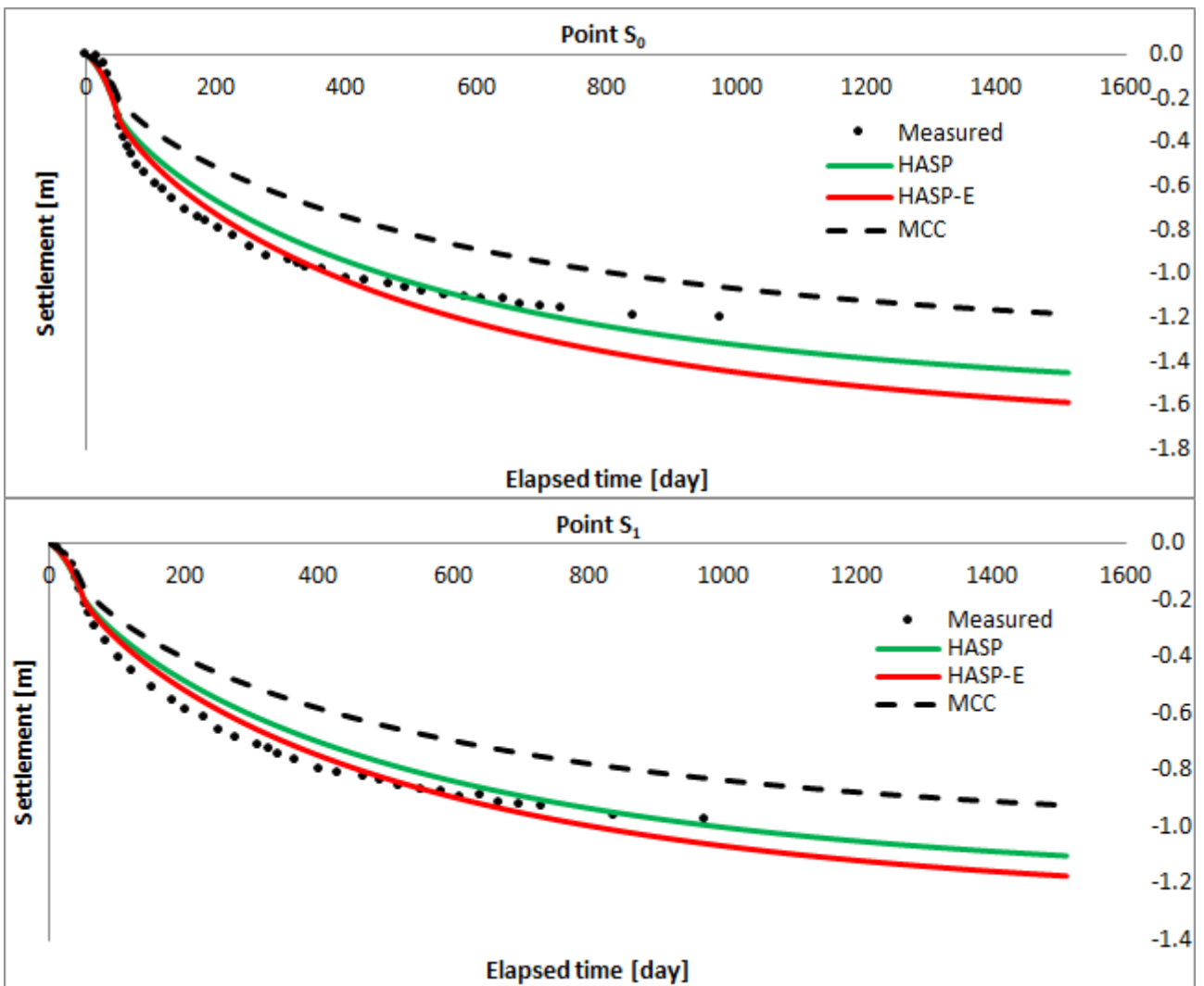


Слика 5-28 Слегање тла у тачки S_0 кроз време за различите методе нумеричке интеграције конститутивних релација



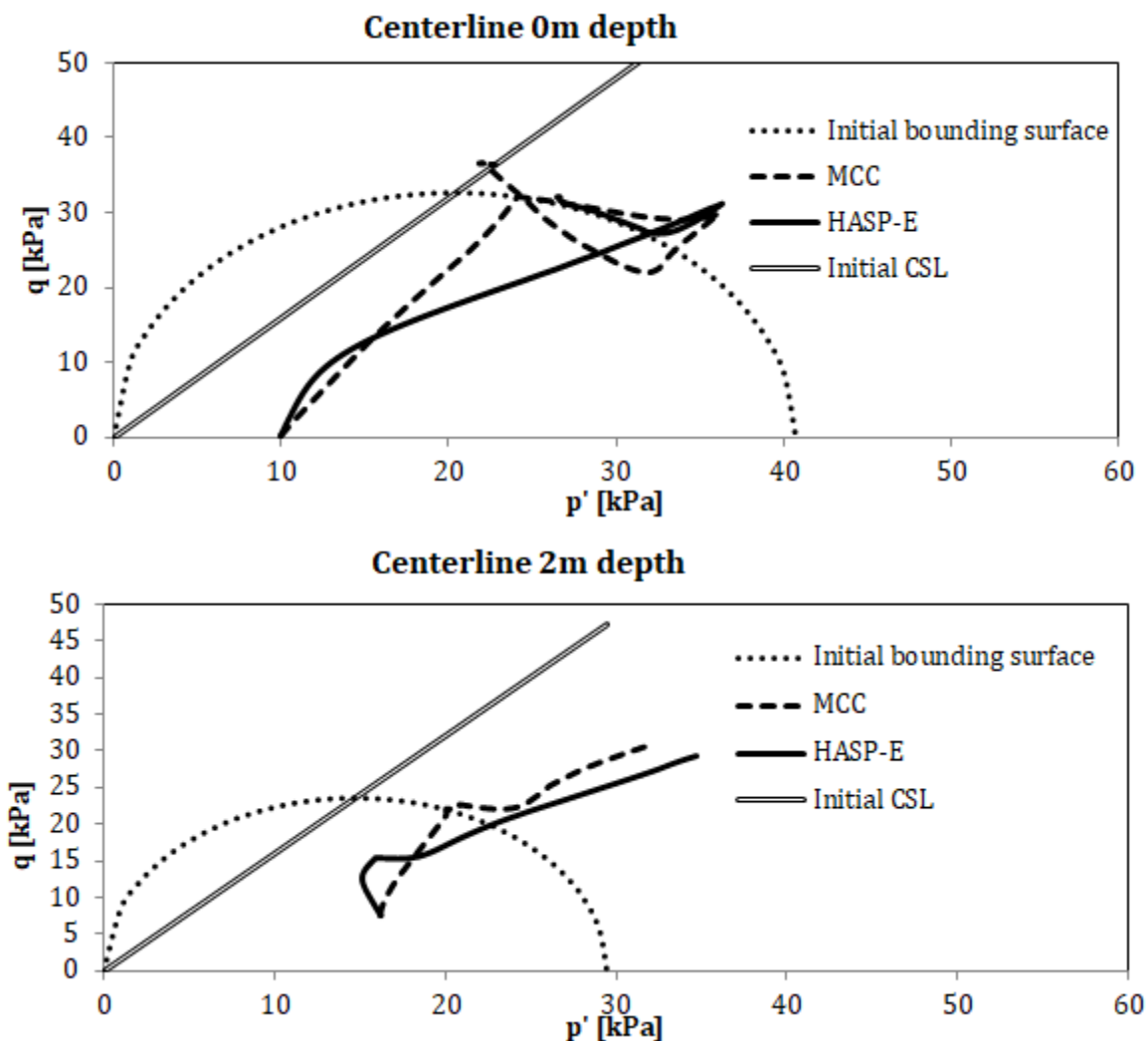


Слика 5-29 Додатни порни притисци у тачки P_1 кроз време за различите методе нумеричке интеграције конститутивних релација



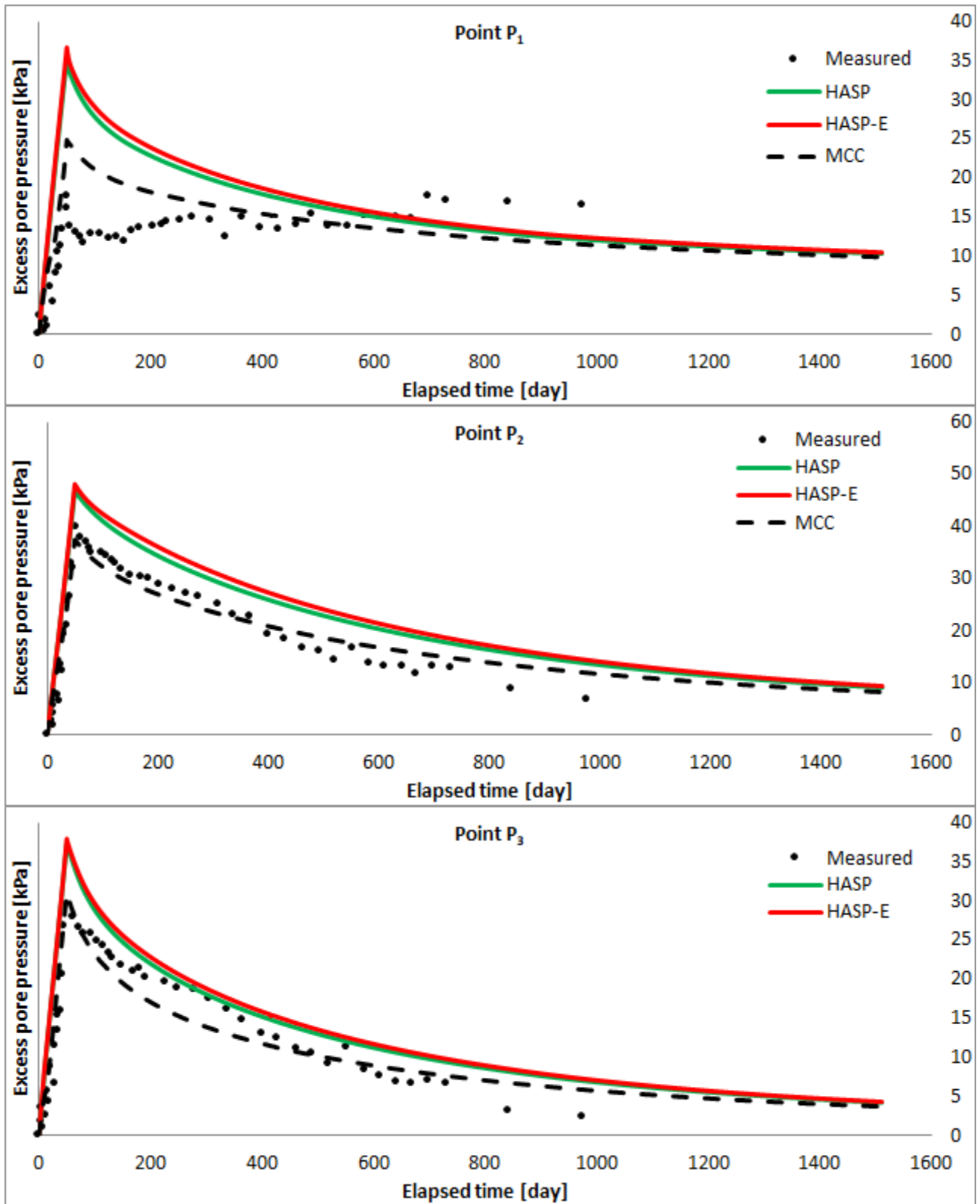
Слика 5-30 Слегање тачака S_0 и S_1 кроз време за Saga Highway Embankment

Пућање ефективних напона за HASP-E и MCC модел за тачке на површини терена и на дубини од 2 m испод средине насипа дате су на **Слика 5-31**. Док је пућања ефективних напона за MCC модел унутар површи течења (иницијална гранична површ), деформације су еластичне. Тек након што пућања ефективних напона додирне површ течења, почињу да се јављају и пластичне деформације. Због тога, MCC модел предвиђа мање деформације тла од измерених. Код HASP модела, од почетка оптерећивања се јављају еластичне и пластичне деформације.

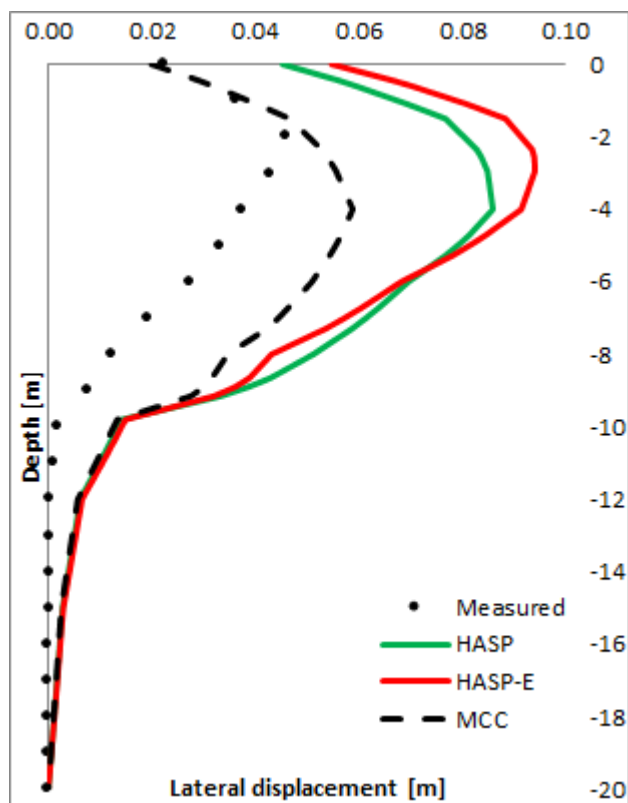


Слика 5-31 Пућање ефективних напона за HASP-E и MCC модел испод средине насипа за тачке на површини терена и на дубини од 2 m

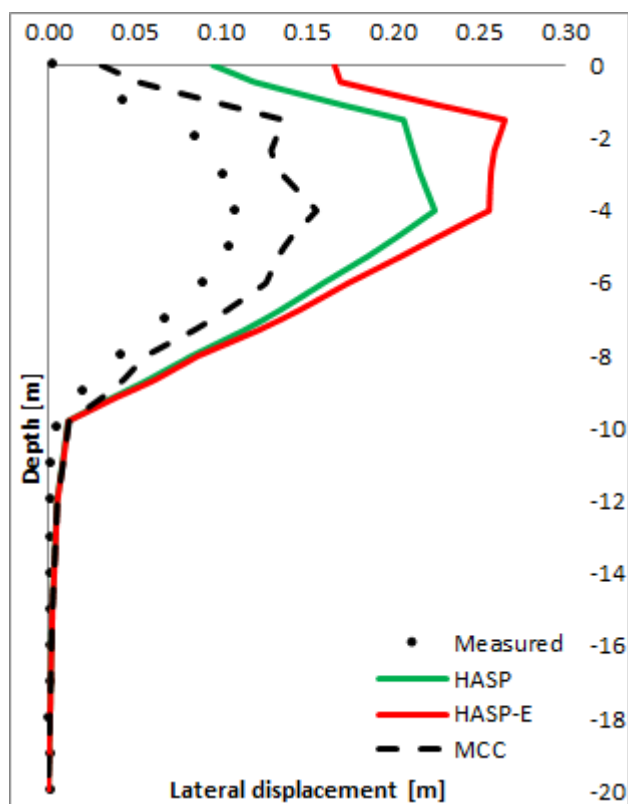
Промена додатних порних притисака кроз време је дата за тачке P_1 (1.4 m испод површине терена), P_2 (5.0 m испод површине терена) и P_3 (8.3 m испод површине терена) на **Слика 5-32**. У осталим тачкама где је била постављена мерна опрема, забележене су веома мале вредности додатних порних притисака, тако да за те тачке резултати нису приказани. Максималне вредности генерисаних порних притисака се јављају непосредно након завршетка изградње насипа. Након тога, њихова вредност опада.



Слика 5-32 Промена додатних порних притисака у тачкама P_1 , P_2 и P_3 кроз време за Saga Highway Embankment



Слика 5-33 Хоризонтална померања тла испод ножице насипа након завршетка изградње насипа



Слика 5-34 Хоризонтална померања тла испод ножице насипа две године након завршетка изградње насипа

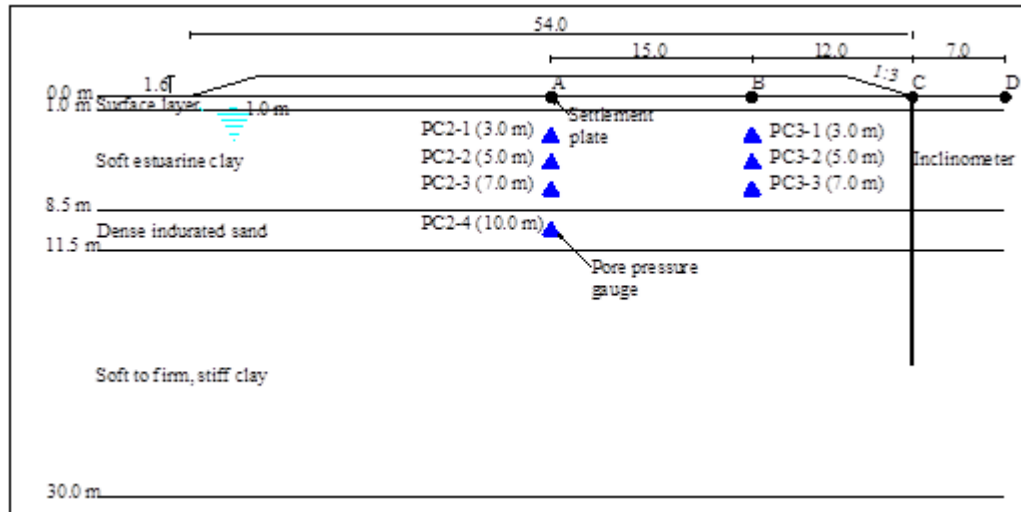
Сва три конститутивна модела дају предвиђања развоја и дисипације додатних порних притисака која су блиска измереним вредностима. МСС модел предвиђа најмањи пик додатних порних притисака, док HASP-E предвиђа највећи пик. HASP модел предвиђа нешто мање порне притиске него HASP-E модел. Четири године након завршетка изградње, сви конститутивни модели предвиђају скоро исте вредности додатних порних притисака.

Хоризонтална померања тла на месту инклинометра (испод ножице насипа) непосредно након завршетка изградње насипа приказана су на **Слика 5-33** и две године након завршетка изградње насипа су приказана на **Слика 5-34**.

МСС модел предвиђа најмања хоризонтална померања тла испод ножице насипа, док HASP-E модел предвиђа највећа хоризонтална померања. HASP модел предвиђа мало мања померања него HASP-E модел, али већа него МСС модел. Предвиђања сва три конститутивна модела су генерално већа него што су измерена хоризонтална померања.

5.5 Предвиђање понашања слоја глине услед изградње насипа – Teven Road Trial Embankment

У овом потпоглављу приказани су резултати нумеричке симулације понашања глине у PLAXIS-у 2D услед изградње насипа на примеру преузетом из (Huang, et al., 2006b) – Teven Road Trial Embankment.



Слика 5-35 Попречни пресек кроз насип са положајем мерних инструмената – адаптирано из (Huang, et al., 2006b)

За потребе изградње аутопута у аустралијској држави Нови Јужни Велс, направљена су два пробна насипа у близини града Ballina. Насипи и темељно тло су били опремљени бројним инструментима како би се утврдило понашање тла услед изградње насипа. Извршена су веома детаљна теренска и лабораторијска испитивања како би се одредили сви параметри тла потребни за пројектовање. Један пробни насип је направљен на природном тлу, а други пробни насип је направљен на побољшаном тлу.

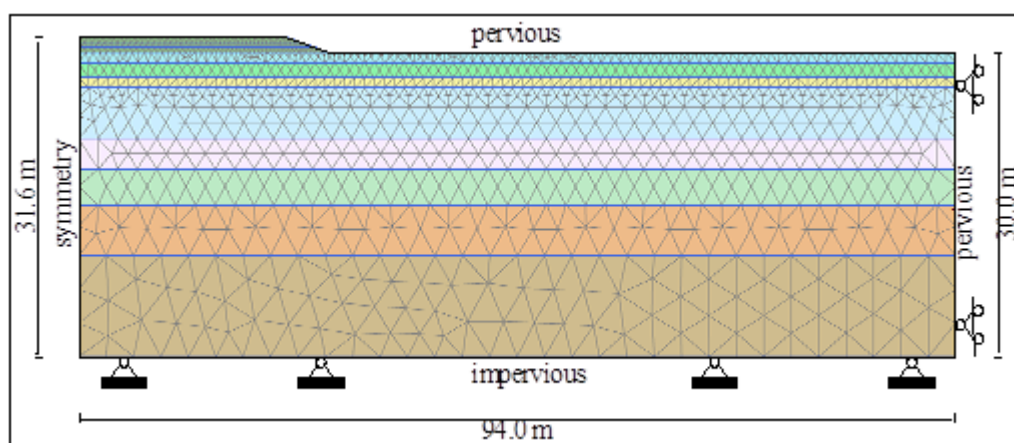
Током и након изградње насипа мерена су слегања површине терена и промене порних притисака у тлу. Мерења су трајала још три године након завршетка изградње насипа.

Профил терена са положајем мерних инструмената је дат на **Слика 5-35**. Пробни насипи су направљени у близини ушћа реке Richmond. На површини терена у дебљини од 1 m налази се слој јако преконсолидоване глине. Испод тог слоја се налази слој меке глине у дебљини од 7.5 m. Испод слоја меке глине налази се слој збијеног песка релативно велике водопропустљивости који се понаша као дренажни слој. Испод слоја песка до дубине од 30 m налазе се слојеви тврдих глине. Ниво подземне воде се налази на дубини од 1 m испод површине терена.

Висина насипа је 1.6 m и изграђен је за 69 дана. Насип је у основи дугачак 84 m и широк 54 m. Нагиб косина насипа је 1: 3 (вертикално:хоризонтално).

5.5.1 Нумерички модел

За нумеричке симулације коришћено је раванско стање деформација. Коришћена мрежа коначних елемената са граничним условима је приказана на **Слика 5-36**. Због симетрије, само једна половина насипа је разматрана у нумеричком моделу. Разматран је домен тла ширине 94 m и дебљине 31.6 m. Коришћено је укупно 2139 коначних елемената и спрегнута напонско-деформацијска и филтрациона анализа (Fully coupled flow-deformation analysis).



Слика 5-36 Мрежа коначних елемената и гранични услови за Teven Road Trial Embankment

Хоризонтална померања на левој и десној ивици нумеричког модела су спречена, док су вертикална слободна. На доњој ивици нумеричког модела, вертикална и хоризонтална померања су спречена, а на горњој ивици су слободна. Од хидрауличних граничних услова, дренарање воде је дозвољено на горњој ивици (површина терена) и на десној ивици како би се омогућило дренарање воде кроз слој песка. Дренарање воде је спречено на левој и доњој ивици нумеричког модела. Хидраулички гранични услови су преузети из (Huang, et al., 2006b).

Прва фаза у нумеричкој симулацији је иницијална фаза у којој се генеришу почетни напони и порни притисци у тлу. Друга фаза је изградња насипа до пуне висине у трајању од 69 дана. Након друге фазе разматрано је још четири фазе од којих свака траје годину дана како би се одредило консолидационо понашање глине након

изградње насипа. Укупан број дана у свих пет фаза је 1530. Минимални број инкремената у свакој фази прорачуна је 1000.

Слој глине изнад нивоа подземне воде је водом засићен са негативним порним притисцима (напони сукције) како би ефективни напони на површини терена били већи од нуле и површински слој глине имао смичућу чврстоћу.

5.5.2 Параметри конститутивних модела

У спроведеним нумеричким симулацијама, за насип и слојеве песка је коришћен Mohr-Coulomb-ов (МС) конститутивни модел, а за слојеве глине су коришћени HASP, HASP-E или MCC модел. Параметри свих конститутивних модела су преузети без измена из (Huang, et al., 2006b). Параметри конститутивних модела су дати у **Табела 5-10**. HASP и MCC модел користе вредност параметра M_c за параметар M . Код HASP-E модела, „Brick“ компонента није била активна.

Табела 5-10 Параметри конститутивних модела за нумеричку симулацију понашања глине услед изградње насипа – *Teven Road Trial Embankment*

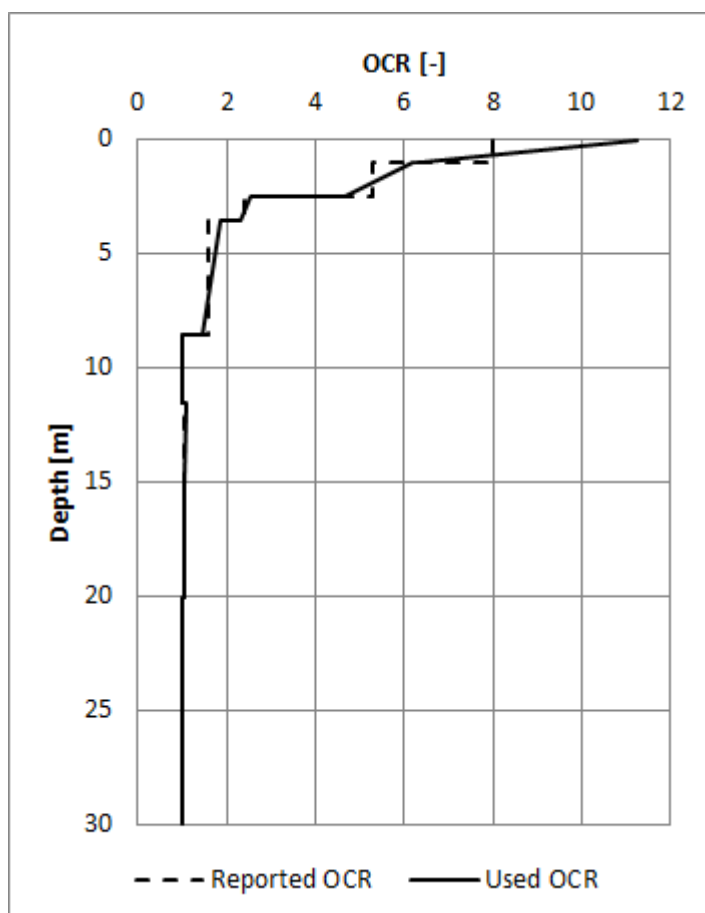
Дубина [m]	Слој тла	Параметри				
		λ [-]	κ [-]	M_c [-]	M_e [-]	μ [-]
0.0-1.0	Површински слој глине	0.078	0.011	1.33	1.064	0.3
1.0-2.5	Мека глина	0.587	0.02	1.07	0.856	0.33
2.5-3.5						
3.5-8.5						
8.5-11.5	Збијени пескови	МС параметри: $E' = 26672 \text{ kPa}$, $c' = 5 \text{ kPa}$, $\phi' = 30^\circ$, $\psi = 0^\circ$, $\mu = 0.28$				
11.5-15.0	Тврда глина	0.134	0.009	1.11	0.888	0.3
15.0-20.0						
20.0-30.0						
Насип		МС параметри: $E' = 5000 \text{ kPa}$, $c' = 10 \text{ kPa}$, $\phi' = 28^\circ$, $\psi = 0^\circ$, $\mu = 0.3$				

Иницијални услови за нумеричке симулације су дати у **Табела 5-11**. Иницијална вредност коефицијента порозности и коефицијент притиска тла у стању мировања за нормално консолидовано тло су преузети из (Huang, et al., 2006b). Преконсолидација слојева глине је задата преко параметра POP чија је вредност за сваки слој глине одређена преко израза (4-296) тако да се добију вредности OCR дате у (Huang, et al., 2006b). Промена OCR по дубини за усвојене вредности POP и вредности OCR из (Huang, et al., 2006b) приказани су на **Слика 5-37**. Коефицијент притиска тла у стању мировања за преконсолидоване слојеве глине је одређен преко израза (4-295) и вредности дате у **Табела 5-11** представљају просечну вредност за један слој глине.

Остали параметри који су потребни за формирање нумеричког модела су дати у **Табела 5-12**. Вредности запреминске тежине и водопропустљивости тла су преузети из (Huang, et al., 2006b). Степен анизотропије у погледу водопропустљивости за све слојеве тла је $k_h/k_v = 2.0$. Водопропустљивост тла се са променом коефицијента порозности мења према изразу (5-1). За вредност C_k је усвојена вредност од $0.4e_0$.

Табела 5-11 Иницијални услови за нумеричку симулацију понашања глине услед изградње насипа – Teven Road Trial Embankment

Дубина [m]	Слој тла	Параметри			
		e_0 [-]	POP [kPa]	K_0^{NC} [-]	K_0^{OC} [-]
0.0-1.0	Површински слој глине	0.8	101	0.45	1.336
1.0-2.5	Мека глина	2.29	98	0.55	1.269
2.5-3.5			41		0.8476
3.5-8.5			27		0.7013
8.5-11.5	Збијени пескови	0.65	0	0.50	
11.5-15.0	Тврда глина	1.69	6	0.53	0.5464
15.0-20.0			4		0.5385
20.0-30.0			0		0.53
Насип		0.5	0	0.53	



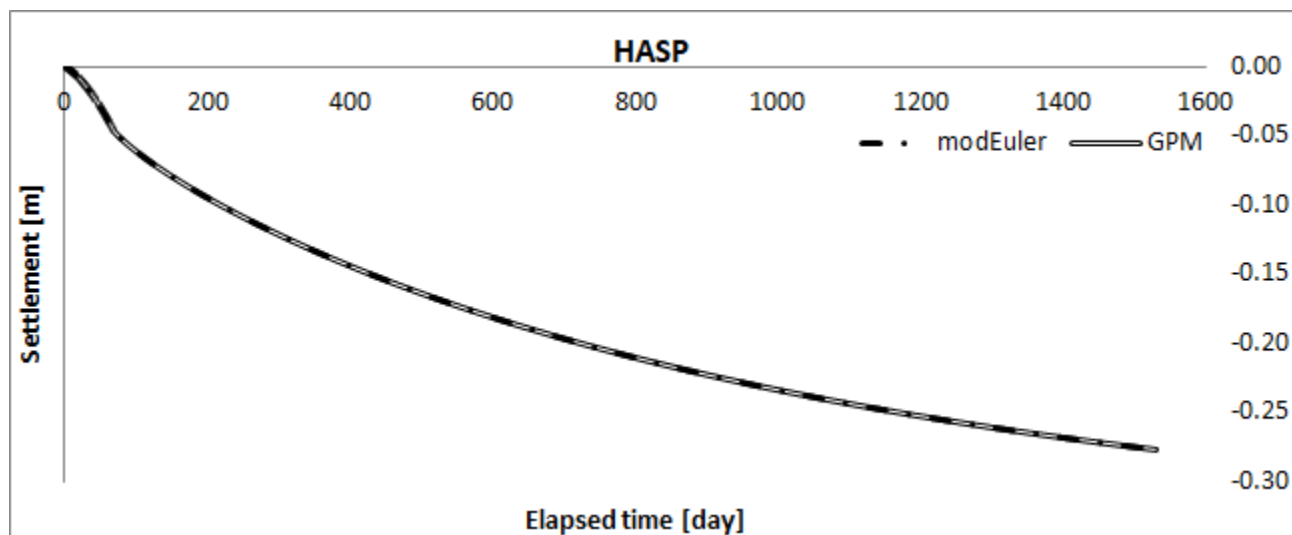
Слика 5-37 Промена степена преконсолидације са порастом дубине у зависности од вредности параметра POP за Teven Road Trial Embankment

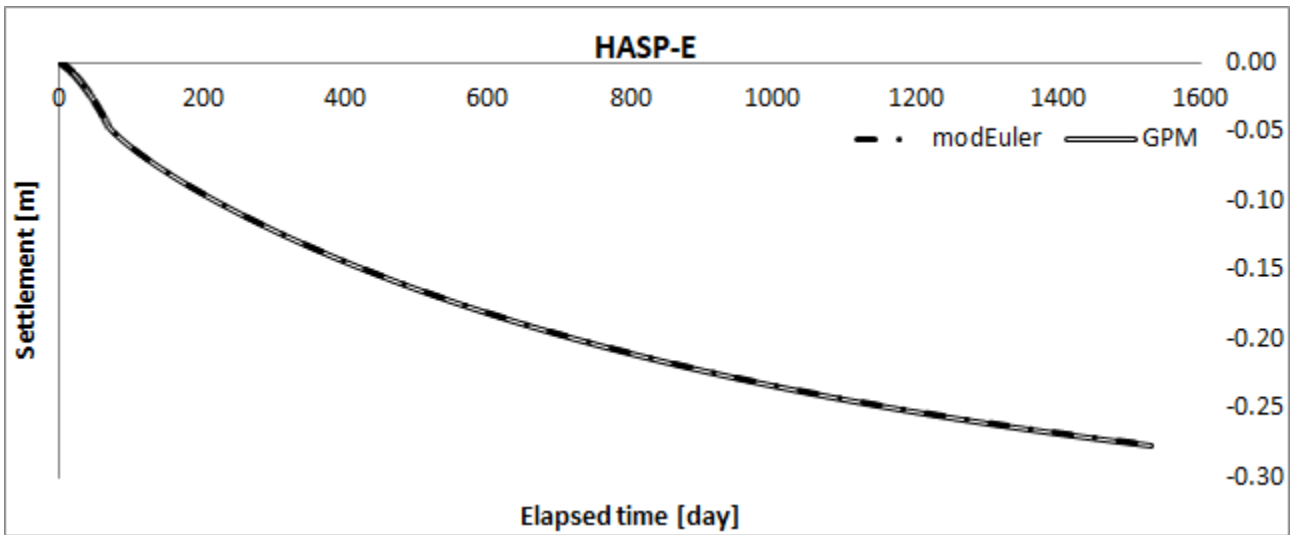
Табела 5-12 Остали параметри нумеричког модела потребни за нумеричку симулацију понашања глине услед изградње насипа – *Teven Road Trial Embankment*

Дубина [m]	Слој тла	Параметри			
		Запреминска тежина [kN/m ³]	k_v [$\frac{10^{-4} m}{day}$]	k_h [$\frac{10^{-4} m}{day}$]	c_k [-]
0.0-1.0	Површински слој глине	18.94	60.48	120.96	0.32
1.0-2.5	Мека глина	15.08	1.43	2.86	0.92
2.5-3.5					
3.5-8.5					
8.5-11.5	Збијени песак	19.8	864	1728	0.26
11.5-15.0	Тврда глина	15.99	0.6	1.2	0.68
15.0-20.0					
20.0-30.0					
Насип		18.0	Веома велика		-

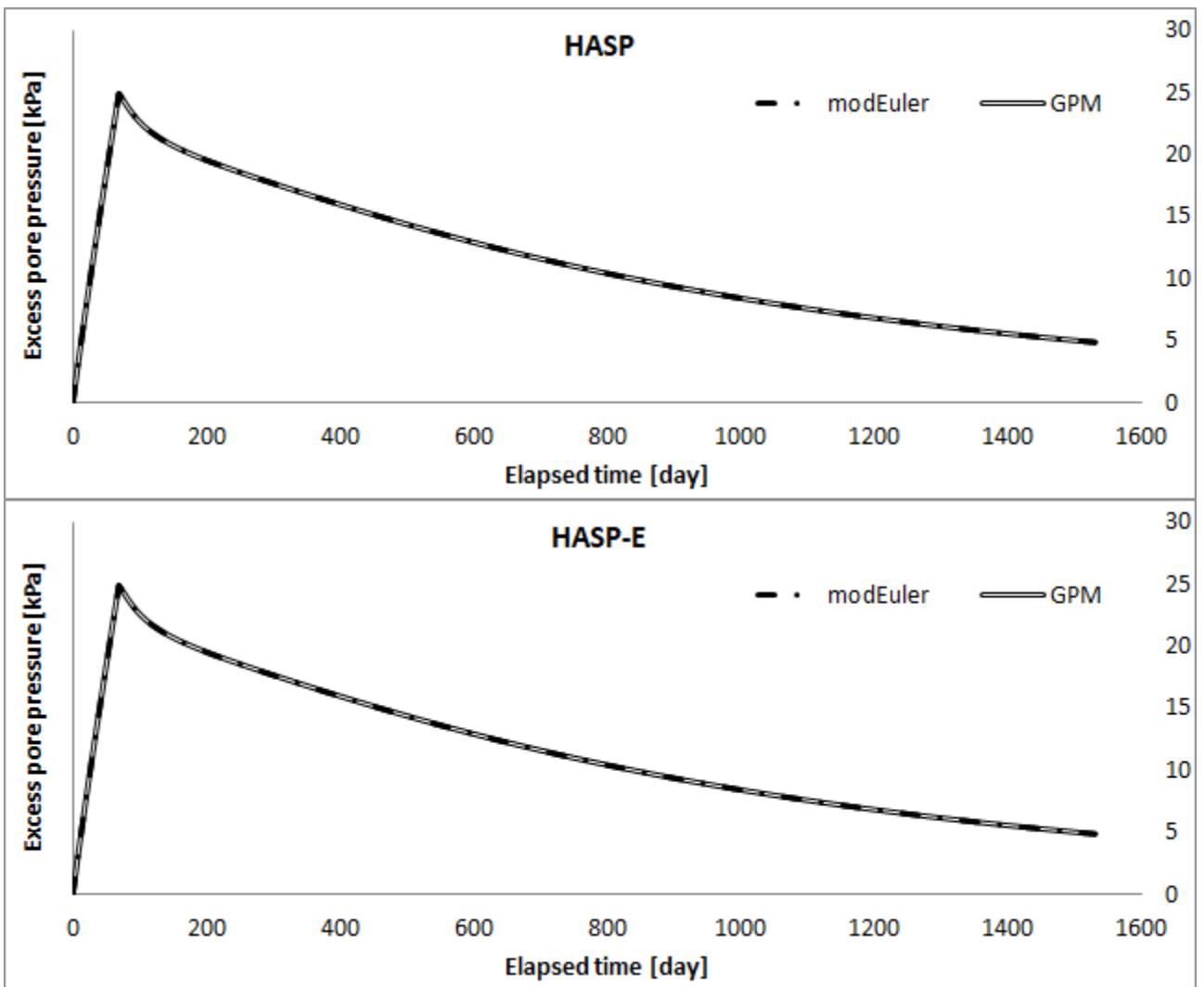
5.5.3 Резултати и дискусија

Прво је спроведена анализа да ли на добијене резултате утиче метода интеграције конститутивних релација HASP и HASP-E модела. Ова анализа је спроведена на исти начин као у потпоглављу 5.4.3. Резултати ове анализе су приказани на **Слика 5-38** и **Слика 5-39**. Из приложених дијаграма може се уочити да избор методе нумеричке интеграције не утиче на добијене резултате. У наставку су за HASP и HASP-E моделе резултати приказани само за модификовани Euler-ов поступак са аутоматском субинкрементацијом и контролом грешке.



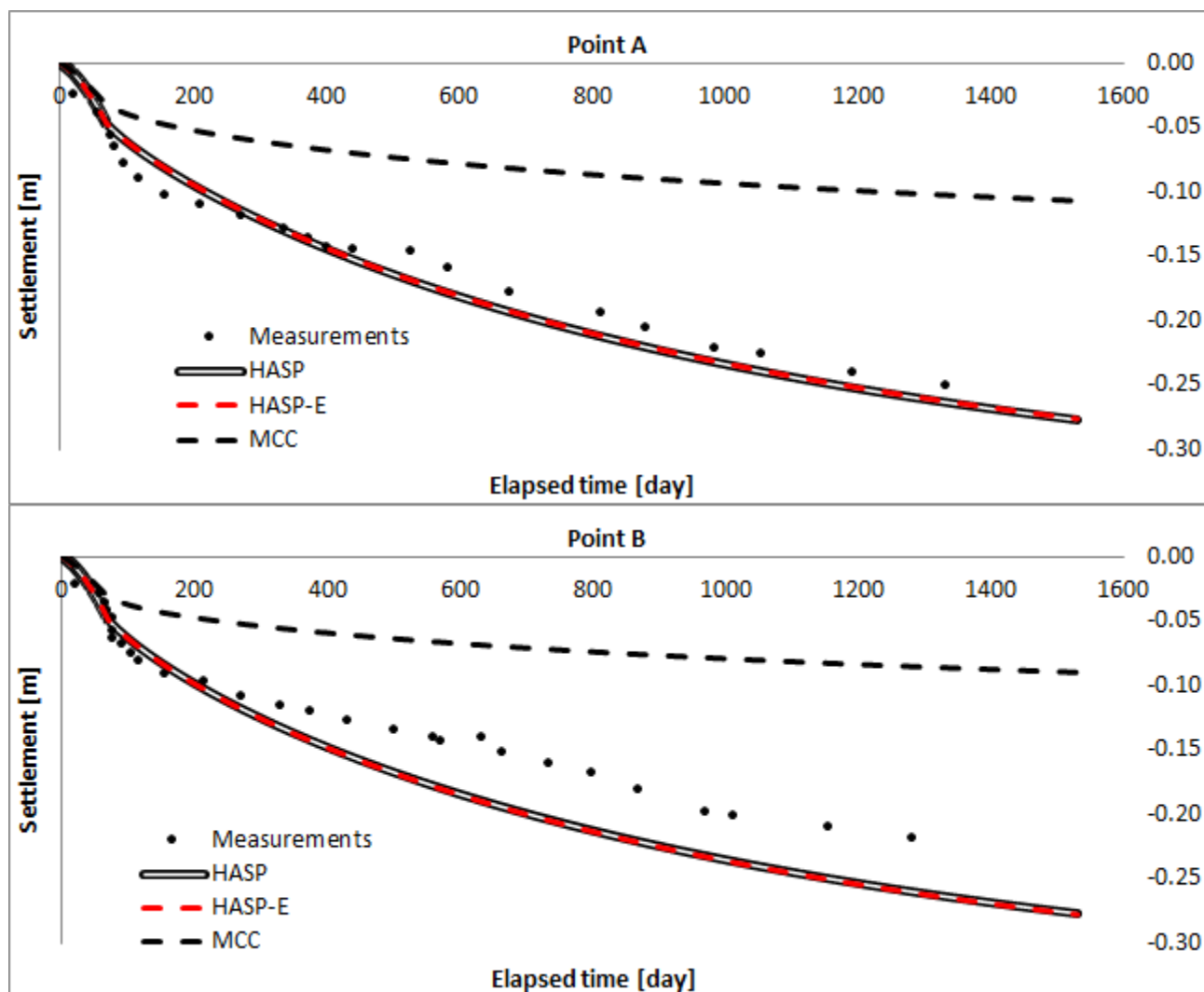


Слика 5-38 Слегање тла у тачки А кроз време за различите методе нумеричке интеграције конститутивних релација



Слика 5-39 Додатни порни притисци у тачки РС2 – 1 кроз време за различите методе нумеричке интеграције конститутивних релација

Слегања тачака *A* и *B* кроз време за HASP, HASP-E и MCC моделе заједно са измереним слегањима су приказани на **Слика 5-40**. MCC модел предвиђа знатно мања слегања него што су измерена. Слегања која се добијају применом HASP и HASP-E модела су међусобно практично иста. Слегања која предвиђају HASP модели су већа него слегања која предвиђа MCC модел и ближа су измереним слегањима. HASP модели дају добра предвиђања слегања за тачку *A*, а за тачку *B* предвиђају већа слегања него што су измерена.

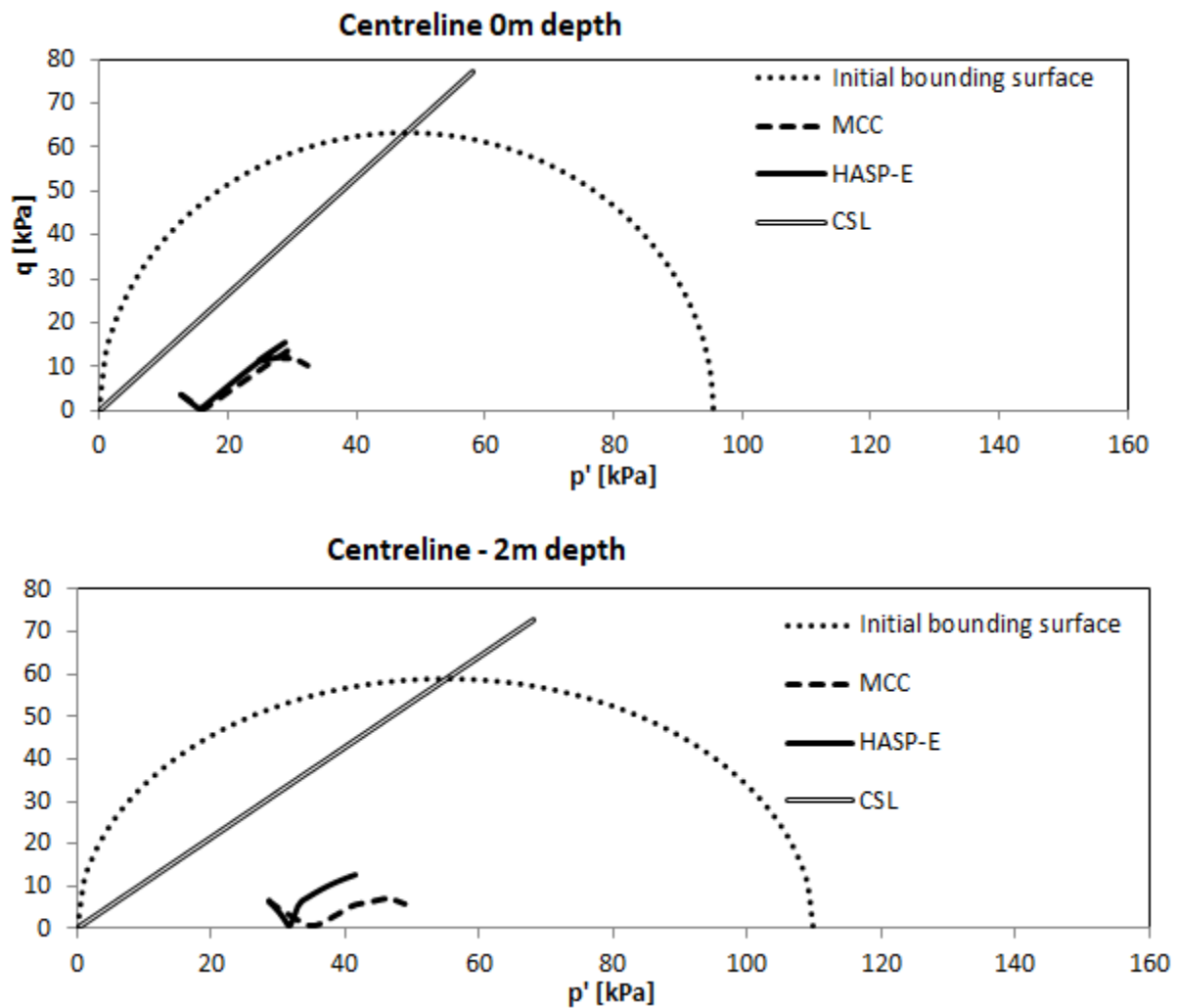


Слика 5-40 Слегање тачака *A* и *B* кроз време за Teven Road Trial Embankment

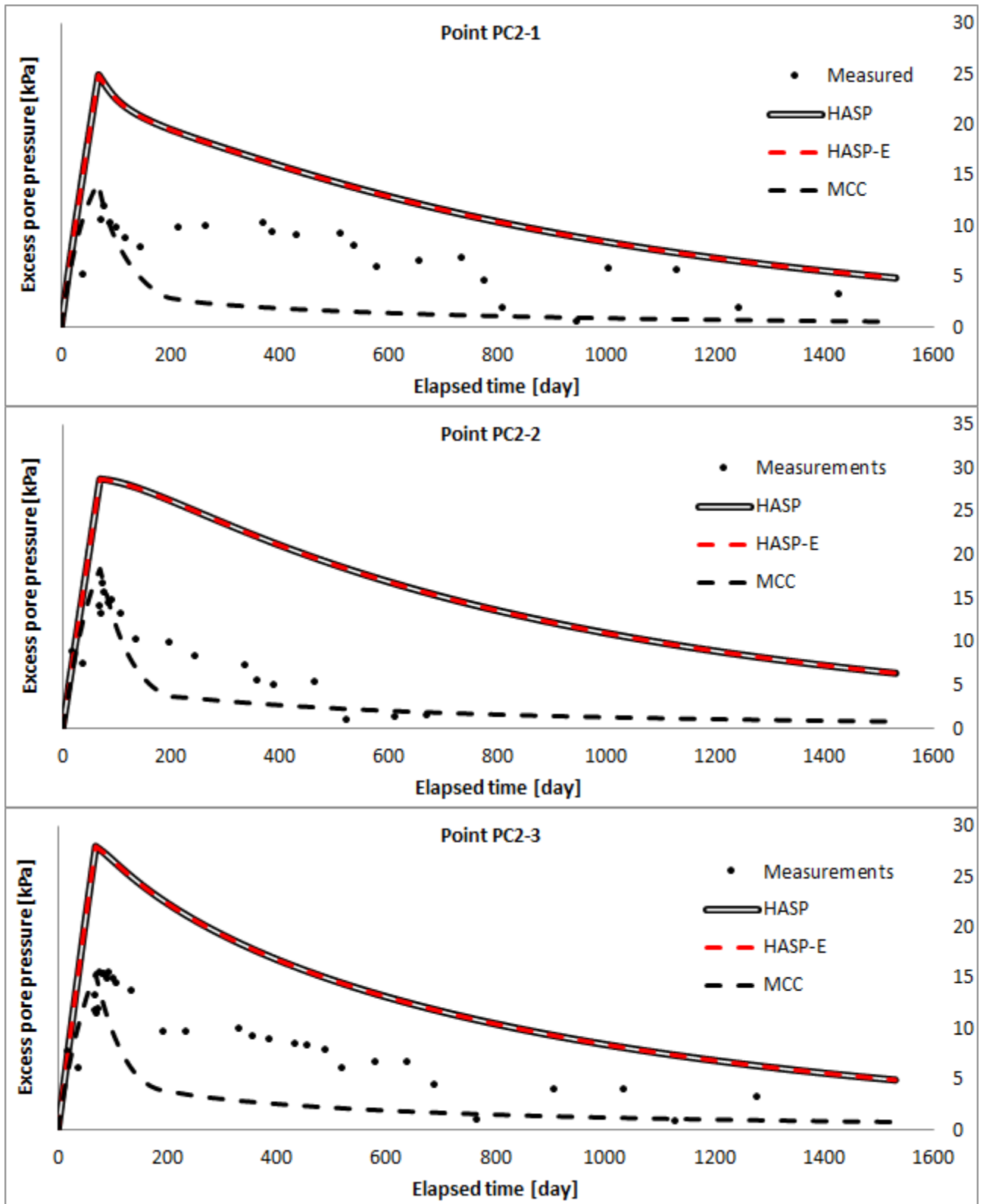
Разлог зашто HASP модели предвиђају већа слегања него што предвиђа MCC модел може се видети на **Слика 5-41**, на којој су приказане путање ефективних напона за тачке на површини терена и на дубини 2 m испод средине насипа. Иницијални услови у овим тачкама су такви да је глина јако преконсолидована. Путања ефективних напона за MCC модел је током целог процеса оптерећивања унутар површи течења (граничне површи за HASP моделе) и деформације су искључиво еластичне. Са друге стране, HASP модели предвиђају поред еластичних и пластичне деформације од самог почетка оптерећивања. Разлог зашто оба HASP модела предвиђају иста слегања би могао да се објасни такође преко приказаних путања напона. За обе тачке, путања напона током целог процеса оптерећивања остаје далеко од линије критичног стања и иницијалне

граничне површи и вредност Lode-овог угла има исту вредност током целог прорачуна, тако да оба конститутивна модела користе исту вредност параметра M .

Развој додатних порних притисака кроз време у тачкама $PC2 - 1$, $PC2 - 2$ и $PC2 - 3$ је приказан на **Слика 5-42**. Сви конститутивни модели предвиђају већи пик додатних порних притисака него што су измерени. Након достизања пика додатних порних притисака (непосредно након завршетка изградње насипа), HASP модели изразито прецењују, док MCC модел потцењује вредности додатних порних притисака. На основу вредности додатних порних притисака на крају прорачуна, четири године након завршетка изградње насипа, предвиђања MCC модела упућују на то да је процес примарне консолидације завршен, док је за HASP моделе потребно још времена да дође до потпуне дисипације додатних порних притисака.



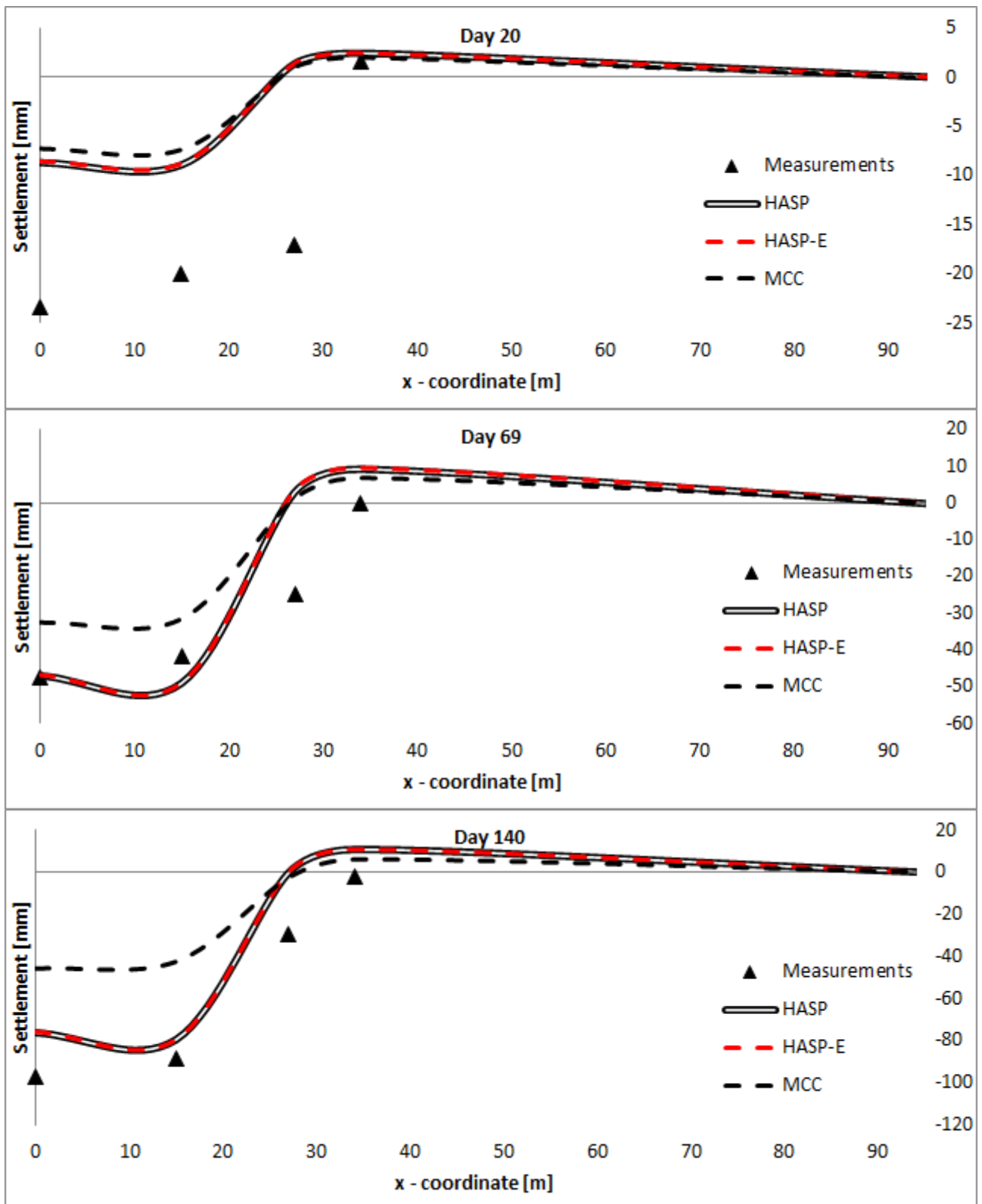
Слика 5-41 Путање ефективних напона за HASP-E и MCC модел испод средине насипа за тачке на површини терена и на дубини од 2 т

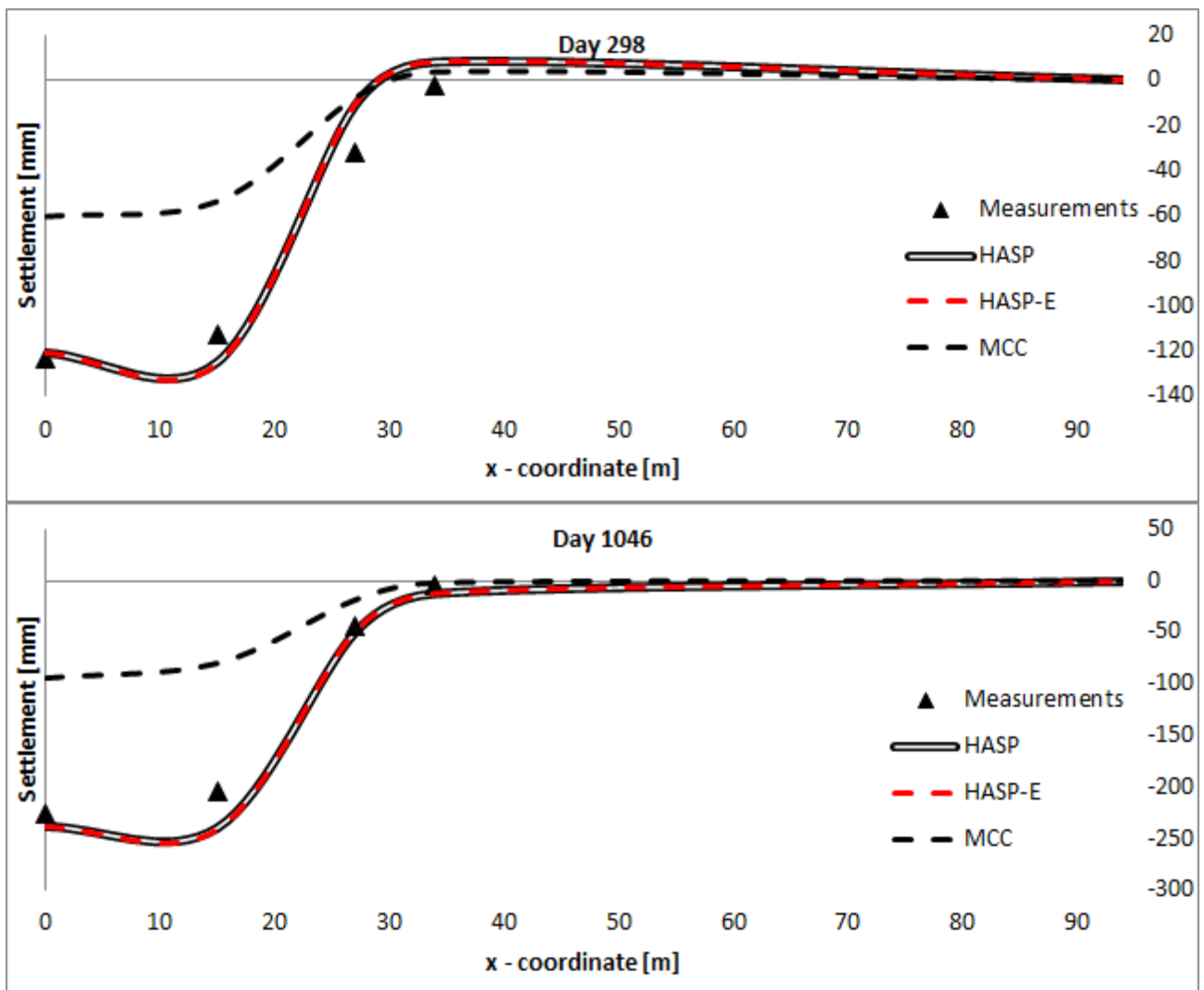


Слика 5-42 Промена додатних порних притисака у тачкама PC2 – 1, PC2 – 2 и PC₂ – 3 кроз време за Teven Road Trial Embankment

Слегање површине терена кроз време је приказано на **Слика 5-43**. Измерена слегања тачака A, B, C и D су била доступна за анализу. Поређења су дата за дан 20 (током

изградње насипа), дан 69 (завршетак изградње насипа) и дане 140, 298 и 1046 након почетка изградње насипа.





Слика 5-43 Слегање површине терена за Teven Road Trial Embankment

Сви конститутивни модели предвиђају облик деформисане површине терена у складу са мерењима. Сви модели предвиђају да након почетка изградње насипа, површина терена изван насипа (тачка *D*) се прво издиже, па онда спушта. MCC модел знатно потцењује величине слегања површине терена током читавог прорачуна. HASP модели задовољавајуће предвиђају деформације површине терена, осим за дан 20 (током изградње насипа). HASP модели предвиђају у свим тачкама већа слегања површине терена него MCC модел. Оба модела предвиђају приближно исто место где нема слегања површине терена.

На основу приказаних резултата понашања преконсолидоване глине услед изградње насипа у овом и претходном потпоглављу, може се закључити да HASP модели предвиђају слегања у тлу која се квалитативно и квантитативно поклапају са резултатима мерења. Са друге стране MCC модел не даје адекватна предвиђања слегања тла. Због тога што су површински слојеви глине преконсолидовани, MCC модел предвиђа само еластичне деформација, док HASP модели предвиђају и еластичне и пластичне деформације од почетка оптерећивања. HASP модели прецењују величину хоризонталних померања тла (Saga Highway Embankment). За оба насипа, HASP модели прецењују пик додатних порних притисака у тлу услед изградње насипа и време потребно за завршетак примарне консолидације у односу на резултате мерења.

Поглавље 6 Закључна разматрања и препоруке за даља истраживања

6.1 Закључна разматрања

Истраживање које је представљено у дисертацији је реализовано кроз неколико фаза:

1. Проширивање релација основне формулације HASP модела на генерални напонски простор узимањем у обзир утицај треће напонске инваријанте (Lode-ов угао) и узимањем у обзир промене модула смицања са променом смичућих деформација. HASP модел са проширеним релацијама је назван HASP-E модел.
2. Развој алгоритама за нумеричку интеграцију конститутивних релација HASP и HASP-E модела и имплементација конститутивних модела у рачунарски програм PLAXIS.
3. Верификација поступка имплементације конститутивних модела.
4. Анализа ефикасности, тачности и робустности алгоритама за нумеричку интеграцију конститутивних релација.
5. Валидација HASP-E модела кроз симулацију дренираних и недренираних цикличних триаксијалних опита и кроз примену конститутивног модела у анализи граничног проблема – понашање слоја преконсолидоване глине услед изградње насипа

6.1.1 Проширивање релација HASP модела и формулација HASP-E модела

HASP модел је формулисан за стање напона које влада у триаксијалним опитима и због тога је примена овог модела у анализи граничних проблема ограничена. Релације HASP-E модела су проширене тако да превазилазе неке од недостатака HASP модела уз услов да релације проширеног конститутивног модела буду релативно једноставне за имплементацију у рачунарски програм заснован на МКЕ, да основни принципи на којима је развијен HASP модел буду задржани и да сви параметри модела буду јасно дефинисани и да се могу одредити из конвенционалних лабораторијских опита.

У изразима за површ течења и граничну површ, параметар M , који је константа у HASP моделу, у HASP-E моделу је променљива која зависи од вредности Lode-овог угла. За функцију која описује промену параметра M са променом Lode-овог угла, усвојена је релација коју је предложио (van Eekelen, 1980). Вредности константи у претходној зависности се одређују тако да се за вредности Lode-овог угла за услове триаксијалне компресије и екстензије добије вредност параметра M као из тих опита и да облик површи течења и граничне површи у девијаторској равни буде заобљена конвексна површ. За облик површи пластичног потенцијала у девијаторској равни је усвојен облик круга, по угледу на слична унапређења MCC модела из литературе, тако да HASP-E користи неасоцијативан закон течења, за разлику од HASP модела који користи асоцијативан закон течења.

Крутост тла у подручју веома малих и малих деформација, као и промена величине крутости тла при промени путање напона и деформација узета је у обзир прикључењем

“overlay” “Brick” модела из (Cudny & Truty, 2020) релацијама HASP модела. Тангентни модул смицања у HASP-E моделу се рачуна из “overlay” “Brick” модела и користи се исти облик S-криве као у OC_CLAY, HSS и HS-Brick конститутивним моделима. Poisson-ов коефицијент има константну вредност у HASP-E (као и у HASP), тако да се из познате вредности тангентног модула смицања може одредити тангентни запремински модул. Као последица константне вредности Poisson-овог коефицијента током целог прорачуна, нагиб линије растерећења и поновног оптерећења у $v - \ln p'$ равни има променљиву вредност κ . У HASP-E моделу користи се променљива вредност κ у закону ојачања, који има исти облик као у HASP моделу и константна вредност k у изразима за одређивање коефицијента ојачања ω и параметра Γ .

У односу на HASP, HASP-E модел користи четири нова материјална параметра. Уместо параметра дрениране смичуће чврстоће M , користе се параметри M_c и M_e који представљају нагибе линије критичног стања у $q - p'$ равни за триаксијалну компресију и триаксијалну екстензију. За описивање облика S-криве користе се референтни модул смицања при веома малим деформацијама G_0^{ref} и смичућа деформација $\gamma_{0,7}$ при којој је секантни модул смицања редукован на 70 % иницијалне вредности.

Са овако проширеним релацијама, повећавају се могућности HASP-E модела да на адекватан начин опише понашање преконсолидованих глина у анализи граничних проблема и при цикличном оптерећењу, уз релативно мали број материјалних параметара са јасним физичким значењем, који се могу на једноставан начин одредити из конвенционалних лабораторијских опита.

6.1.2 Развој алгоритама за нумеричку интеграцију конститутивних релација

Конститутивни модели HASP и HASP-E модел су имплементирани у рачунарски програм PLAXIS заснован на MKE преко UDSM потпрограма користећи рачунарски језик FORTRAN. За имплементацију оба конститутивна модела развијени су експлицитни и имплицитни алгоритми за нумеричку интеграцију конститутивних релација. Имплементација HASP модела је извршена имплицитном методом водећег параметра са средњим ефективним напоном као водећим параметром, имплицитним алгоритмом (Bogja & Lee, 1990), експлицитним модификованим Euler-овим поступком са аутоматском субинкрементацијом и контролом грешке и експлицитним Runge-Kutta-Dormand-Prince поступком са аутоматском субинкрементацијом и контролом грешке. Имплементација HASP-E модела је извршена имплицитном методом водећег параметра са средњим ефективним напоном као водећим параметром, експлицитним модификованим Euler-овим поступком са аутоматском субинкрементацијом и контролом грешке и експлицитним Runge-Kutta-Dormand-Prince поступком са аутоматском субинкрементацијом и контролом грешке.

Верификација поступка имплементације HASP и HASP-E модела извршена је на два начина:

1. Поређењем резултата симулације CD триаксијалног опита на Fujinomi глини и CU триаксијалног опита на Cardiff глини у PLAXIS-у 2D са резултатима за Euler-

ову интеграцију деформација дуж познате путање напона (EIPN) и са резултатима предвиђања OC_CLAY модела,

2. Поређењем резултата симулације CD цикличног триаксијалног опита на Fujiomogi глини и CU цикличног триаксијалног опита на Newfield глини у PLAXIS-у 2D са резултатима предвиђања OC_CLAY модела.

За оба начина верификације, приказани су резултати симулације и то: напонско-деформацијске криве, промене запремине у дренираним опитима и промена додатних порних притисака у недренираним опитима.

Код монотоних опита, резултати за све методе нумеричке интеграције HASP и HASP-E модела се изузетно добро поклапају са резултатима EIPN методе нумеричке интеграције и са предвиђањима OC_CLAY модела. Код CD опита триаксијалне екстензије, за степене преконсолидације од 8 и 4, HASP модели имплементирани са имплицитним алгоритмима имају потешкоћа када напонска путања прелази први пут линију критичног стања. Имплицитни алгоритми превазилазе тај проблем, али се јавља грешка у предвиђању напона. Од величине инкремента зависи величина грешке. Без обзира на то, сви алгоритми предвиђају исту вредност вршног напонског односа и разлике у предвиђањима се јављају тек при већим девијаторским деформацијама. Проблем при прорачуну напона при првом преласку линије критичног стања код имплицитних алгоритама би могао да се превазиђе увођењем субинкрементације или коришћењем мањих инкремената. Такође, HASP-E модел може да на основу стања напона предвиди понашање глине, односно да ли је у питању екстензија или компресија, док за HASP и OC_CLAY модел је потребно унети одговарајућу вредност параметра M (M_c или M_e) у зависности од типа опита.

Код цикличних опита, предвиђања HASP-E модела се веома добро поклапају са предвиђањима OC_CLAY модела. Мале разлике између HASP-E модела са активном "Brick" компонентном и OC_CLAY модела са активним SSO моделом могу бити последица коришћења степенасте S-криве у HASP-E моделу, уместо континуалне криве у OC_CLAY моделу. Ове разлике су израженије у недренираним опитима.

6.1.3 Анализа ефикасности, тачности и робустности алгоритама за нумеричку интеграцију конститутивних релација

Алгоритми за нумеричку интеграцију HASP и HASP-E модела су међусобно упоређени према критеријумима ефикасности, тачности и робустности кроз анализу граничног проблема – одређивање носивости тракастог темеља у недренираним условима у програму PLAXIS 2D. Упоредна анализа за HASP модел је извршена за три алгоритма и то: метода водећег параметра, алгоритама (Borja & Lee, 1990) и модификовани Euler-ов поступак са аутоматском субинкрементацијом и контролом грешке. Упоредна анализа за HASP-E модел је извршена за два алгоритма - методу водећег параметра и модификовани Euler-ов поступак са аутоматском субинкрементацијом и контролом грешке. Као "тачно" решење за оба конститутивна модела, коришћен је експлицитни RKDP поступак са аутоматском субинкрементацијом и контролом грешке са малом толеранцијом на локалну грешку. У анализи су варирани иницијални услови и величина инкремента наносења оптерећења, а код експлицитног алгоритма додатно је варирана величина толеранције на локалну грешку *STOL*.

Из приказаних резултата, имплицитни алгоритам (Borja & Lee, 1990) захтева најмањи утрошак рачунарског времена, али даје решења мање тачности него друга два алгоритма. Такође, код овог алгоритма се јавља проблем са конвергенцијом глобалног инкрементално-итеративног поступка за јако преконсолдовано тло када се оптерећење наноси у малом броју инкремената. Метода водећег параметра захтева већи утрошак рачунарског времена, али даје решење веће тачности. Модификовани Euler-ов поступак са аутоматском субинкрементацијом и контролом грешке захтева највећи утрошак рачунарског времена, али даје решење највеће тачности. У односу на утицај иницијалних услова и величину инкремената, експлицитни алгоритми су робустнији него имплицитни. За велики број инкремената (односно малу величину инкремента) у којима се наноси оптерећење, сви алгоритми предвиђају скоро исту граничну носивост. Код експлицитног алгоритма, што је већи број коришћених инкремената, вредност *STOL* не утиче битно на добијени резултат. Без обзира на величину и број инкремента, вредност *STOL* од 10^{-4} код експлицитне методе даје решења задовољавајуће тачности.

Из приказаних резултата може се још уочити да код конститутивних модела са неасоцијативним законом течења потребан је већи утрошак рачунарског времена него код модела са асоцијативним законом течења. Такође се може уочити да са порастом степена преконсолидације расте утрошак рачунарског времена.

6.1.4 Валидација HASP-E модела

Валидација HASP-E модела је извршена кроз симулацију дренираних и недренираних цикличних триаксијалних опита и кроз примену конститутивног модела у анализи граничног проблема – понашање слоја преконсолидоване глине услед изградње насипа.

Симулирани су CD циклични триаксијални опити на Fujinomorі глини и CU циклични триаксијални опити на Newfield глини. Симулације су прво извршене са материјалним параметрима из литературе, а онда је спроведена параметарска анализа са циљем да се испита како промена материјалних параметара утиче на предвиђање понашања глине. На основу резултата параметарске анализе, извршена је калибрација материјалних параметара како би се резултати симулације што више поклапали са експерименталним резултатима.

Из приказаних резултата цикличних триаксијалних опита јасно се уочавају предности HASP-E модела у описивању понашања глине у односу на HASP и MCC моделе. Иако је глина на почетку сваког опита нормално консолидована, након растеређивања и при поновном оптеређивању постаје преконсолидована и MCC модел не може ни квалитативно, ни квантитативно да опише понашање глине и предвиђа искључиво еластичне деформације приликом растеређивања и поновног оптеређивања. HASP модел даје боља предвиђања понашања тла него MCC модел, може донекле да репродукује хистерезис који се јавља у једном циклусу оптерећења и растеређења, али величина хистерезиса је мања и деформације су веће него измерене. Предвиђања HASP-E модела могу да опишу понашање глине при цикличном оптерећењу у дренираним и недренираним условима. Предвиђања HASP-E су веома блиска предвиђањима OC_CLAY модела и мале разлике између предвиђања HASP-E и OC_CLAY модела су последица различитог облика S-криве коју ови модели користе.

На основу приказаних резултата нумеричке симулације цикличних триаксијалних опита могу се извести следећи општи закључци о могућностима HASP-E модела:

- HASP-E модел може да се користи за немонотона оптерећења, као што је растерећење тла и циклично оптерећење,

- HASP-E модел може да опише хистерезисно понашање при растерећењу и поновном оптерећењу,

-“Brick” компонента HASP-E модела може на адекватан начин да опише величину хистерезиса,

- HASP-E модел је најосетљивији на промену параметара деформабилности λ и κ ,

- у дренираним цикличним опитима, HASP-E модел квалитативно и уз адекватну калибрацију материјалних параметара и квантитативно добро описује понашање тла,

- у недренираним цикличним опитима, HASP-E модел квалитативно и уз калибрацију материјалних параметара и квантитативно добро описује величину деформација и додатних порних притисака.

Могућности HASP-E модела да опише понашање слоја преконсолидоване глине услед изградње насипа приказана је на два примера из литературе: Saga Highway Embankment и Teven Road Trial Embankment. За оба насипа, HASP-E модел предвиђа слегања тла која су мало већа од измерених, док MCC модел значајно потцењује величине слегања. Пошто је површински слој глине за оба насипа јако преконсолидован, MCC модел предвиђа еластичне деформације током већег дела оптерећивања, док HASP-E предвиђа еласто-пластичне деформације од самог почетка оптерећивања. Предвиђања величине додатних порних притисака и њихова дисипација кроз време HASP-E и MCC модела се веома добро слажу са измереним вредностима. Већи пик додатних порних притисака него што је измерено предвиђа HASP-E модел и време потребно за потпуну дисипацију додатних порних притисака је веће него што предвиђа MCC модел и у односу на мерења. За Saga Highway Embankment, HASP-E модел даје већа предвиђања хоризонталних померања у ножици насипа на месту инклинометра него што су измерена. За Teven Road Trial Embankment, HASP-E предвиђа слегања површине терена која се добро поклапају са измереним.

6.2 Препоруке за даља истраживања

Даља истраживања везана за унапређење релација тренутне формулације HASP-E модела:

1. Модификација израза за одређивање параметра Γ и коефицијента ојачања ω тако да се узме у обзир променљив нагиб линије бубрења у $\ln p' - v$ равни
2. Проширивање релација модела тако да се уведе могућност да запремински модул и модул смицања буду међусобно независни (варијабилан Poisson-ов коефицијент)
3. Утврђивање адекватног облика површи пластичног потенцијала у девијаторској равни. Спровођење додатних анализа како би се утврдило да ли је адекватније

наставити коришћење неасоцијативног или користити асоцијативни закон течења.

4. Додатне анализе како би се у потпуности утврдиле могућности и недостаци HASP-E модела у предвиђању понашања нормално консолидованих и преконсолидованих глина при цикличном оптерећењу.

Даља истраживања везана за општа унапређења формулације HASP модела:

1. Проширивање релација модела за предвиђање механичког понашања преконсолидованих делимично засићених глина.
2. Проширивање релација модела за предвиђање механичког понашања преконсолидованих глина узимањем у обзир напонски индуковане анизотропије.
3. Проширивање релација модела за описивање реолошког понашања преконсолидованих глина склоних секундарној компресији.
4. Проширивање релација модела за предвиђање механичког понашања преконсолидованих глина узимањем у обзир деструктурирације глинe.

Даља истраживања везана за развој и унапређење алгоритама за нумеричку интеграцију конститутивних релација:

1. Формирање и имплементација у рачунарски програм заснован на МКЕ еласто-пластичне матрице како би се повећала брзина конвергенције у глобалном Newton-Raphson-овом итеративном поступку.
2. Унапређење имплицитних алгоритама који су разматрани у оквиру ове дисертације у погледу превазилажења проблема који се јавља при прорачуну напона када напонска путања прелази први пут линију критичног стања.
3. Унапређење имплицитних алгоритама укључивањем субинкрементације и контроле грешке.
4. Развој методе водећег параметра разматрањем могућности употребе других величина за водећи параметар, као што су прираштај пластичних запреминских деформација и параметар ојачања на крају инкремента.

Литература

Al-Tabba, A. & Wood, D. M., 1989. *An experimentally based "bubble" model for clay*. Rotterdam, Proceedings of Numerical Modelling in Geomechanics.

Banerjee, P. K. & Stipho, A. S., 1978. Associated and non-associated constitutive relations for undrained behaviour of isotropic soft clays. *International Journal for Numerical and Analytical Methods in Geomechanics*, 2(1), pp. 35-56.

Bardet, J. P., 1990. Lode Dependences for Isotropic Pressure-Sensitive Elastoplastic Materials. *Journal of Applied Mechanics*, Том 57, pp. 498-506.

Bentley Systems, Incorporated, 2023. *Bentley communities*. [На мрежи] Available at: <https://communities.bentley.com/products/geotech-analysis/w/wiki/46137/manuals---plaxis> [Последњи приступ 21 November 2023].

Bentley, 2019. *OC-Clay - PLAXIS UDSM*. [Online] Available at: [https://communities.bentley.com/cfs-file/key/communityserver-wikis-components-files/00-00-00-05-58/PLAXIS 2D00 UDSM 2D00 OCclay 5F002800 V20Update1 2900 .pdf](https://communities.bentley.com/cfs-file/key/communityserver-wikis-components-files/00-00-00-05-58/PLAXIS%20UDSM%20OCclay%205F002800%20V20Update1%202900.pdf) [Accessed 6 April 2023].

Benz, T., 2007. *Small-strain stiffness of soils and its numerical consequences*. Ph.D.: University of Stuttgart.

Bolton, M. D., Dasari, G. R. & Britto, A. M., 1994. *Putting small strain non-linearity into Modified Cam Clay model*. Virginia, USA, Eight International Conference on Computer Methods and Advances in Geomechanics.

Borja, R. I., 1991. Cam-Clay plasticity, Part II: Implicit integration of constitutive equation based on a nonlinear elastic stress predictor. *Computer Methods in Applied Mechanics and Engineering*, Том 88, pp. 225-240.

Borja, R. I. & Lee, S. R., 1990. Cam-Clay Plasticity, part I: Implicit Integration of Elasto-Plastic Constitutive Relations. *Computer Methods in Applied Mechanics and Engineering*, 78(1), pp. 49-72.

Borja, R. I., Sama, K. M. & Sanz, P. F., 2003. On the numerical integration of three-invariant elastoplastic constitutive models. *Computer Methods in Applied Mechanics and Engineering*, Том 192, pp. 1227-1258.

Carow, C. & Rackwitz, F., 2021. Comparison of implicit and explicit numerical integration schemes for a bounding surface soil model without elastic range. *Computers and Geotechnics*, Том 140, p. 104206.

- Castellon, J. L., 2019. *Estudio del comportamiento del suelo en el rango de las pequenas deformaciones y desarrollo del modelo constitutivo EPHYSS*. Phd dissertation: Universitat Politecnica de Catalunya.
- Castellon, J. & Ledesma, A., 2022a. Small Strains in Soil Constitutive Modeling. *Archives of Computational Methods in Engineering*, Issue 29, pp. 3223-3280.
- Castellon, J. & Ledesma, A., 2022b. Development of a new advanced elastoplastic constitutive model that considers soil behavior at small strains. The EPHYSS model. *International Journal for Numerical and Analytical Methods in Geomechanics*, Tom 46, pp. 1991-2032.
- Chai, J. C. & Miura, N., 1999. Investigation on some factors affecting vertical drain behavior. *Journal of Geotechnical and Geoenvironmental Engineering*, Volume 125, pp. 216-226.
- Chai, J., Igaya, Y., Hino, T. & Carter, J., 2013. Finite element simulation of an embankment on soft clay - Case study. *Computers and Geotechnics*, Volume 48, pp. 117-126.
- Chapra, S. C. & Canale, R. P., 2021. *Numerical Methods for Engineers*. 8th yp. New York: McGraw-Hill.
- Chen, C., Sun, Z., Wu, X. & Wang, Y., 2024. Implicit integration with a substepping scheme of the zero-elastic range SANISAND-Z model for sand. *Computers and Geotechnics*, Tom 165, p. 105899.
- Clarke, S. D., 2009. *Enhancement of the BRICK constitutive model to incorporate viscous soil behaviour*. Phd thesis: The University of Sheffield.
- Clarke, S. D. & Hird, C. C., 2012. Modelling of viscous effects in natural clays. *Canadian Geotechnical Journal*, Tom 49, pp. 129-140.
- Clayton, C. R. I., 2011. Stiffness at small strain: research and practice. *Geotechnique*, 61(1), pp. 5-37.
- Cudny, M. & Truty, A., 2020. Refinement of the hardening soil model within the small strain range. *Acta Geotechnica*, Tom 15, pp. 2031-2051.
- De la Morena, G., Asensio, L., Navarro, V. & Yustres, A., 2017. A simple procedure to improve the explicit integration of Cam-Clay models. *Computers and Geotechnics*, Tom 81, pp. 207-211.
- Dong, Y., 2023. Performance of explicit substepping integration scheme for complex constitutive models in finite element analysis. *Computers and Geotechnics*, Tom 162, p. 105629.
- Dormand, J. R. & Prince, P. J., 1980. A family of embedded Runge-Kutta formulae. *Journal of Computational and Applied Mathematics*, 6(1), pp. 19-26.
- E-Kan, M. & Taiebat, H. A., 2013. On implementation of bounding surface plasticity models with no overshooting effect in solving boundary value problems. *Computers and Geotechnics*, Tom 55, pp. 113-116.

- Ellison, K. C., Soga, K. & Simpson, B., 2012. A strain space soil model with evolving stiffness anisotropy. *Geotechnique*, 62(7), pp. 627-641.
- Geng, D. и други, 2021. Implicit numerical integration of highly nonlinear plasticity models. *Computers and Geotechnics*, Том 132, p. 103961.
- Gens, A. & Potts, D. M., 1988. Critical state models in computational geomechanics. *Engineering Computations*, 5(3), pp. 178-197.
- Georgiadis, K., Potts, D. M. & Zdravković, L., 2004. Modelling the shear strength of soils in the general stress space. *Computers and Geotechnics*, Том 31, pp. 357-364.
- Gu, C. et al., 2016. Deformation characteristics of overconsolidated clay sheared under constant and variable confining pressure. *Soils and Foundations*, 56(3), pp. 427-439.
- Hardin, B. O., 1978. *The nature of stress-strain behaviour of soils*. Pasadena, USA, Earthquake Engineering and Soil Dynamics.
- Hong, P. Y., Pereira, J. M., Cui, Y. J. & Tang, A. M., 2012. Explicit integration of a thermo-mechanical model for clays. *Computers and Geotechnics*, Том 46, pp. 12-25.
- Hsieh, Y.-M., Dang, P. H. & Lin, H.-D., 2017. How Small Strain Stiffness and Yield Surface Affect Undrained Excavation Prediction. *International Journal of Geomechanics*, 17(3).
- Huang, W. et al., 2006b. Finite-Element Parametric Study of the Consolidation Behavior of a Trial Embankment on Soft Clay. *International Journal of Geomechanics*, 6(5), pp. 328-341.
- Huang, W.-X., Wu, W., Sun, D.-A. & Sloan, S., 2006a. A simple hypoplastic model for normally consolidated clay. *Acta Geotechnica*, Том 1, pp. 15-27.
- Hu, C. & Liu, H., 2014. Implicit and explicit integration schemes in the anisotropic bounding surface plasticity model for cyclic behaviours of saturated clay. *Computers and Geotechnics*, Том 55, pp. 27-41.
- Iizuka, A. & Ohta, H., 1987. A determination procedure of input parameters in elasto-viscoplastic finite element analysis. *Soils and Foundations*, 27(3), pp. 71-87.
- Jardine, R. J., Potts, D. M., Fourie, A. B. & Burland, J. B., 1986. Studies of the influence of non-linear stress-strain characteristics in soil-structure interaction. *Geotechnique*, 36(3), pp. 377-396.
- Jocković, S., 2017. *Формулација и имплементација конститутивног модела за преkonsolidоване глине*. докторска дисертација: Грађевински факултет Универзитета у Београду.
- Jocković, S. & Vukićević, M., 2017. Bounding surface model for overconsolidated clays with new state parameter formulation of hardening rule. *Computers and Geotechnics*, Volume 83, pp. 16-29.
- Jocković, S. & Vukićević, M., 2018a. Validation and implementation of HASP constitutive model for overconsolidated clays. *Building Materials and Structures*, 61(1), pp. 91-110.

- Jocković, S. & Vukićević, M., 2018b. *Critical state constitutive model for overconsolidated clays—HASP model*. Skopje, North Macedonia, Proceedings of the XVI Danube—European Conference on Geotechnical Engineering.
- Jocković, S., Vukićević, M. & Obradović, N., 2021. *State parameter for soil behaviour prediction in undrained conditions*. Vrnjačka Banja, Serbia, s.n., pp. 525-536.
- Jovičić, V., Jurečić, N. & Vilhar, G., 2023. Modelling the structure and anisotropy of London clay using the SA_BRICK model. *Applied Sciences*, Том 13, p. 880.
- Kavvasas, M., 1982. *Non-linear consolidation around driven piles in clays*. PhD thesis: Massachusetts Institute of Technology.
- Kojić, M., 1996. The governing parameter method for implicit integration of viscoplastic constitutive relations for isotropic and orthotropic metals. *Computational Mechanics*, Том 19, pp. 49-57.
- Kojić, M. & Bathe, K.-J., 2003. *Inelastic Analysis of Solids and Structures*. 1st yп. Berlin: Springer-Verlag.
- Kojić, M., Slavković, R., Grujić, N. & Vukićević, M., 1994. Implicit stress integration algorithm for Modified Cam Clay material. *Teorijska i primenjena mehanika*, Том 20, pp. 95-119.
- Kojić, M. & Vukićević, M., 1992. *An Implicit procedure for stress integration in elastic-plastic deformation on the modified Cam-Clay model*. Barcelona, COMPLAST.
- Kojić, M. & Vukićević, M., 1998. *Implicit stress integration - the governing parameter method for bounding surface CS material*. Barcelona, Proceedings of Fifth International Conference on Computational Plasticity.
- Koskinen, M., Karstunen, M. & Wheeler, S., 2002. *Modelling destructuration and anisotropy of a soft natural clay*. Paris, 5th European Conference on Numerical Methods in Geotechnical Engineering.
- Lade, P. V., 2005. Overview of constitutive models for soil. У: J. A. Yamamuro & V. N. Kaliakin, уредници *ASCE Geotechnical Special Publication No. 128, Soil Constitutive Models: Evaluation, Selection, and Calibration*. Washington: ASCE, pp. 1-34.
- Lade, P. V. & Duncan, J. V., 1975. Elastoplastic stress-strain theory for cohesionless soil. *Journal of the Geotechnical Engineering Division*, 101(10), pp. 1037-1053.
- Lansivaara, T. & Nordal, S., 1998. *A soil model for the overconsolidated region of clays*. Udine, Proceedings of the Fourth European Conference on Numerical Methods in Geotechnical Engineering.
- Lazović, M. & Vukićević, M., 1997. *Конститутивни модели тла: I део (еластични модели тла)*. 1st ur. Београд: Грађевински факултет Београд.
- Ledesma, O., Manzanal, D. & Sfriso, A., 2021. Formulation and numerical implementation of a state parameter-based generalized plasticity model for mine tailings. *Computers and Geotechnics*, Том 135, p. 104158.

- Lehane, B. M. & Simpson, B., 2000. Modelling glacial till under triaxial conditions using BRICK soil model. *Canadian Geotechnical Journal*, Том 37, pp. 1078-1088.
- Lloret-Cabot, M. & Sheng, D., 2022. Assessing the accuracy and efficiency of different order implicit and explicit integration schemes. *Computers and Geotechnics*, Том 141, p. 104531.
- Lloret-Cabot, M., Sloan, S. W., Sheng, D. & Abbo, A. J., 2016. Error behaviour in explicit integration algorithms with automatic substepping. *International Journal for numerical methods in engineering*, 108(9), pp. 1030-1053.
- Lloret-Cabot, M., Wheeler, S. J., Gens, A. & Sloan, S. W., 2021. Numerical integration of an elasto-plastic critical state model for soils under unsaturated conditions. *Computers and Geotechnics*, Том 137, p. 104299.
- Maksimović, M., 2008. *Механика тла*. 4 ур. Београд: АГМ књига.
- Mašin, D., 2005. A hypoplastic constitutive model for clays. *International Journal for Numerical and Analytical Methods in Geomechanics*, Том 29, pp. 311-336.
- Matsuoka, H., 1976. On the significance of the "spatial mobilized plane". *Soils and Foundations*, 16(1), pp. 91-100.
- Matsuoka, H. & Nakai, T., 1974. *Stress-deformation and strength characteristics of soil under three different principal stresses*. Japan, Proceedings of the Japan Society of Civil Engineers.
- Matsuoka, H. & Sun, D., 2006. *The SMP Concept-based 3D Constitutive Models for Geomaterials*. 1st yp. London, UK: Taylor & Francis Group.
- Matsuoka, H., Yao, Y.-P. & Sun, D., 1999. The Cam-Clay models revised by the SMP criterion. *Soils and Foundations*, 39(1), pp. 81-95.
- Matsuoka, H., Yao, Y.-P. & Sun, D., 2005. SMP criterion-based unified constitutive model for geomaterials. У: J. A. Yamamuro & V. N. Kaliakin, уредници *Soil Constitutive Models: Evaluation, Selection, and Calibration*. Austin, Texas, United States: American Society of Civil Engineers, pp. 333-357.
- Moghadam, S. I., Taheri, E., Ahmadi, M. & Amiri, S. A. G., 2022. Unified bounding surface model for monotonic and cyclic behaviour of clay and sand. *Acta Geotechnica*, Volume 17, pp. 4359-4375.
- Mouratidis, A. & Magnan, J., 1983. *An anisotropic elastoplastic model with increase in strength through plastic deformation for the design calculation of structures on compressible soils*, Paris: Recherche Laboratoire Central des Ponts et Chaussées.
- Nakai, T. & Hinokio, M., 2004. A simple elastoplastic model for normally and over consolidated soils with unified material parameters. *Soils and Foundations*, 44(2), pp. 53-70.
- Nayak, G. C. & Zienkiewicz, O. C., 1972. Elastoplastic stress analysis: a generalization for various constitutive relations including strain softening. *Internationa Journal for Numerical Methods in Engineering*, Том 5, pp. 113-115.

- Nordal, S., Jostad, H. P., Kavli, A. & Grande, L., 1990. *A coulombian soil model applied to an offshore platform*. Rio De Janeiro, 12th International Conference on Soil Mechanics and Foundation Engineering: Offshore exploration and foundations.
- Obrzud, R. F. & Truty, A., 2018. *The Hardening soil model - A practical guidebook*, Switzerland: Z_Soil.PC 100701 report.
- Ortiz, M. & Popov, E. P., 1985. Accuracy and stability of integration algorithms for elastoplastic constitutive relations. *International journal for numerical methods in engineering*, 21(9), pp. 1561-1576.
- Ortiz, M. & Simo, J. C., 1986. An Analysis of a New Class of Integration Algorithms for Elastoplastic Constitutive Relations. *International Journal for Numerical Methods in Engineering*, Volume 23, pp. 353-366.
- Osman, A. S. & Bolton, M. D., 2005. Simple plasticity-based prediction of the undrained settlement of shallow circular foundations on clay. *Geotechnique*, 55(6), pp. 435-447.
- Owen, D. R. J. & Hinton, E., 1980. *Finite Elements in Plasticity: Theory and Practice*. 1st yp. Swansea, Wales: Pineridge Press.
- Pestana, J. M. & Whittle, A. J., 1999. Formulation of a unified constitutive model for clays and sands. *International Journal for Numerical and Analytical Methods in Geomechanics*, Volume 23, pp. 1215-1243.
- Petalas, A. L. & Dafalias, Y. F., 2019. Implicit integration of incrementally non-linear, zero-elastic range, bounding surface plasticity. *Computers and Geotechnics*, Tom 112, pp. 386-402.
- Potts, D. M. & Ganendra, D., 1992. *A comparison of solution strategies for non-linear finite element analysis of geotechnical problems*. Barcelona, Proceeding of the 3rd International Conference on Computational Plasticity.
- Potts, D. M. & Ganendra, D., 1994. An evaluation of substepping and implicit stress point algorithms. *Computer Methods in Applied Mechanics and Engineering*, Tom 119, pp. 341-354.
- Potts, D. M. & Zdravković, L., 1999. *Finite element analysis in geotechnical engineering: Theory*. 1st ed. London: Thomas Telford Publishing.
- Potts, D. & Zdravković, L., 2000. *Some pitfalls when using modified cam clay*, London, UK: Imperial College.
- Prevost, J.-H., 1977. Mathematical Modelling of Monotonic and Cyclic Undrained Clay Behaviour. *International Journal for Numerical and Analytical Methods in Geomechanics*, Volume 1, pp. 195-216.
- Puzrin, A. M. & Burland, J. B., 1998. Non-linear model of small-strain behaviour of soils. *Geotechnique*, 48(2), pp. 217-233.
- Roscoe, H. K., Schofield, N. A. & Wroth, P. C., 1958. On the yielding od soils. *Geotechnique*, Issue 8, pp. 22-53.

- Roscoe, K. H. & Burland, J. B., 1968. On the generalised stress-strain behaviour of 'wet' clay. У: J. Heyman & F. Leckie, уредници *Engineering plasticity*. Cambridge: Cambridge University Press, pp. 535-609.
- Roscoe, K. H. & Schofield, A. N., 1963. *Mechanical behaviour of an idealized 'Wet Clay'*. Wiesbaden, Proceedings of 2nd European Conference on Soil Mechanics, pp. 47-54.
- Rouainia, M. & Wood, M., 2001. Implicit numerical integration for a kinematic hardening soil plasticity model. *International Journal for Numerical and Analytical Methods in Geomechanics*, Том 25, pp. 1305-1325.
- Sangrey, D. A., Henkel, D. J. & Esrig, M. I., 1969. The effective stress response of a saturated clay soil to repeated loading. *Canadian Geotechnical Journal*, 6(3), pp. 241-252.
- Schadlich, B. & Schweiger, H. F., 2012. A multilaminate constitutive model accounting for anisotropic small strain stiffness. *International Journal for Numerical and Analytical Methods in Geomechanics*, 37(10), pp. 1337-1362.
- Schadlich, B. & Schweiger, H. F., 2014. Modelling the shear strength of overconsolidated clays with a Hvorslev surface. *Geotechnik*, 37(1), pp. 47-56.
- Schanz, T., Vermeer, P. A. & Bonnier, P. G., 1999. *The hardening soil model: Formulation and verification*. Rotterdam, Beyond 2000 in Computational Geotechnics - 10 Years of PLAXIS.
- Schofield, A. & Wroth, P., 1968. *Critical State Soil Mechanics*. 1st yp. Cambridge: McGraw-Hill.
- Sekiguchi, H. & Ohta, H., 1977. *Induced anisotropy and time dependency in clays*. Tokyo, Proceedings of the 9th ICSMFE.
- Sheng, D., Sloan, S. W. & Yu, H. S., 2000. Aspects of finite element implementation of critical state models. *Computational Mechanics*, Том 26, pp. 185-196.
- Simo, J. C. & Taylor, R. L., 1985. Consistent tangent operators for rate-independent elastoplasticity. *Computer methods in applied mechanics and engineering*, Том 48, pp. 101-118.
- Simo, J. C. & Taylor, R. L., 1986. A return mapping algorithm for plane stress elastoplasticity. *International Journal for Numerical Methods in Engineering*, 22(3), pp. 649-670.
- Simpson, B., 1992. Retaining structures: displacement and design. *Geotechnique*, 42(4), pp. 541-576.
- Simpson, B., O'Riordan, N. J. & Croft, D. D., 1979. A computer model for analysis of ground movements in London Clay. *Geotechnique*, 29(2), pp. 149-175.
- Sivasithamparam, N., Karstunen, M. & Bonnier, P., 2015. Modelling sreeep behaviour of anisotropic soft soils. *Computers and Geotechnics*, Том 69, pp. 46-57.
- Sloan, S. W., 1987. Substepping schemes for numerical integration of elastoplastic stress-strain relations. *International Journal for Numerical Methods in Engineering*, Том 24, pp. 893-911.

- Sloan, S. W., Abbo, A. J. & Sheng, D., 2001. Refined explicit integration of elastoplastic models with automatic error control. *Engineering Computations*, 18(1/2), pp. 121-194.
- Solowski, W. T. & Gallipoli, D., 2010a. Explicit stress integration with error control for the Barcelona Basic Model Part I: Algorithms formulations. *Computers and Geotechnics*, Том 37, pp. 59-67.
- Solowski, W. T. & Gallipoli, D., 2010b. Explicit stress integration with error control for the Barcelona Basic Model. Part II: Algorithms efficiency and accuracy. *Computers and Geotechnics*, Том 37, pp. 68-81.
- Solowski, W. T. и други, 2012. A comparative study of stress integration methods for the Barcelona Basic Model. *Computers and Geotechnics*, Том 44, pp. 22-33.
- Stallebrass, S. E., 1990. *Modelling the effect of recent stress history on the deformation of overconsolidated soils*. PhD thesis: The City University London.
- Stolle, D. F. E., Bonnier, P. G. & Vermeer, P. A., 1997. *A soft soil model and experiences with tow integration schemes*. Montreal, Numerical models in geomechanics: NUMOG VI.
- Surarak, C. et al., 2012. Stiffness and strength parameters for hardening soil model of soft and stiff Bangkok clays. *Soils and Foundations*, 52(4), pp. 682-697.
- Taborda, D. M. G., Potts, D. M. & Zdravković, L., 2016. On the assessment of energy dissipated through hysteresis in finite element analysis. *Computers and Geotechnics*, Volume 71, pp. 180-194.
- Taborda, D. M. & Zdravković, L., 2012. Application of a Hill-Climbing technique to the formulation of a new cyclic nonlinear elastic constitutive model. *Computers and Geotechnics*, Volume 43, pp. 80-91.
- Tavenas, F., Tremblay, M., Larouche, G. & Lerouei, I. S., 1986. *In situ measurement of permeability in soft clays*. Blacksburg, Va, USA, Proceedings of the ASCE Special Conference on Use of In-situ Test in Geotechnical Engineering.
- van Eekelen, H., 1980. Isotropic yield surfaces in three dimensions for use in soil mechanics. *International Journal for Numerical and Analytical Methods in Geomechanics*, 4(1), pp. 89-101.
- Vermeer, P. A. & Neher, H. P., 1999. *A soft soil model that accounts for creep*. Rotterdam, Beyond 2000 in Computatuion Geotechnics - 10 Years of Plaxis International.
- Vukićević, M., 2010. *Governing parameter method for implicit integration of modified Cam-Clay model, using the mean stress as the governing parameter*. Bratislava, Proceedings of XIV Danube-European Conference on Geotechnical Engineering, From research to design in European practice.
- Vukićević, M. & Jocković, S., 2018. *Governing parameter method for numerical integration of constitutive models for clays*. Porto, 9th European Conference on Numerical Methods in Geotechnical Engineering.

- Vukićević, M. & Jocković, S., 2019. *Конститутивни модели за тло II Еластопластични модели*. 1 ур. Београд: s.n.
- Vukićević, M. & Rakić, D., 2010. *The mean stress as the governing parameter in the implicit GPM stress integration of modified Cam-Clay model, part II*. Shanghai, Proceeding of 1st International Conference on Information Technology in Geo-Engineering.
- Wang, H., Li, L., Li, J. & Sun, D., 2022. A simple stress correction method for explicit integration algorithm of elastoplastic constitutive models and its application to advanced anisotropic S-CLAY1 model. *Computers and Geotechnics*, Том 148, p. 104817.
- Wang, S. & Wu, W., 2021. A simple hypoplastic model for overconsolidated clays. *Acta Geotechnica*, Том 16, pp. 21-29.
- Weifner, T. & Kolymbas, D., 2007. A hypoplastic model for clay and sand. *Acta geotechnica*, Том 2, pp. 103-112.
- Wei, S.-M., Yuan, L. & Cui, Z.-D., 2021. Application of closes point projection method to unified hardening model. *Computers and Geotechnics*, Том 133, p. 104064.
- Wheeler, S. J., Naatanen, A., Karstunen, M. & Lojander, M., 2003. An anisotropic elastoplastic model for soft clays. *Canadian Geotechnical Journal*, 40(2), pp. 403-418.
- Whittle, A. J., 1987. *A constitutive model for overconsolidated clays*. PhD thesis: Massachusetts Institute of Technology.
- Wilkins, M. L., 1963. *Calculation of elastic-plastic flow*, Livermore, California: Lawrence Radiation Laboratory, University of California.
- Wissman, J. W. & Hauck, C., 1983. Efficient elasto-plastic finite element analysis with higher order stress point algorithms. *Computers & Structures*, Том 17, pp. 89-95.
- Wood, D. M., 1995. *Kinematic hardening model for structured soil*. Rotterdam, Proceedings of Numerical Models in Geomechanics.
- Yamaguchi, E., 1993. *A comparative study of numerical methods for computing stress increments in elastic-plastic materials*. Hong Kong Proceedings of the Asia-Pacific Symposium on Advances in Plasticity and Its Applications, s.n.
- Yao, Y. P., Hou, W. & Zhou, A. N., 2009. UH model: three-dimensional unified hardening model for overconsolidated clays. *Geotechnique*, 59(5), pp. 451-469.
- Yao, Y. P., Sun, D. & Matsuoka, H., 2008. A unified constitutive model for both clay and sand with hardening parameter independent on stress path. *Computers and Geotechnics*, Том 35, pp. 210-222.
- Yao, Y.-P. & Wang, N.-D., 2014. Transformed Stress Method for Generaliyng Soil Constitutive Models. *Journal of Engineering Mechanics*, 140(3), pp. 614-629.
- Yu, H. S., 1998. CASM: A Unified State Parameter Model for Clay and Sand. *International Journal for Numerical and Analytical Methods in Geomechanics*, Volume 22, pp. 621-653.

Zdravković, L. и други, 2020. Finite-element modelling of laterally loaded piles in a stiff glacial clay till at Cowden. *Geotechnique*, 70(11), pp. 999-1013.

Zhao, J., Sheng, D., Rouainia, M. & Sloan, S. W., 2005. Explicit stress integration of complex soil models. *International Journal for Numerical and Analytical Methods in Geomechanics*, Том 29, pp. 1209-1229.

Zhao, L.-Y., Liu, L.-H. & Lai, Y.-M., 2023. An enhanced cutting plane algorithm of elastoplastic constitutive models for geomaterials. *Computers and Geotechnics*, Том 155, p. 105253.

Zhou, D., 2022. *Numerical Analysis of Double-O-Tube Shield Tunnelling*. Phd thesis: Imperial College London.

Zhou, X. и други, 2022. An open-source unconstrained stress updating algorithm for the modified Cam-clay model. *Computer Methods in Applied Mechanics and Engineering*, Том 390, p. 114356.

Прилог 1 Структура UDSM за PLAXIS

У оквиру овог Прилога дата је структура UDSM (User-defined Soil Model) у PLAXIS-у. UDSM омогућава корисницима PLAXIS-а да изврше имплементацију било ког конститутивног модела. UDSM треба да буде написан у FORTRAN-у и компајлиран у DLL формат (Dynamic Link Library). UDSM треба да врати PLAXIS-у напоне и параметре стања на крају инкремента уколико су познати напони, параметри стања и деформације на почетку инкремента, као и прираштаји деформација и времена.

UDSM треба да изврши шест задатака (Tasks) како би прорачун напона могао да се спроведе:

- Иницијализација параметара стања (Initialization of state variables).
- Прорачун напона у складу са конститутивним моделом (Calculation of constitutive stresses). У оквиру овог задатка примењује се алгоритам за интеграцију релација конститутивног модела.
- Прорачун конститутивне матрице за ефективне напоне (Creation of effective material stiffness matrix).
- Враћање PLAXIS-у број параметара стања (Return of the number of state variables).
- Враћање PLAXIS-у особина конститутивне матрице (Return matrix attributes).
- Прорачун еластичне конститутивне матрице (Creation of elastic material stiffness matrix)

UDSM се састоји од неколико субрутина (subroutine) у оквиру којих се поред претходно набројаних задатака извршавају и додатни задаци. Све те субрутине треба да буду заједно компајлиране у DLL. Те субрутине су:

- USRMOD.for
- USRLIB.for
- USRADDF.for
- OKMSGIVF.for
- USERMODEL.for (или било које друго име са екстензијом .for). Ова субрутина садржи алгоритам за нумеричку интеграцију релација конститутивног модела и може их бити више у оквиру једног UDSM (али свака субрутина мора имати своје јединствено име).
- Остале субрутине са екстензијом .for које су потребне кориснику за имплементацију модела

Субрутина USRMOD.for је главна субрутина и оквиру ње се позивају остале субрутине. Ова субрутина служи за комуникацију са PLAXIS-ом. Један UDSM може да садржи више различитих конститутивних модела и сваки од њих се посебно може позвати да се изврши у оквиру USRMOD.for. Ова могућност је коришћена у оквиру дисертације, где је један конститутивни модел имплементиран преко више различитих алгоритама за нумеричку интеграцију напона, тако да сваки алгоритам представља посебан модел. Број различитих модела у оквиру једног UDSM је ограничен на десет.

Субрутина USRLIB.for садржи помоћне функције и субрутине које служе за обављање разних рачунских операција и других инструкција које су потребне кориснику.

Субрутина USRADDF.for садржи помоћне субрутине које се позивају у USRMOD.for и враћају PLAXIS-у податке о броју различитих модела у DLL-у, имена тих модела, број, имена и јединице улазних параметара сваког модела, број, имена и јединице параметара стања сваког модела.

Субрутина OKMSGIVF.for садржи функцију за генерисање прозора са информацијом за корисника. Тај прозор се може позвати било где у програму и обавестити корисника током прорачуна са унапред спремљеном поруком (нпр. упозорење кориснику да је унео Poisson-ов коефицијент већи од 0.5).

Субрутина USERMODEL.for садржи алгоритам за интеграцију конститутивних напона и у оквиру које се извршава шест задатака набројаних на почетку овог Прилога. Ових субрутина може да буде до десет у једном DLL-у. Ова субрутина треба да почне са:

Subroutine USERMODEL (IDTask, iMod, IsUndr, iStep, iTer, iEl, Int, X, Y, Z, Time0, dTime, Props, Sig0, Swp0, StVar0, dEps, D, BulkW, Sig, Swp, StVar, ipl, nStat, NonSym, iStrsDep, iTimeDep, iTang, iAbort)

Променљиве које су у заглављу субрутине имају следеће значење:

- IDTask : садржи број задатка (Task) који треба да се изврши.
- iMod : број модела који се позива (може имати вредност од 1 до 10).
- IsUndr : = 0 за дрениране услове, = 1 за недрениране услове.
- iStep : број текућег инкремента.
- iTer : број текуће итерације.
- iEl : број текућег коначног елемента.
- Int : број текуће тачке нумеричке интеграције.
- X, Y, Z : координате текуће тачке нумеричке интеграције у глобалном координатном систему.
- Time0 : време на почетку инкремента.
- dTime : временски инкремент.
- Props : низ од 50 елемената који садржи улазне параметре модела.
- Sig0 : низ од 20 елемената који садржи ефективне напоне на почетку инкремента и још неке друге променљиве. Првих шест елемената низа су напони ($\sigma_x^0, \sigma_y^0, \sigma_z^0, \tau_{xy}^0, \tau_{yz}^0, \tau_{zx}^0$) на почетку инкремента у текућој тачки нумеричке интеграције.
- Swp0 : додатни порни притисци на почетку инкремента у текућој тачки нумеричке интеграције.
- StVar0 : низ од nStat елемената који садржи параметре стања на почетку инкремента у текућој тачки нумеричке интеграције.
- dEps : низ од 12 елемената који садржи прираштаје деформација и укупне деформације на почетку инкремента у текућој тачки нумеричке интеграције. Првих шест елемената су прираштаји деформација ($\Delta\varepsilon_x, \Delta\varepsilon_y, \Delta\varepsilon_z, \Delta\gamma_{xy}, \Delta\gamma_{yz}, \Delta\gamma_{zx}$), а других шест елемената су укупне деформације на почетку инкремента ($\varepsilon_x^0, \varepsilon_y^0, \varepsilon_z^0, \gamma_{xy}^0, \gamma_{yz}^0, \gamma_{zx}^0$).
- D : конститутивна матрица за ефективне напоне са 6x6 елемената.
- Bulk_W : запремински модул воде (за тип А недренирану анализу).

- Sig : низ од 6 елемената који треба да садржи напоне на крају инкремента ($\sigma'_x, \sigma'_y, \sigma'_z, \tau'_{xy}, \tau'_{yz}, \tau'_{zx}$).
- Swp : додатни порни притисци на крају инкремента у текућој тачки нумеричке интеграције.
- StVar : низ од nStat елемената који садржи параметре стања на крају инкремента у текућој тачки нумеричке интеграције.
- Ipl : индикатор типа пластичних деформација, 1 = Mohr-Coulomb (failure) point, 2 = Tension cut-off point, 3 = Cap hardening point, 4 = Cap friction point, 5 = Friction hardening point, 6 = Plotting liquefaction indicators.
- nStat : број параметара стања. У UDSM нема ограничења у броју параметара стања, односно у оквиру низова StVar0 и StVar могу се чувати било који подаци који су потребни кориснику.
- NonSym : индикатор да ли је конститутивна матрица симетрична (= 0) или несиметрична (= 1).
- iStrsDep : индикатор да ли је конститутивна матрица зависи од напона (= 1) или не (=0).
- iTimeDep : индикатор да ли конститутивна матрица зависи од времена (=1) или не (=0).
- iTang : индикатор да ли се користи пун Newton-Raphson-ов поступак (= 1) или модификовани (= 0).
- iPrjDir : име пројектног директоријума.
- iPrjLen : дужина имена пројектног директоријума.
- iAbort : индикатор који зауставља прорачун (= 1).

У оквиру задатка $IDTask = 1$ попуњава се низ који садржи параметре стања StVar0 на почетку прорачуна у једној фази. Овај низ може да садржи било које променљиве. То могу бити на пример параметри који описују где је тренутни положај површи течења, или напон преконсолидације и друго. Овај задатак се увек извршава на почетку сваке фазе прорачуна. Уколико је потребно да се вредности параметара стања пренесу из једне фазе прорачуна у другу (нпр. у едометарском опиту једна фаза може бити примарно оптерећивања, а друга фаза растерећивање), онда је потребно да се у оквиру кода предвиди да се параметри стања не рачунају на почетку нове фазе. У једној фази, на почетку једног инкремента параметри стања се читају из StVar0. За наредни инкремент параметри стања треба да се упишу у StVar. Уколико се параметри стања нису мењали у току инкремента онда се њихова вредност само преписује из StVar0 у StVar, а уколико су променили вредност, нова вредност се уписује у StVar. Када је активна наредба Reset State Variables, сви елементи StVar0 се постављају да су једнаки нули.

Задатак $IDTask = 2$ садржи алгоритам за прорачун напона на крају једног инкремента. На крају овог задатка треба да се главном програму врате попуњени низови Sig и StVar. У оквиру задатка $IDTask = 3$ се рачуна конститутивна матрица, која може бити еластична матрица или еласто-пластична матрица. Такође се у овом задатку рачуна и запремински модул воде Bulk_W који се за недренирану анализу додаје конститутивној матрици. У задатку $IDTask = 4$ се главном програму враћа колико вектори StVar0 и StVar имају елемената. У задатку $IDTask = 5$ се главном програму враћају особине конститутивне матрице (симетричност, зависност од напона, времена и да ли се користи пун или модификовани Newton-Raphson-ов поступак). У оквиру задатка

$IDTask = 6$ рачуна се еластична конститутивна матрица. Уколико се и у $IDTask = 3$ користи исто еластична матрица, онда ова два задатка могу да се обједине у један. UDSM користи исту променљиву за обе матрице у задацима 3 и 6, али их главни програм користи за различите задатке. Матрицу D из задатка 3 користи за формирање матрице крутости коначних елемената, а матрицу D из задатка 6 користи за прорачун глобалне грешке у инкрементално-итеративном поступку.

У наставку је дат рачунарски код у Fortran-у за имплементацију HASP и HASP-E модела.

Прилог 2 UDSM for PLAXIS – Fortran code for HASP model

У овом прилогу је дат рачунарски код у Fortran-у за имплементацију HASP модела у PLAXIS (2D и 3D). Садржај субрутина USRMOD.for, USRLIB.for, USRADD.F.for и OKMSGIVF.for је описан у Прилогу 1. Субрутина MYMODEL1.for садржи рачунарски код за имплементацију HASP модела преко имплицитног алгоритма методе водећег параметра. Субрутина MYMODEL2.for садржи рачунарски код за имплементацију HASP модела преко имплицитног алгоритма (Borja & Lee, 1990). Субрутина MYMODEL3.for садржи рачунарски код за имплементацију HASP модела преко експлицитног алгоритма – модификован Euler-ов поступак са аутоматском субинкрементацијом и контролом грешке. Субрутина MYMODEL4.for садржи рачунарски код за имплементацију HASP модела преко експлицитног алгоритма – Runge-Kutta-Dormand-Prince-ов поступак са аутоматском субинкрементацијом и контролом грешке.

Приликом читавања dll, различити алгоритми за имплементацију конститутивних релација су доступни следећим редом:

- iMod=1 : имплицитни алгоритам методе водећег параметра
- iMod=2 : имплицитни алгоритам (Borja & Lee, 1990)
- iMod=3 : експлицитни алгоритам – модификовани Euler-ов поступак са аутоматском субинкрементацијом и контролом грешке
- iMod=4 : експлицитни алгоритам - Runge-Kutta-Dormand-Prince-ов поступак са аутоматском субинкрементацијом и контролом грешке

Улазни параметри које корисник задаје су:

- Props(1) : параметар λ
- Props(2) : параметар κ
- Props(3) : параметар M
- Props(4) : Poisson-ов коефицијент μ
- Props(5) : иницијални коефицијент порозности e_0
- Props(6) : степен преконсолидације OCR
- Props(7) : параметар POP
- Props(8) : коефицијент притиска тла у стању мировања за нормално консолидовано тло K_0^{NC}
- Props(9) : толеранција за локалну грешку STOL (само за експлицитне алгоритме)

Као параметри стања, за сваку напонску тачку, чувају се следеће величине:

- StVar(1) : индикатор да ли је потребно поново срачунати параметре који описују иницијално стање (параметар Γ – израз (3-48)) на почетку нове фазе. Има вредност 0 ако се поново рачуна Γ (увек у иницијалној фази када се генерише почетно стање напона и када је активна наредба Reset State Variables) или вредност 1 када не треба да се поново рачуна Γ
- StVar(2) : параметар Γ
- StVar(3) : изотропни степен преконсолидације R

- StVar(4) : параметар стања за тренутну напонску тачку Ψ
- StVar(5) : параметар стања за имагинарну напонску тачку $\bar{\Psi}$
- StVar(6) : коефицијент ојачања ω
- StVar(7) : параметар ојачања за тренутну површ течења p'_0
- StVar(8) : параметар ојачања за граничну површ \bar{p}'_0
- StVar(9) : максимални претходни вертикални ефективни напон
- StVar(10) : тренутни степен преконсолидације, рачуна се као количник тренутног вертикалног ефективног напона и максималног претходног вертикалног ефективног напона
- StVar(11) : тангентни модул смицања G
- StVar(12) : контролни индикатор, указује који алгоритам је коришћен и има исту вредност као iMod
- StVar(13) : бројач укупних (успешних и неуспешних) субинкремената у претходном инкременту (само за експлицитне алгоритме)
- StVar(14) : бројач успешних субинкремената у претходном инкременту (само за експлицитне алгоритме)
- StVar(15) : достигнуто псеудо време T . Индикатор да ли је нанет целокупни инкремент (само за експлицитне алгоритме)
- StVar(16) : локална грешка након последњег субинкремента (само за експлицитне алгоритме)

USRMOD.for

```

Subroutine User_Mod ( IDTask, iMod, IsUndr,
*           iStep, iTer, iEl, Int,
*           X, Y, Z,
*           Time0, dTime,
*           Props, Sig0, Swp0, StVar0,
*           dEps, D, BulkW,
*           Sig, Swp, StVar, ipl,
*           nStat, NonSym, iStrsDep, iTimeDep, iTang,
*           iPrjDir, iPrjLen, iAbort )
!
! Purpose: User supplied soil model, basic formulation of HASP - HASP
!           Types : iMod=1 : implicit gpm with mean effective stress as governing
parameter
!                   iMod=2 : implicit Borja and Lee algorithm
!                   iMod=3 : explicit modified Euler with automatic subincrementation
and error control
!                   iMod=4 : explicit fifth-order accurate (Runge-Kutta-Dormand-
Prince) with automatic subincrementation and error control
!
! Depending on IDTask, 1 : Initialize state variables
!                      2 : calculate stresses,
!                      3 : calculate material stiffness matrix
!                      4 : return number of state variables
!                      5 : inquire matrix properties
!                      return switch for non-symmetric D-matrix
!                      stress/time dependent matrix
!                      6 : calculate elastic material stiffness matrix
! Arguments:
!           I/O Type
! IDTask   I   I   : see above
! iMod     I   I   : model number (1..10)
! IsUndr   I   I   : =1 for undrained, 0 otherwise
! iStep    I   I   : Global step number
! iter     I   I   : Global iteration number
! iel      I   I   : Global element number
! Int      I   I   : Global integration point number
! X        I   R   : X-Position of integration point
! Y        I   R   : Y-Position of integration point
! Z        I   R   : Z-Position of integration point
! Time0    I   R   : Time at start of step
! dTime    I   R   : Time increment
! Props    I   R()  : List with model parameters
! Sig0     I   R()  : Stresses at start of step
! Swp0     I   R   : Excess pore pressure start of step
! StVar0   I   R()  : State variable at start of step
! dEps     I   R()  : Strain increment
! D        I/O  R(,) : Material stiffness matrix
! BulkW    I/O  R   : Bulkmodulus for water (undrained only)
! Sig      0   R()  : Resulting stresses
! Swp      0   R   : Resulting excess pore pressure
! StVar    0   R()  : Resulting values state variables
! ipl      0   I   : Plasticity indicator
! nStat    0   I   : Number of state variables
! NonSym   0   I   : Non-Symmetric D-matrix ?
! iStrsDep 0   I   : =1 for stress dependent D-matrix
! iTimeDep 0   I   : =1 for time dependent D-matrix
! iTang    0   I   : =1 for tangent matrix
! iAbort   0   I   : =1 to force stopping of calculation
!

```

```

    Implicit Double Precision (A-H, O-Z)
!
Dimension Props(*), Sig0(*), StVar0(*), dEps(*), D(6,6),
*      Sig(*), StVar(*), iPrjDir(*)

Data iounit / 0 /
Save iounit

!
!--- Local variables
!
Character*100 BaseName

!DEC$ ATTRIBUTES DLLEExport, StdCall, reference :: User_Mod

BaseName = 'HASP'
! Possibly open a file for debugging purposes
If (iounit.Eq.0) Then

    Call Open_Dbg_File( iPrjDir, iPrjLen, BaseName )

    Write(1,*)'File 1 opened: ', Trim(baseName)
    ! maybe write some more info on version to debug file ?
    write(1,*)'Compiled : ',__DATE__, ' ', __TIME__
!DEC$ IF DEFINED(_X86_)
    ! this 32-bit ??
    Write(1,1050)'IF32', __INTEL_COMPILER
!DEC$ ELSE
    ! this 64-bit ??
    Write(1,1050)'IF64', __INTEL_COMPILER
!DEC$ ENDIF
1050 format ( 1X,A,1x,I0 )
    iounit = 1
    Call WriVec(1, 'Props', Props, 50)
    Call Flush(1)
End If

Call WriIvl( -1, 'iounit', iounit )
Call WriIvl( -1, 'IDTask', IDTask )
Select Case (iMod)
    Case (1) ! gpm integration with mean effective stress as governing parameter
        Call MyModel1( IDTask, iMod, IsUndr, iStep, iTer, iEl, Int,
*           X, Y, Z, Time0, dTime,
*           Props, Sig0, Swp0, StVar0,
*           dEps, D, BulkW, Sig, Swp, StVar, ipl,
*           nStat, NonSym, iStrsDep, iTimeDep, iTang,
*           iAbort )
    Case (2) ! implicit Borja and Lee algorithm
        Call MyModel2( IDTask, iMod, IsUndr, iStep, iTer, iEl, Int,
*           X, Y, Z, Time0, dTime,
*           Props, Sig0, Swp0, StVar0,
*           dEps, D, BulkW, Sig, Swp, StVar, ipl,
*           nStat, NonSym, iStrsDep, iTimeDep, iTang,
*           iAbort )
    Case (3) ! explicit modified Euler with automatic subincrementation and
error control
        Call MyModel3( IDTask, iMod, IsUndr, iStep, iTer, iEl, Int,
*           X, Y, Z, Time0, dTime,
*           Props, Sig0, Swp0, StVar0,
*           dEps, D, BulkW, Sig, Swp, StVar, ipl,
*           nStat, NonSym, iStrsDep, iTimeDep, iTang,
*           iAbort )

```

```

Case (4) ! explicit fifth-order accurate (Runge-Kutta-Dormand-Prince) with
automatic subincrementation and error control

```

```

* Call MyModel4( IDTask, iMod, IsUndr, iStep, iTer, iEl, Int,
* X, Y, Z, Time0, dTime,
* Props, Sig0, Swp0, StVar0,
* dEps, D, BulkW, Sig, Swp, StVar, ipl,
* nStat, NonSym, iStrsDep, iTimeDep, iTang,
* iAbort )

```

```

Case Default

```

```

Write(1,*) 'invalid model number in UsrMod', iMod

```

```

Write(1,*) 'IDTask: ',IDTask

```

```

Stop 'invalid model number in UsrMod'

```

```

iAbort=1

```

```

Return

```

```

End Select ! iMod

```

```

If (IDTask .Eq. 5.And.iel+int.Eq.2) Then

```

```

Write(1,*)'nStat : ',nStat

```

```

Write(1,*)'NonSym : ',NonSym

```

```

Write(1,*)'StrsDep : ',iStrsDep

```

```

Write(1,*)'TimeDep : ',iTimeDep

```

```

Write(1,*)'Tangent : ',iTang

```

```

End If

```

```

If (IDTask == -333 .And. iel+int == -1234) Then

```

```

! Write(1,*)'IDTask: ',IDTask,' iStep,iTer',iStep,iTer

```

```

! Call Flush(1)

```

```

End If

```

```

Call WriIvl( -1, 'IDTask end',IDTask )

```

```

Return

```

```

End ! User_Mod

```

```

! *****

```

```

Subroutine Open_Dbg_File( iPrjDir, iPrjLen, BaseName )

```

```

Implicit None

```

```

Integer, intent(in) :: iPrjLen, iPrjDir(*)

```

```

Character*(*), intent(in):: BaseName

```

```

Character*255 PrjDir, Dbg_Name

```

```

Integer i, nErr, ios

```

```

PrjDir=' '

```

```

Do i=1,iPrjLen

```

```

PrjDir(i:i) = Char( iPrjDir(i) )

```

```

End Do

```

```

Dbg_Name=PrjDir(:iPrjLen)//'data.'//trim(BaseName)//'.rr0'

```

```

nErr=0

```

```

1 Continue

```

```

Open( Unit= 1, File= Dbg_Name,iostat=ios)

```

```

If (ios.Eq.0) Close(Unit=1,Status='delete',iostat=ios)

```

```

If (ios.Ne.0) Then

```

```

!
! in case of error try ...udsmex1 or udsmex2 or ..
!

```

```

nErr=nErr+1

```

```

Dbg_Name=PrjDir(:iPrjLen)//'data.'//

```

```

* trim(BaseName)//char(48+nErr)//'.rr0'

```

```

If (nErr.Lt.10) Goto 1

```



```

    End If

    Open( Unit= 1, File= Dbg_Name,blocksize=4096)

    End Subroutine Open_Dbg_File

! *****
!
! Interfaces to routines to report parameter names, counts etc.
!

Subroutine Add_Str_Length( aString )
Implicit None
Character*(*) aString
Character *255 tString
Integer Lt          ! length of incoming string

! routine should add the length of the string as the first character

tString = aString
Lt      = Len_Trim(tString)
aString = Char(Lt) // tString(1:Lt)

End Subroutine Add_Str_Length

Subroutine GetModelCount( nMod )
!
! Return the maximum model number (nMod) in this DLL
!
Implicit None
Integer (Kind=4) nMod

!DEC$ ATTRIBUTES DLLEExport, StdCall, reference :: GetModelCount

Call Get_Model_Count( nMod )

Return
End ! GetModelCount

Subroutine GetModelName( iMod , ModelName )
!
! Return the name of the different models
!
Implicit None
Integer iMod
Character (Len= * ) ModelName
Character (Len=255) tName
!DEC$ ATTRIBUTES DLLEExport, StdCall, reference :: GetModelName

Call Get_Model_Name( iMod , ModelName )
Call Add_Str_Length( ModelName )

End ! GetModelName

Subroutine GetParamCount( iMod , nParam )
!
! Return the number of parameters of the different models
!
Implicit None
Integer iMod, nParam

```

```

!DEC$ ATTRIBUTES DLLEExport, StdCall, reference  :: GetParamCount
Call Get_Param_Count( iMod , nParam )

End ! GetParamCount

Subroutine GetParamName( iMod , iParam, ParamName )
!
! Return the parameters name of the different models
!
Implicit None
Integer iMod, iParam
Character (Len=255) ParamName, Units

!DEC$ ATTRIBUTES DLLEExport, StdCall, reference  :: GetParamName

Call GetParamAndUnit(iMod,iParam,ParamName,Units)
Call Add_Str_Length( ParamName )

End ! GetParamName

Subroutine GetParamUnit( iMod , iParam, Units )
!
! Return the units of the different parameters of the different models
!
Implicit None
Integer iMod, iParam
Character (Len=255) ParamName, Units

!DEC$ ATTRIBUTES DLLEExport, StdCall, reference  :: GetParamUnit

Call GetParamAndUnit(iMod,iParam,ParamName,Units)
Call Add_Str_Length( Units )

End ! GetParamUnit

Subroutine GetStateVarCount( iMod , nVar )
!
! Return the number of state variables of the different models
!
Implicit None
Integer iMod, nVar

!DEC$ ATTRIBUTES DLLEExport, StdCall, reference  :: GetStateVarCount

Call Get_StateVar_Count( iMod , nVar )

End ! GetStateVarCount

Subroutine GetStateVarName( iMod , iVar, Name )
!
! Return the name of the different state variables
! of the different models
!
Implicit None
Integer iMod, iVar
Character (Len=255) Name, Unit

!DEC$ ATTRIBUTES DLLEExport, StdCall, reference  :: GetStateVarName

Call GetStateVarNameAndUnit( iMod , iVar, Name, Unit )
Call Add_Str_Length( Name )

```

```

End ! GetStateVarName

Subroutine GetStateVarUnit( iMod , iVar, Unit )
!
! Return the units of the different state variables of the different models
!
Implicit None
Integer iMod, iVar
Character (Len=255) Name, Unit

!DEC$ ATTRIBUTES DLLEExport, StdCall, reference :: GetStateVarUnit

Call GetStateVarNameAndUnit( iMod , iVar, Name, Unit )
Call Add_Str_Length( Unit )

End ! GetStateVarUnit

```

USRLIB.for

```
Subroutine MZEROR(R,K)
C
C*****
C
C   Function: To make a real array R with dimension K to zero
C
C*****
C
C   Implicit Double Precision (A-H,O-Z)
C   Dimension R(*)

C   Do J=1,K
C     R(J) = 0.0D0
C   End Do

C   Return
C   End
Subroutine COPYRVEC(R1,R2,K)
C
C*****
C
C   Function: To copy a Double array R1 with Dimension K to R2
C
C*****
C
C   Implicit Double Precision (A-H,O-Z)
C   Dimension R1(*),R2(*)

C   Do J=1,K
C     R2(J)=R1(J)
C   End Do

C   Return
C   End
C*****
Subroutine MatVec(xMat,IM,Vec,N,VecR)
C*****
C
C   Calculate VecR = xMat*Vec
C
C I  xMat  : (Square) Matrix (IM,*)
C I  Vec   : Vector
C I  N     : Number of rows/columns
C O  VecR  : Resulting vector
C
C*****
C   Implicit Double Precision (A-H,O-Z)
C   Dimension xMat(IM,*),Vec(*),VecR(*)
C*****
C   Do I=1,N
C     X=0
C     Do J=1,N
C       X=X+xMat(I,J)*Vec(J)
C     End Do
C     VecR(I)=X
C   End Do
C   Return
C   End    ! Subroutine MatVec
```

```

C*****
Subroutine AddVec(Vec1,Vec2,R1,R2,N,VecR)
C*****
C
C   Calculate VecR() = R1*Vec1()+R2*Vec2()
C
C I  Vec1,
C I  Vec2  : Vectors
C I  R1,R2 : Multipliers
C I  N     : Number of rows
C O  VecR  : Resulting vector
C
C*****
Implicit Double Precision (A-H,O-Z)
Dimension Vec1(*),Vec2(*),VecR(*)
C*****
Do I=1,N
  X=R1*Vec1(I)+R2*Vec2(I)
  VecR(I)=X
End Do
Return
End      ! Subroutine AddVec
C
C*****
Double Precision Function DInProd(A,B,N)
C*****
C
C   Returns the Inproduct of two vectors
C
C I  A,B  : Two vectors
C I  N    : Used length of vectors
C*****
Implicit Double Precision (A-H,O-Z)
Dimension A(*),B(*)
C*****

X = 0
Do I=1,N
  X = X + A(I)*B(I)
End Do
DInProd = X
Return
End      ! Function DInProd
C*****
subroutine CalculateStressInvariants(Stress,p,q,eta,theta)
C*****
! Stress(6)      : Array of stresses
! p              : Mean effective stress
! q              : Deviatoric stress
! eta            : Stress ratio
! theta          : Lodes angle
!
! ---Local variables---
!
! D3              : Third invariant of deviator stress
! Pie             : Value of Pi
!
Implicit Double Precision (A-H,O-Z)
Dimension Stress(6)
Parameter(
*Pie=3.14159 26535 89793 23846 26433 83279 50288 41971 69399d0)
p=(Stress(1)+Stress(2)+Stress(3))/3.0

```

```

if (p.le.0.0) p=0.001
q=sqrt(3.0*(1.0/6.0*((Stress(1)-Stress(2))**2.0+(Stress(2)-
*Stress(3))**2.0+(Stress(3)-Stress(1))**2.0)+Stress(4)**2.0+
*Stress(5)**2.0+Stress(6)**2.0))
eta=q/p
D3=(Stress(1)-p)*(Stress(2)-p)*(Stress(3)-p)+2.0*Stress(4)*
*Stress(5)*Stress(6)-(Stress(1)-p)*Stress(5)**2.0-(Stress(2)-
*p)*Stress(6)**2.0-(Stress(3)-p)*Stress(4)**2.0
theta=-1.0/3.0*asin(27/2*D3/q**3.0)
if (isnan(theta)) theta=-Pie/6.0
if (theta.gt.Pie/6.0) theta=Pie/6.0
if (theta.lt.-Pie/6.0) theta=-Pie/6.0
return
end ! CalculateStressInvariants
C*****
subroutine CalculateStressDeviator(Stress,p,Stressd)
C*****
! Stress(6)      : Array of stresses
! p              : Mean effective stress
! Stressd(6)     : Array of deviatoric stress
!
Implicit Double Precision (A-H,O-Z)
Dimension Stress(6), Stressd(6)
do i=1,3
    Stressd(i)=Stress(i)-p
    Stressd(i+3)=Stress(i+3)
end do
return
end ! CalculateStressDeviator
C*****
subroutine CalculateVoidRatio(Strain,e0,e)
C*****
! Strain(6)      : Array of strains
! e0             : Initial void ratio
! e              : Current void ratio
!
! ---Local variables---
!
! Epsv           : Current volumetric strain
!
Implicit Double Precision (A-H,O-Z)
Dimension Strain(6)
Epsv=Strain(1)+Strain(2)+Strain(3)
e=(1+e0)*exp(-Epsv)-1
if (e.lt.0.0) e=0.0
if (e.gt.1000) e=1000
return
end ! CalculateVoidRatio
C*****
subroutine CalculateStrainInvariants(Strain,EpsV,Epsq,ShearStrain)
C*****
! Strain(6)      : Array of strains
! EpsV           : Volumetric strain
! Epsq           : Deviatoric strain
! ShearStrain    : Shear strain invariant
!
Implicit Double Precision (A-H,O-Z)
Dimension Strain(6)
EpsV=Strain(1)+Strain(2)+Strain(3)
Epsq=sqrt(2.0/3.0*((Strain(1)-EpsV/3.0)**2.0+(Strain(2)-EpsV/3.0)
**2.0+(Strain(3)-EpsV/3.0)**2.0+1.0/2.0*(Strain(4)**2.0+
*Strain(5)**2.0+Strain(6)**2.0))

```

```

ShearStrain=3.0/2.0*Epsq
return
end ! CalculateStrainInvariants
C*****
subroutine CalculateStrainDeviator(Strain,EpsV,Straind)
C*****
! Strain(6)      : Array of strains
! EpsV          : Volumetric strain
! Straind(6)    : Array of deviatoric strain
!
Implicit Double Precision (A-H,O-Z)
Dimension Strain(6), Straind(6)
do i=1,3
  Straind(i)=Strain(i)-EpsV/3.0
  Straind(i+3)=Strain(i+3)/2.0
end do
return
end ! CalculateStrainDeviator
C*****
subroutine CalculateElasticModuli(e,p,xk,xNu,Bulk,G)
C*****
! e              : Void ratio
! p              : Mean effective stress
! xk             : Kappa, slope of URL
! xNu           : Poissons ratio
! Bulk          : Bulk modulus
! G             : Shear modulus
!
Implicit Double Precision (A-H,O-Z)
Bulk=(1.0+e)*p/xk
If (Bulk.lt.1.0) Bulk=1.0
G=3.0*(1.0-2.0*xNu)*Bulk/(2.0*(1.0+xNu))
return
end ! CalculateStrainDeviator
C*****
subroutine CalculateOmega(xl,xk,xM,Gama,p,q,e,omegalim,Rmax,Rmin,
*omega,R,PSIOC,PSINC)
C*****
! xl            : Lambda, slope oc ICL
! xk            : Kappa, slope of URL
! xM            : M, slope of CSL
! Gama          : Specific volume at CSL for p'=1 kPa
! p             : Mean effective stress
! q             : Deviator stress
! e             : Void ratio
! omegalim     : Limiting value of omega in order to avoid numerical
instability
! Rmax         : Max value of R in order to avoid numerical instability
! Rmin         : Min value of R in order to avoid non-possible values
! omega        : Hardening coefficient
! R            : R, isotropic OCR
! PSIOC        : State parameter for OC
! PSINC        : State parameter for NC
!
Implicit Double Precision (A-H,O-Z)
eta=q/p
PSIOC=1.0+e+xl*log(p)-Gama
PSINC=(xl-xk)*log(2.0*xM**2.0/(xM**2.0+eta**2.0))
R=exp((PSINC-PSIOC)/(xl-xk))
If (R.gt.Rmax) then
  PSIOC=PSINC-(xl-xk)*log(Rmax)
  R=Rmax

```

```

end if
if (R.lt.Rmin) then
  PSIOC=PSINC-(xl-xk)*log(Rmin)
  R=Rmin
end if
omega=(1.0+(PSINC-PSIOC)/PSINC)*R
If (omega.gt.omegalim) omega=omegalim
If (omega.lt.-omegalim) omega=-omegalim
if (omega.gt.0.0.and.omega.lt.0.0001) omega=0.0001
if (omega.gt.-0.0001.and.omega.lt.0.0) omega=-0.0001
return
end ! CalculateOmega
C*****
subroutine CalculateGMPYield(xl,xk,xM,et,ett,G,pt,qt,p0t,dEpsq,
dEpsd,Sdt,CStol,omega,ptt,f)
C*****
! xl          : Slope of ICL
! xk          : Slope of URL
! xM          : Slope of CSL
! et          : Void ratio at start of step
! ett         : Void ratio at end of step
! G           : Shear modulus
! pt          : Mean effective stress at start of step
! qt          : Shear stress at start of step
! p0t         : Hardening parameter at start of step
! dEpsq       : Increment of deviatoric strain
! dEpsd(6)    : Array of deviatoric strain increments
! Sdt(6)      : Array of deviatoric stresses at the end of step
! CStol       : Tolerance for assessing whether the critical state is reached
! omega       : Hardening coefficient
! ptt         : Mean effective stress at the end of step
! f           : Value of yield function for governing parameter ptt
!
! ---Local variables---
!
! Sdtt(6)     : Array of deviatoric stresses at the end of increment
! Stt(6)      : Array of stresses at the end of step
! p0tt        : Hardening parameter at the end of step
! qtt         : Shear stress at the end of step
! dq          : increment of shear stress
! dEpsqp      : Increment of plastic shear strain
! dlam        : Plastic multiplier
! dEpsvp      : Increment of plastic volumetric strain
!
Implicit Double Precision (A-H,O-Z)
Dimension dEpsd(6), Sdt(6), Sdtt(6), Stt(6)
p0tt=p0t*exp((et-ett)*omega/(xl-xk))*(pt/ptt)**(xk*omega/(xl-xk))
if (p0tt.gt.huge(1.0d0)) p0tt=2*pt ! check if p0tt is bigger than max possible
real number
if (isnan(p0tt)) p0tt=p0t ! check if p0tt is a number
if (p0tt.lt.tiny(1.0d0)) p0tt=0.001 ! check if p0tt is not too small
if (abs(2*ptt-p0tt).lt.CStol) then ! check if critical state is reached
  qtt=xM*ptt
  dq=qtt-qt
  dEpsqp=(dEpsq-dq/3.0/G)/omega
  dlam=xM*dEpsqp/2.0/ptt
else
  dEpsvp=(xl-xk)/(1.0+et)*log(p0tt/p0t)/omega
  dlam=dEpsvp/(2*ptt-p0tt)
end if
do i=1,6
  Sdtt(i)=(dEpsd(i)+Sdt(i)/2.0/G)/(3.0*dlam/xM**2.0+1.0/2.0/G)

```



```

end do
do i=1,3
  Stt(i)=Sdtt(i)+ptt
  Stt(i+3)=Sdtt(i+3)
end do
qtt=sqrt(3.0*(1.0/6.0*((Stt(1)-Stt(2))**2.0+(Stt(2)-Stt(3))**2.0+
*(Stt(3)-Stt(1))**2.0)+Stt(4)**2.0+Stt(5)**2.0+Stt(6)**2.0))
f=qtt**2.0/xM**2.0+ptt*(ptt-p0tt)
return
end ! CalculateGMPYield
C*****
subroutine CalculateGMPStress(xl,xk,xM,et,ett,G,pt,qt,p0t,dEpsq,
*dEpsd,Sdt,CStol,omega,ptt,dlam,p0tt,Stt)
C*****
! xl      : Slope of ICL
! xk      : Slope of URL
! xM      : Slope of CSL
! et      : Void ratio at start of step
! ett     : Void ratio at end of step
! G       : Shear modulus
! pt      : Mean effective stress at start of step
! qt      : Shear stress at start of step
! p0t     : Hardening parameter at start of step
! dEpsq   : Increment of deviatoric strain
! dEpsd(6) : Array of deviatoric strain increments
! Sdt(6)  : Array of deviatoric stresses at the end of step
! CStol   : Tolerance for assessing whether the critical state is reached
! omega   : Hardening coefficient
! ptt     : Mean effective stress at the end of step
! dlam    : Plastic multiplier
! p0tt    : Hardening parameter at the end of step
! Stt(6)  : Array of stresses at the end of step
!
! ---Local variables---
!
! Sdtt(6) : Array of deviatoric stresses at the end of increment
! qtt     : Shear stress at the end of step
! dq      : increment of shear stress
! dEpsqp  : Increment of plastic shear strain
! dEpsvp  : Increment of plastic volumetric strain
! f       : Value of yield function for governing parameter ptt
!
Implicit Double Precision (A-H,O-Z)
Dimension dEpsd(6), Sdt(6), Sdtt(6), Stt(6)
p0tt=p0t*exp((et-ett)*omega/(xl-xk))*(pt/ptt)**(xk*omega/(xl-xk))
if (p0tt.gt.huge(1.0d0)) p0tt=2*pt ! check if p0tt is bigger then max possible
real number
if (isnan(p0tt)) p0tt=p0t ! check if p0tt is a number
if (p0tt.lt.tiny(1.0d0)) p0tt=0.001 ! check if p0tt is not too small
if (abs(2*ptt-p0tt).lt.CStol) then ! check if critical state is reached
  qtt=xM*ptt
  dq=qtt-qt
  dEpsqp=(dEpsq-dq/3.0/G)/omega
  dlam=xM*dEpsqp/2.0/ptt
  dEpsvp=0.0
else
  dEpsvp=(xl-xk)/(1+et)*log(p0tt/p0t)/omega
  dlam=dEpsvp/(2*ptt-p0tt)
end if
do i=1,6
  Sdtt(i)=(dEpsd(i)+Sdt(i)/2.0/G)/(3.0*dlam/xM**2.0+1.0/2.0/G)
end do

```

```

do i=1,3
  Stt(i)=Sdtt(i)+ptt
  Stt(i+3)=Sdtt(i+3)
end do
qtt=sqrt(3.0*(1.0/6.0*((Stt(1)-Stt(2))**2.0+(Stt(2)-Stt(3))**2.0+
*(Stt(3)-Stt(1))**2.0)+Stt(4)**2.0+Stt(5)**2.0+Stt(6)**2.0))
f=qtt**2.0/xM**2.0+ptt*(ptt-p0tt)
eta=qtt/ptt
return
end ! CalculateGMPPStress
C*****
subroutine CalculateVectorA(p,p0,xM,Stress,Va)
C*****
! p          : Mean effective stress
! p0         : Hardening parameter
! xM         : Slope of CSL
! Stress(6)  : Array of stresses
! Va(6)      : Array of yield function derivatives
!
Implicit Double Precision (A-H,0-Z)
Dimension Stress(6), Va(6)
Va(1)=(2.0*p-p0)/3.0+3.0*(Stress(1)-p)/xM**2.0
Va(2)=(2.0*p-p0)/3.0+3.0*(Stress(2)-p)/xM**2.0
Va(3)=(2.0*p-p0)/3.0+3.0*(Stress(3)-p)/xM**2.0
Va(4)=6.0*Stress(4)/xM**2.0
Va(5)=6.0*Stress(5)/xM**2.0
Va(6)=6.0*Stress(6)/xM**2.0
return
end ! CalculateVectorA
C*****
subroutine CalculateDeMatrix(Bulk,G,De)
C*****
! K          : Bulk modulus
! G          : Shear modulus
! De(6)      : Elastic De matrix
! Stress(6)  : Array of stresses
! Va(6)      : Array of yield function derivatives
!
! ---Local variables---
!
! D1,D2,D3   : Auxilliary variables for filling of De matrix
!
Implicit Double Precision (A-H,0-Z)
Dimension De(6,6)
D1=Bulk+4.0*G/3.0
D2=Bulk-2.0*G/3.0
D3=G
Call MZeroR(De,36)
De(1,1)=D1
De(2,2)=D1
De(3,3)=D1
De(1,2)=D2
De(1,3)=D2
De(2,1)=D2
De(2,3)=D2
De(3,1)=D2
De(3,2)=D2
De(4,4)=D3
De(5,5)=D3
De(6,6)=D3
return
end ! CalculateDeMatrix

```

USRADDF.for

```
! Subroutines in this file:
!
! Subroutine GetModelCount( nMod )
! Subroutine GetModelName ( iMod , ModelName )
! Subroutine GetParamCount( iMod , nParam )
! Subroutine GetParamName ( iMod , iParam, ParamName )
! Subroutine GetParamUnit ( iMod , iParam, Units )
! Subroutine GetStateVarCount( iMod , nVar )
! Subroutine GetStateVarName ( iMod , iVar, Name )
! Subroutine GetStateVarUnit ( iMod , iVar, Unit )
!
! Local:
! Subroutine GetParamAndUnit( iMod , iParam, ParamName, Units )
! Subroutine GetStateVarNameAndUnit( iMod , iVar, Name, Unit )
```

```
Subroutine Get_Model_Count(nMod)
!
! Return the maximum model number (iMod) in this DLL
!
Integer (Kind=4) nMod

nMod = 4 ! Maximum model number (iMod) in current DLL

Return
End ! GetModelCount

Subroutine Get_Model_Name( iMod , ModelName )
!
! Return the name of the different models
!
Integer iMod
Character (Len= * ) ModelName

Select Case (iMod)
Case (1)
  ModelName = 'GPM P'
Case (2)
  ModelName = 'Borja and Lee'
Case (3)
  ModelName = 'modEuler'
Case (4)
  ModelName = 'RKDP'
Case Default
  ModelName = 'not in DLL'
End Select

Return
End ! Get_Model_Name

Subroutine Get_Param_Count( iMod , nParam )
!
! Return the number of parameters of the different models
!

Select Case (iMod)
Case ( 1 )
```

```

        nParam = 8
    Case ( 2 )
        nParam = 8
    Case ( 3 )
        nParam = 9
    Case ( 4 )
        nParam = 9
    Case Default
        nParam = 0
End Select
Return
End ! Get_Param_Count
Subroutine GetParamAndUnit( iMod , iParam, ParamName, Units )
!
! Return the parameters name and units of the different models
!
! Units: use F          for force unit
!             L          for length unit
!             T          for time unit
!             %          for percentage
!             DEG        for degrees
!             TEMP       for temperature
!             TEMPINKELVIN for temperature in Kelvin
!
Character (Len=255) ParamName, Units, tName
Select Case (iMod)
Case (1)
! ModName = 'GPM P'
Select Case (iParam)
Case (1)
ParamName = 'lambda' ; Units = '-'
Case (2)
ParamName = 'kappa' ; Units = '-'
Case (3)
ParamName = 'M' ; Units = '-'
Case (4)
ParamName = 'nu' ; Units = '-'
Case (5)
ParamName = 'evoid0' ; Units = '-'
Case (6)
ParamName = 'OCR' ; Units = '-'
Case (7)
ParamName = 'POP' ; Units = 'F/L^2#'
Case (8)
ParamName = 'K0NC' ; Units = '-'
Case Default
ParamName = '???' ; Units = '???'
End Select
Case (2)
! ModName = 'Borja and Lee'
Select Case (iParam)
Case (1)
ParamName = 'lambda' ; Units = '-'
Case (2)
ParamName = 'kappa' ; Units = '-'
Case (3)
ParamName = 'M' ; Units = '-'
Case (4)
ParamName = 'nu' ; Units = '-'
Case (5)
ParamName = 'evoid0' ; Units = '-'
Case (6)

```

```

        ParamName = 'OCR'      ; Units    = '-'
    Case (7)
        ParamName = 'POP'      ; Units    = 'F/L^2#'
    Case (8)
        ParamName = 'K0NC'     ; Units    = '-'
    Case Default
        ParamName = '???'      ; Units    = '???'
End Select
Case (3)
! ModName = 'modEuler'
Select Case (iParam)
    Case (1)
        ParamName = 'lambda'   ; Units    = '-'
    Case (2)
        ParamName = 'kappa'    ; Units    = '-'
    Case (3)
        ParamName = 'M'        ; Units    = '-'
    Case (4)
        ParamName = 'nu'       ; Units    = '-'
    Case (5)
        ParamName = 'evoid0'   ; Units    = '-'
    Case (6)
        ParamName = 'OCR'      ; Units    = '-'
    Case (7)
        ParamName = 'POP'      ; Units    = 'F/L^2#'
    Case (8)
        ParamName = 'K0NC'     ; Units    = '-'
    Case (9)
        ParamName = 'STOL'     ; Units    = '-'
    Case Default
        ParamName = '???'      ; Units    = '???'
End Select
Case (4)
! ModName = 'RKDP'
Select Case (iParam)
    Case (1)
        ParamName = 'lambda'   ; Units    = '-'
    Case (2)
        ParamName = 'kappa'    ; Units    = '-'
    Case (3)
        ParamName = 'M'        ; Units    = '-'
    Case (4)
        ParamName = 'nu'       ; Units    = '-'
    Case (5)
        ParamName = 'evoid0'   ; Units    = '-'
    Case (6)
        ParamName = 'OCR'      ; Units    = '-'
    Case (7)
        ParamName = 'POP'      ; Units    = 'F/L^2#'
    Case (8)
        ParamName = 'K0NC'     ; Units    = '-'
    Case (9)
        ParamName = 'STOL'     ; Units    = '-'
    Case Default
        ParamName = '???'      ; Units    = '???'
End Select
Case Default
! model not in DLL
ParamName = ' N/A '          ; Units    = ' N/A '
End Select

Return

```

```
End ! GetParamAndUnit
```

```
Subroutine Get_StateVar_Count( iMod , nVar )
```

```
!  
! Return the number of state variables of the different models  
!
```

```
Select Case (iMod)
```

```
Case (1)  
  nVar = 12
```

```
Case (2)  
  nVar = 12
```

```
Case (3)  
  nVar = 16
```

```
Case (4)  
  nVar = 16
```

```
Case Default  
  nVar = 0
```

```
End Select
```

```
Return
```

```
End
```

```
Subroutine GetStateVarNameAndUnit( iMod , iVar, Name, Unit )
```

```
!  
! Return the name and unit of the different state variables of the different  
models
```

```
!  
Character (Len=255) Name, Unit
```

```
Select Case (iMod)
```

```
Case (1)
```

```
  Select Case (iVar)
```

```
    Case (1)  
      Name = 'RSV'          ; Unit = '-'
```

```
    Case (2)  
      Name = 'Gama'        ; Unit = '-'
```

```
    Case (3)  
      Name = 'R'           ; Unit = '-'
```

```
    Case (4)  
      Name = 'PSIOC'       ; Unit = '-'
```

```
    Case (5)  
      Name = 'PSINC'       ; Unit = '-'
```

```
    Case (6)  
      Name = 'omega'       ; Unit = '-'
```

```
    Case (7)  
      Name = 'p0'          ; Unit = 'F/L^2#'
```

```
    Case (8)  
      Name = 'p0bs'        ; Unit = 'F/L^2#'
```

```
    Case (9)  
      Name = 'Sigma,v,max' ; Unit = 'F/L^2#'
```

```
    Case (10)  
      Name = 'OCR'         ; Unit = '-'
```

```
    Case (11)  
      Name = 'G'           ; Unit = 'F/L^2#'
```

```
    Case (12)  
      Name = 'IMC'         ; Unit = '-'
```

```
    Case Default  
      Name='N/A'           ; Unit = '?'
```

```
  End Select
```

```
Case (2)
```

```
  Select Case (iVar)
```

```

Case (1)
  Name = 'RSV'           ; Unit = '-'
Case (2)
  Name = 'Gama'         ; Unit = '-'
Case (3)
  Name = 'R'            ; Unit = '-'
Case (4)
  Name = 'PSIOC'        ; Unit = '-'
Case (5)
  Name = 'PSINC'        ; Unit = '-'
Case (6)
  Name = 'omega'        ; Unit = '-'
Case (7)
  Name = 'p0'           ; Unit = 'F/L^2#'
Case (8)
  Name = 'p0bs'         ; Unit = 'F/L^2#'
Case (9)
  Name = 'Sigma,v,max'  ; Unit = 'F/L^2#'
Case (10)
  Name = 'OCR'          ; Unit = '-'
Case (11)
  Name = 'G'            ; Unit = 'F/L^2#'
Case (12)
  Name = 'IMC'          ; Unit = '-'
Case Default
  Name='N/A'            ; Unit = '?'
End Select
Case (3)
Select Case (iVar)
  Case (1)
    Name = 'RSV'           ; Unit = '-'
  Case (2)
    Name = 'Gama'         ; Unit = '-'
  Case (3)
    Name = 'R'            ; Unit = '-'
  Case (4)
    Name = 'PSIOC'        ; Unit = '-'
  Case (5)
    Name = 'PSINC'        ; Unit = '-'
  Case (6)
    Name = 'omega'        ; Unit = '-'
  Case (7)
    Name = 'p0'           ; Unit = 'F/L^2#'
  Case (8)
    Name = 'p0bs'         ; Unit = 'F/L^2#'
  Case (9)
    Name = 'Sigma,v,max'  ; Unit = 'F/L^2#'
  Case (10)
    Name = 'OCR'          ; Unit = '-'
  Case (11)
    Name = 'G'            ; Unit = 'F/L^2#'
  Case (12)
    Name = 'IMC'          ; Unit = '-'
  Case (13)
    Name = 'its'          ; Unit = '-'
  Case (14)
    Name = 'iss'          ; Unit = '-'
  Case (15)
    Name = 'subtime'      ; Unit = '-'
  Case (16)
    Name = 'Error'        ; Unit = '-'
Case Default

```

```

        Name='N/A'                ; Unit = '?'
    End Select
Case (4)
Select Case (iVar)
Case (1)
    Name = 'RSV'                ; Unit = '-'
Case (2)
    Name = 'Gama'              ; Unit = '-'
Case (3)
    Name = 'R'                  ; Unit = '-'
Case (4)
    Name = 'PSIOC'              ; Unit = '-'
Case (5)
    Name = 'PSINC'              ; Unit = '-'
Case (6)
    Name = 'omega'              ; Unit = '-'
Case (7)
    Name = 'p0'                  ; Unit = 'F/L^2#'
Case (8)
    Name = 'p0bs'               ; Unit = 'F/L^2#'
Case (9)
    Name = 'Sigma,v,max'        ; Unit = 'F/L^2#'
Case (10)
    Name = 'OCR'                 ; Unit = '-'
Case (11)
    Name = 'G'                   ; Unit = 'F/L^2#'
Case (12)
    Name = 'IMC'                 ; Unit = '-'
Case (13)
    Name = 'its'                 ; Unit = '-'
Case (14)
    Name = 'iss'                 ; Unit = '-'
Case (15)
    Name = 'subtime'            ; Unit = '-'
Case (16)
    Name = 'Error'              ; Unit = '-'
Case Default
    Name='N/A'                  ; Unit = '?'
End Select
Case Default
    Name='N/A'                  ; Unit = '?'
End Select

Return
End

```


OKMSGIVF.for

```
Subroutine OK_MessageBox(t)
  use dfwin
  use dfwinty
  integer (kind=Int_Ptr_Kind()) hWnd

  character*(*) t

  character(len=256) :: msg, title

! Display a messagebox with an OK button
! Note that all strings must be null terminated for C's sake

  msg = Trim(t) // char(0)
  title = 'UDSM' // char(0)

  hWnd = 0

  iret = MessageBox( hWnd,
*                      msg,
*                      title,
*                      MB_OK )

  Return
End
```

MYMODEL1.for

```

Subroutine MyModel1 ( IDTask, iMod, IsUndr,
*           iStep, iTer, iEl, Int,
*           X, Y, Z,
*           Time0, dTime,
*           Props, Sig0, Swp0, StVar0,
*           dEps, D, BulkW,
*           Sig, Swp, StVar, ipl,
*           nStat,
*           NonSym, iStrsDep, iTimeDep, iTang,
*           iAbort )
!
! Purpose: User supplied soil model, basic HASP, implicit gpm with mean effective
stress as governing parameter
! Depending on IDTask, 1 : Initialize state variables
!                   2 : calculate stresses,
!                   3 : calculate material stiffness matrix
!                   4 : return number of state variables
!                   5 : inquire matrix properties
!                   return switch for non-symmetric D-matrix
!                   stress/time dependent matrix
!                   6 : calculate elastic material stiffness matrix
! Arguments:
! I/O  Type
! IDTask I  I   : see above
! iMod   I  I   : model number (1..10)
! IsUndr I  I   : =1 for undrained, 0 otherwise
! iStep  I  I   : Global step number
! iter   I  I   : Global iteration number
! iel    I  I   : Global element number
! Int    I  I   : Global integration point number
! X      I  R   : X-Position of integration point
! Y      I  R   : Y-Position of integration point
! Z      I  R   : Z-Position of integration point
! Time0  I  R   : Time at start of step
! dTime  I  R   : Time increment
! Props  I  R()  : List with model parameters
! Sig0   I  R()  : Stresses at start of step
! Swp0   I  R   : Excess pore pressure start of step
! StVar0 I  R()  : State variable at start of step
! dEps   I  R()  : Strain increment
! D      I/O R(,) : Material stiffness matrix
! BulkW  I/O R   : Bulkmodulus for water (undrained only)
! Sig    0  R()  : Resulting stresses
! Swp    0  R   : Resulting excess pore pressure
! StVar  0  R()  : Resulting values state variables
! ipl    0  I   : Plasticity indicator
! nStat  0  I   : Number of state variables
! NonSym 0  I   : Non-Symmetric D-matrix ?
! iStrsDep 0 I   : =1 for stress dependent D-matrix
! iTimeDep 0 I   : =1 for time dependent D-matrix
! iAbort  0  I   : =1 to force stopping of calculation
!
! Implicit Double Precision (A-H, 0-Z)
!
! Dimension Props(*), Sig0(*), StVar0(*), dEps(*), D(6,6),
*           Sig(*), StVar(*)
!
!--- Local variables (Arrays)
!
```

```

! dStrain(6)      : Array of strain increments (compression is positive) (IDTask=2)
! dStraind(6)    : Array of deviatoric strain increments (compression is positive)
(IDTask=2)
! dStress(6)     : Array of stress increments (compression is positive) (IDTask=2)
! Straint(6)     : Array of strains at start of step (compression is positive)
(IDTask=2)
! Straintt(6)    : Array of strains at the end of step (compression is positive)
(IDTask=2)
! Stressdt(6)    : Array of deviatoric stresses at start of step (compression is
positive) (IDTask=2)
! Stresst(6)     : Array of stresses at start of step (compression is positive)
(IDTask=2)
! Stresstt(6)    : Array of stresses at the end of step (compression is positive)
(IDTask=2)
!
!--- Local variables (Matrices)
!
!
!--- Local variables
!
! Bulk           : Bulk modulus at start of step (IDTask=2, IDTask=4, IDTask=6)
! BulkW          : Bulk modulus of water (IDTask=4, IDTask=6)
! CStol          : Tolerance for assessing whether the critical state is reached
(IDTask=2)
! dEpsV          : Increment of volumetric strain (IDTask=2)
! dEpsq          : Increment of deviatoric strain (IDTask=2)
! dp             : Increment of mean effective stress in zero search (IDTask=2)
! dShearStrain  : Increment of shear strain invariant (IDTask=2)
! dSwp           : Increment of excess pore water pressure (IDTask=2)
! D1,D2 and D3  : Auxilliary variables for calculation od D matrix (IDTask=4,
IDTask=6)
! EpsV           : Volumetric strain (IDTask=3, IDTask=6)
! et             : Void ratio at start of step (IDTask=2, IDTask=4, IDTask=6)
! etai          : Initial stress ratio (IDTask=1)
! etamax        : Max previous stress ratio (IDTask=1)
! etat          : Shear ratio at start of step (IDTask=2)
! ett           : Void ratio at the end of step (IDTask=2)
! e0            : Initial void ratio (IDTask=1, IDTask=2, IDTask=4, IDTask=6)
! fprim         : First derivative of yield function in GPMP algorithm (IDTask=2)
! fsecond       : Second derivative of yield function in GPMP algorithm (IDTask=2)
! ftol          : Tolerance for zero value of yield function (IDTask=2)
! f1,f2,f3,...  : Values of yield function for p1,p2,p3,... (IDTask=2)
! G             : Shear modulus at start of step (IDTask=2, IDTask=4, IDTask=6)
! Gama          : specific volume for CLS at p'=1 kPa (IDTask=1, IDTask=2)
! ilam          : Auxilliary counter in GPMP algorithm (IDTask)=2
! kmax          : Max number of iterations for zero search (IDTask=2)
! k             : counter of number of iterations for zero searh (IDTask=2)
! nStatV        : Number of used state variables
! OCR           : Input value of overconsolidation ratio (IDTask=1)
! omega         : Hardening coefficient (IDTask=1, IDTask=2)
! omegalim     : Limiting value for hardening coefficient in order to avoid
numerical instability (IDTask=2)
! pe           : Elastic trial value of p in GPMP algorithm (IDTask=2)
! pi           : Initial value of mean effective stress
! Pie          : Value of Pi (IDTask=1)
! pmax         : Max previous mean effective stress (IDTask=1)
! POM1, POM2, ...: Auxilliary real variables
! POP          : Input value of preoverburden pressure (IDTask=1)
! pnew         : New guess of p
! pt           : mean effective stress at start of step (IDTask=2, IDTask=4,
IDTask=6)

```

```

! ptt          : mean effective stress at the end of step (IDTask=2)
! PSINC        : State parameter for NC soil (IDTask=1, IDTask=2)
! PSIOC        : State parameter for OC soil ((IDTask=1, IDTask=2)
! p0           : Hardening parameter for yield surface (IDTask=1, IDTask=2)
! p0bs        : Hardening parameter for bounding surface (IDTask=1, IDTask=2)
! p0t         : Hardening parameter for current yield surface at start of step
(IDTask=2)
! p0tt        : Hardening parameter for current yield surface at start of step
(IDTask=2)
! p1,p2,p3,... : guesses of p in GMP algorithm (IDTask=2)
! qi          : Initial value of shear deviator stress (IDTask=1)
! qmax        : Max previous shear deviator stress (IDTask=1)
! qt          : Shear deviator stress at start of step (IDTask=2)
! qtt        : Shear deviator stress at the end of step (IDTask=2)
! R           : Isotropic OCR ratio (IDTask=1, IDTask=2)
! Rmax        : Max value of R in order to avoid numerical instability (IDTask=1,
IDTask=2)
! Rmin        : Min value of R in order to avoid non-possible values (IDTask=1,
IDTask=2)
! Sxmax       : Max previous horizontal stress in x direction (IDTask=1)
! Symax       : Max previous vertical stress (IDTask=1)
! Szmax       : Max previous horizontal stress in y direction (IDTask=1)
! thetat      : Lodes angle at start of step
! xk          : kappa, slope of U/R line (IDTask=1, IDTask=2, IDTask=4, IDTask=6)
! xKNC        : Ratio of horizontal and vertical stresses at NC state (IDTask=1)
! xl          : lambda, slope of ICL (IDTask=1, IDTask=2, IDTask=4, IDTask=6))
! xM          : M, slope of CSL in p-q plane (IDTask=1, IDTask=2)
! xNu         : Poissons ratio (IDTask=1, IDTask=2, IDTask=4, IDTask=6)
!
!--- Auxilliaty Subroutines
!
! CalculateStressInvariants : Calculates p,q,eta and theta
! CalculateStressDeviator  : Calculates array of deviatoric stress
! CalculateVoidRatio       : Calculates current void ratio
! CalculateStrainInvariants : Calculates volumetric strain, deviatoric strain
invariant and shear strain invariant
! CalculateStrainDeviator  : Calculates array of deviatoric strain
! CalculateElasticModuli   : Calculates bulk and shear modulus at start of step
! CalculateOmega           : Calculates hardening coefficient
! CalculateGMPYield        : Calculates value of yield function for input mean
effective stress using GPM integration with mean effective stress as governing
parameter
! CalculateGMPStress       : Calculates stresses at the end of step for input mean
effective stress using GPM integration with mean effective stress as governing
parameter
! MZEROR                   : Fills real array with zeros
! COPYRVEC                  : Copies a Double array R1 with Dimension K to R2
!
Dimension Stresst(6), Straint(6), dStrain(6), Straintt(6),
*          Stressdt(6), dStraind(6), dStress(6), Stresstt(6),
*          Stressdtt(6), dSig(6)
Parameter (ftol=1.D-9, dp=1.D-6, CStol=1.D-6,
*omegalim=15000, Rmax=100.0, Rmin=1.0,
*Pie=3.14159 26535 89793 23846 26433 83279 50288 41971 69399d0)

! Expected contents of Props(1..?)
! 1 : lambda slope of ICL
! 2 : kappa slope of U/R line
! 3 : M slope of CSL in p-q plane
! 4 : nu Poisson ratio
! 5 : evoid0 Initial void ration
! 6 : OCR overconsolidation ratio

```

```

! 7 : POP      preoverburden pressure
! 8 : K0NC     ratio of horizontal and vertical stresses

nStatV = 12
If (IDTask .Eq. 1) Then ! Initialize state variables
  if (StVar0(1).eq.0) then !Reset state variables
    ! Get Props
    xl=Props(1)
    xk=Props(2)
    xM=Props(3)
    xNu=Props(4)
    e0=Props(5)
    OCR=Props(6)
    POP=Props(7)
    xKNC=Props(8)
    ! Calculate Gama from OCR, POP
    Symax=-OCR*Sig0(2)+POP
    Sxmax=xKNC*Symax
    Szmax=xKNC*Symax
    pmax=(Symax+Sxmax+Szmax)/3
    qmax=abs(Symax-Sxmax)
    etamax=qmax/pmax
    pi=-(Sig0(1)+Sig0(2)+Sig0(3))/3.0
    qi=sqrt(3.0*(1.0/6.0*((Sig0(1)-Sig0(2))**2.0+
*(Sig0(2)-Sig0(3))**2.0+(Sig0(3)-Sig0(1))**2.0)+Sig0(4)**2.0+
*Sig0(5)**2.0+Sig0(6)**2.0))
    etai=qi/pi
    POM1=(pmax*(etamax**2.0+xM**2.0)/2.0/xM**2.0)**((xl-xk)/xl)
    POM2=pi**((xk/xl)
    Gama=1.0+e0+xl*log(POM1*POM2)
    PSIOC=1.0+e0+xl*log(pi)-Gama
    PSINC=(xl-xk)*log(2.0*xM**2.0/(xM**2.0+etai**2.0))
    R=exp((PSINC-PSIOC)/(xl-xk))
    If (R.gt.Rmax) then
      Gama=1.0+e0+xl*log(pi)-PSINC+(xl-xk)*log(Rmax)
      R=Rmax
    end if
    if (R.lt.Rmin) then
      Gama=1.0+e0+xl*log(pi)-PSINC+(xl-xk)*log(Rmin)
      R=Rmin
    end if
    omega=(1.0+(PSINC-PSIOC)/PSINC)*R
    p0=qi**2.0/(pi*xM**2.0)+pi
    p0bs=R*p0
    ! Check if value of Poissons ratio is ok
    If (xNu.Gt.0.5d0-1d-6 .Or. xNu.Lt.-1d-6) Then
      Call OK_MessageBox(' Invalid value for "nu" (param 4)')
      iAbort = 1
      Return
    End If
    if (xNu.Lt.0) then
      xNu = 0.0
      Props(4) = xNu
    end if
    if (xNu.Gt.0.495d0) then
      xNu = 0.495
      Props(4) = xNu
    end if
    ! Fill state variable array
    do i=1,nStatV
      StVar0(i)=0.0
    end do

```

```

StVar0(1)=1.0
StVar0(2)=Gama
StVar0(3)=R
StVar0(4)=PSIOC
StVar0(5)=PSINC
StVar0(6)=omega
StVar0(7)=p0
StVar0(8)=p0bs
StVar0(9)=Symax
StVar0(10)=Symax/(-Sig0(2))
StVar0(12)=1.0
end if
If (iel+int.eq.2) Call WriVec(1, 'Props', Props, 8)
End If ! IDTask = 1

If (IDTask .Eq. 2) Then ! Calculate stresses
  If (IsUndr.Eq.1) Then
    dEpsV = dEps(1) + dEps(2) + dEps(3)
    dSwp = BulkW * dEpsV
    Swp = Swp0 + dSwp
  Else
    Swp = Swp0
  End If
  If (iEl+Int+iter.Eq.3 .And. iStep.Lt.10) Then
    Call WriMat( -1, 'D66', D, 6, 6, 6 )
  End If
! Material properties
xl=Props(1)
xk=Props(2)
xM=Props(3)
xNu=Props(4)
e0=Props(5)
! State variables
Gama=StVar0(2)
! Reversing sign for input stresses and strains
do i=1,6
  Stresst(i)=-Sig0(i)
  Straint(i)=-dEps(i+6)
  dStrain(i)=-dEps(i)
  Straintt(i)=Straint(i)+dStrain(i)
end do
!Stress and strain invariants
Call CalculateStressInvariants(Stresst,pt,qt,etat,thetat)
Call CalculateStressDeviator(Stresst,pt,Stressdt)
Call CalculateVoidRatio(Straint,e0,et)
Call CalculateVoidRatio(Straintt,e0,ett)
Call CalculateStrainInvariants(dStrain,dEpsv,dEpsq,dShearStrain)
Call CalculateStrainDeviator(dStrain,dEpsv,dStraind)
!Material modules
Call CalculateElasticModuli(et,pt,xk,xNu,Bulk,G)
!Hardening parameter for curent yield surface
p0t=qt**2.0/pt/xM**2.0+pt
!Calculate OMEGA - hardening coefficient at the beginning of increment
Call CalculateOmega(xl,xk,xM,Gama,pt,qt,et,omegalim,Rmax,Rmin,
*omega,R,PSIOC,PSINC)
! Stress determination GPM P algorithm
!
! Trial step
pe=pt*exp((et-ett)/xk)
ipl=4
k=1
ilam=1

```

```

kmax=25
p1=pe
if (p1.lt.0.001) p1=0.001
100 if (k.gt.kmax) goto 110
Call CalculateGMPYield(xl,xk,xM,et,ett,G,pt,qt,p0t,dEpsq,
*dStraind,Stressdt,CStol,omega,p1,f1)
if (abs(f1).lt.ftol) then
goto 110
else
p2=p1+dp
p3=p1-dp
Call CalculateGMPYield(xl,xk,xM,et,ett,G,pt,qt,p0t,dEpsq,
*dStraind,Stressdt,CStol,omega,p2,f2)
Call CalculateGMPYield(xl,xk,xM,et,ett,G,pt,qt,p0t,dEpsq,
*dStraind,Stressdt,CStol,omega,p3,f3)
if (isnan((f2-f3)/2.0/dp)) then
if (ilam.eq.2) goto 110
ilam=2
pomA=exp((et-ett)*omega/(xl-xk))
p1=pt*(1.0/pomA)**((xk-xl)/xk/omega)
if (p1.lt.0.001) p1=0.001
k=1
goto 100
End If
fprim=(f2-f3)/2.0/dp
fsecond=(f2-2.0*f1+f3)/dp**2.0
if (isnan(p1-2.0*f1*fprim/(2.0*fprim**2.0-f1*fsecond))) then
if (ilam.eq.2) goto 110
ilam=2
pomA=exp((et-ett)*omega/(xl-xk))
p1=pt*(1.0/pomA)**((xk-xl)/xk/omega)
if (p1.lt.0.001) p1=0.001
k=1
goto 100
end if
pnew=p1-2.0*f1*fprim/(2.0*fprim**2.0-f1*fsecond)
k=k+1
p1=pnew
if (p1.lt.0.001) then
p1=0.001
goto 110
end if
goto 100
end if
110 continue
ptt=p1
Call CalculateGMPStress(xl,xk,xM,et,ett,G,pt,qt,p0t,dEpsq,
*dStraind,Stressdt,CStol,omega,ptt,dlam,p0tt,Stresstt)
if (dlam.lt.0.0.and.ilam.eq.1) then
ilam=2
pomA=exp((et-ett)*omega/(xl-xk))
p1=pt*(1.0/pomA)**((xk-xl)/xk/omega)
if (p1.lt.0.001) p1=0.001
k=1
goto 100
end if
if (dlam.lt.0.0.and.ilam.eq.2) then
pomA=exp((et-ett)*omega/(xl-xk))
ptt=pt*(1.0/pomA)**((xk-xl)/xk/omega)
if (ptt.lt.0.001) ptt=0.001
Call CalculateGMPStress(xl,xk,xM,et,ett,G,pt,qt,p0t,dEpsq,
*dStraind,Stressdt,CStol,omega,ptt,dlam,p0tt,Stresstt)

```

```

end if
if (isnan(ptt).or.ptt.gt.huge(1.0d0)) then
    ptt=pt
    Stresstt=Stresst
    p0tt=p0t
end if
!
!Return stresses
do i=1,6
    Sig(i)=-Stresstt(i)
end do
150 continue
!Update state variables
StVar(1)=StVar0(1)
StVar(2)=StVar0(2)
StVar(3)=R
StVar(4)=PSIOC
StVar(5)=PSINC
StVar(6)=omega
StVar(7)=p0t
StVar(8)=p0t*R
If (StVar0(9).lt.Stresstt(2)) then
    StVar(9)=Stresstt(2)
else
    StVar(9)=StVar0(9)
end if
StVar(10)=StVar(9)/Stresstt(2)
StVar(11)=G
StVar(12)=1.0
End If ! IDTask = 2

If ( IDTask .Eq. 3 .Or.
* IDTask .Eq. 6 ) Then ! Calculate D-Matrix
    xk=Props(2)
    xNu=Props(4)
    e0=Props(5)
    pt=-(Sig0(1)+Sig0(2)+Sig0(3))/3.0
    EpsV=-(dEps(7)+dEps(8)+dEps(9))
    et=(1.0+e0)*exp(-EpsV)-1.0
    Call CalculateElasticModuli(et, pt, xk, xNu, Bulk, G)
    D1=Bulk+4.0*G/3.0
    D2=Bulk-2.0*G/3.0
    D3=G
    Call MZeroR(D, 36)
    D(1,1)=D1
    D(2,2)=D1
    D(3,3)=D1
    D(1,2)=D2
    D(1,3)=D2
    D(2,1)=D2
    D(2,3)=D2
    D(3,1)=D2
    D(3,2)=D2
    D(4,4)=D3
    D(5,5)=D3
    D(6,6)=D3
    BulkW = 0
    If (IsUndr.Eq.1) Then
        ! BulkW = ...
        xNu_U = 0.495d0
        Fac=(1+xNu_U)/(1-2*xNu_U) - (1+xNu)/(1-2*xNu)
        Fac=2D0*G/3D0 * Fac
    
```



```
        BulkW = Fac
    End If
End If ! IDTask = 3, 6

If (IDTask .Eq. 4) Then ! Number of state parameters
    nStat = nStatV
End If ! IDTask = 4

If (IDTask .Eq. 5) Then ! matrix type
    NonSym = 0 ! 1 for non-symmetric D-matrix
    iStrsDep = 1 ! 1 for stress dependent D-matrix
    iTang = 0 ! 1 for tangent D-matrix
    iTimeDep = 0 ! 1 for time dependent D-matrix
End If ! IDTask = 5

Return
End ! MyModel1
```

MYMODEL2.for

```
Subroutine MyModel2 ( IDTask, iMod, IsUndr,  
*               iStep, iTer, iEl, Int,  
*               X, Y, Z,  
*               Time0, dTime,  
*               Props, Sig0, Swp0, StVar0,  
*               dEps, D, BulkW,  
*               Sig, Swp, StVar, ipl,  
*               nStat,  
*               NonSym, iStrsDep, iTimeDep, iTang,  
*               iAbort )  
!  
! Purpose: User supplied soil model, basic HASP, implicit Borja and Lee algorithm  
! Depending on IDTask, 1 : Initialize state variables  
!                   2 : calculate stresses,  
!                   3 : calculate material stiffness matrix  
!                   4 : return number of state variables  
!                   5 : inquire matrix properties  
!                   return switch for non-symmetric D-matrix  
!                   stress/time dependent matrix  
!                   6 : calculate elastic material stiffness matrix  
!  
! Arguments:  
!       I/O  Type  
! IDTask    I   I   : see above  
! iMod      I   I   : model number (1..10)  
! IsUndr    I   I   : =1 for undrained, 0 otherwise  
! iStep     I   I   : Global step number  
! iter      I   I   : Global iteration number  
! iel       I   I   : Global element number  
! Int       I   I   : Global integration point number  
! X         I   R   : X-Position of integration point  
! Y         I   R   : Y-Position of integration point  
! Z         I   R   : Z-Position of integration point  
! Time0     I   R   : Time at start of step  
! dTime     I   R   : Time increment  
! Props     I   R( ) : List with model parameters  
! Sig0      I   R( ) : Stresses at start of step  
! Swp0      I   R   : Excess pore pressure start of step  
! StVar0    I   R( ) : State variable at start of step  
! dEps      I   R( ) : Strain increment  
! D         I/O  R(, ) : Material stiffness matrix  
! BulkW     I/O  R   : Bulkmodulus for water (undrained only)  
! Sig       0   R( ) : Resulting stresses  
! Swp       0   R   : Resulting excess pore pressure  
! StVar     0   R( ) : Resulting values state variables  
! ipl       0   I   : Plasticity indicator  
! nStat     0   I   : Number of state variables  
! NonSym    0   I   : Non-Symmetric D-matrix ?  
! iStrsDep  0   I   : =1 for stress dependent D-matrix  
! iTimeDep  0   I   : =1 for time dependent D-matrix  
! iAbort    0   I   : =1 to force stopping of calculation  
!  
!       Implicit Double Precision (A-H, O-Z)  
!  
!       Dimension Props(*), Sig0(*), StVar0(*), dEps(*), D(6,6),  
*           Sig(*), StVar(*)  
!  
!---- Local variables (Arrays)  
!  
! dSig(6)      : Array of stress increment (IDTask=2)
```

```

! dStrain(6)      : Array of strain increments (compression is positive) (IDTask=2)
! dStraind(6)    : Array of deviatoric strain increments (compression is positive)
(IDTask=2)
! dStress(6)     : Array of stress increments (compression is positive) (IDTask=2)
! Straint(6)     : Array of strains at start of step (compression is positive)
(IDTask=2)
! Straintt(6)    : Array of strains at the end of step (compression is positive)
(IDTask=2)
! Stressdt(6)    : Array of deviatoric stresses at start of step (compression is
positive) (IDTask=2)
! Stressdtt(6)   : Array of deviatoric stresses at the end of step (compression is
positive) (IDTask=2)
! Stresst(6)     : Array of stresses at start of step (compression is positive)
(IDTask=2)
! Stresstr(6)    : Array of stresses for elastic prediction (IDTask=2)
! Stresstt(6)    : Array of stresses at the end of step (compression is positive)
(IDTask=2)
!
!--- Local variables (Matrices)
!
!
!--- Local variables
!
! AUX            : Auxilliary real variable (IDTask=2)
! Bulk           : Bulk modulus at start of step (IDTask=2, IDTask=4, IDTask=6)
! BulkW          : Bulk modulus of water (IDTask=4, IDTask=6)
! dEpsV          : Increment of volumetric strain (IDTask=2)
! dEpsq          : Increment of deviatoric strain (IDTask=2)
! dEpsqp         : Increment of plastic deviatoric strain (IDTask=2)
! df             : First derivative of yield function (IDTask=2)
! dH             : First derivative of function H (IDTask=2)
! dlam           : Plastic multiplier (IDTask=2)
! dq             : Increment of shear stress invariant (IDTask=2)
! dShearStrain  : Increment of shear strain invariant (IDTask=2)
! dSwp           : Increment of excess pore water pressure (IDTask=2)
! D1,D2 and D3  : Auxilliary variables for calculation od D matrix (IDTask=4,
IDTask=6)
! EpsV          : Volumetric strain (IDTask=3, IDTask=6)
! et            : Void ratio at start of step (IDTask=2, IDTask=4, IDTask=6)
! etai          : Initial stress ratio (IDTask=1)
! etamax        : Max previous stress ratio (IDTask=1)
! etat          : Shear ratio at start of step (IDTask=2)
! e0            : Initial void ratio (IDTask=1, IDTask=2, IDTask=4, IDTask=6)
! ftol          : Tolerance for zero value of yield function (IDTask=2)
! G             : Shear modulus at start of step (IDTask=2, IDTask=4, IDTask=6)
! Gama          : Specific volume for CLS at p'=1 kPa (IDTask=1, IDTask=2)
! H             : Value of function H for finding p0tt value (IDTask=2)
! iH            : Counter, number of iterations in Borja and Lee algorithm for
finding p0tt (IDTask)=2
! il            : Counter, number of iterations in Borja and Lee algorithm for
finding plastic multiplier (IDTask)=2
! nStatV        : Number of used state variables
! OCR           : Input value of overconsolidation ratio (IDTask=1)
! omega         : Hardening coefficient (IDTask=1, IDTask=2)
! omegalim      : Limiting value for hardening coefficient in order to avoid
numerical instability (IDTask=2)
! pi            : Initial value of mean effective stress
! Pie           : Value of Pi (IDTask=1)
! pmax          : Max previous mean effective stress (IDTask=1)
! POM1, POM2, ... : Auxilliary real variables
! POP           : Input value of preoverburden pressure (IDTask=1)

```

```

! pt          : Mean effective stress at start of step (IDTask=2, IDTask=4,
IDTask=6)
! ptr          : Elastic trial value of p (IDTask=2)
! ptt          : mean effective stress at the end of step (IDTask=2)
! PSINC        : State parameter for NC soil (IDTask=1, IDTask=2)
! PSIOC        : State parameter for OC soil ((IDTask=1, IDTask=2)
! p0           : Hardening parameter for yield surface (IDTask=1, IDTask=2)
! p0bs         : Hardening parameter for bounding surface (IDTask=1, IDTask=2)
! p0t          : Hardening parameter for current yield surface at start of step
(IDTask=2)
! p0tt         : Hardening parameter for current yield surface at start of step
(IDTask=2)
! p1,p2,p3,... : guesses of p in GPMP algorithm (IDTask=2)
! qi           : Initial value of shear deviator stress (IDTask=1)
! qmax         : Max previous shear deviator stress (IDTask=1)
! qt           : Shear deviator stress at start of step (IDTask=2)
! qtr          : Elastic trial value of q (IDTask=2)
! qtt          : Shear deviator stress at the end of step (IDTask=2)
! R            : Isotropic OCR ratio (IDTask=1, IDTask=2)
! Rmax         : Max value of R in order to avoid numerical instability (IDTask=1,
IDTask=2)
! Rmin         : Min value of R in order to avoid non-possible values (IDTask=1,
IDTask=2)
! Sxmax        : Max previous horizontal stress in x direction (IDTask=1)
! Symax        : Max previous vertical stress (IDTask=1)
! Szmax        : Max previous horizontal stress in y direction (IDTask=1)
! thetat       : Lodes angle at start of step
! xk           : kappa, slope of U/R line (IDTask=1, IDTask=2, IDTask=4, IDTask=6)
! xKNC         : Ratio of horizontal and vertical stresses at NC state (IDTask=1)
! xL           : lambda, slope of ICL (IDTask=1, IDTask=2, IDTask=4, IDTask=6))
! xM           : M, slope of CSL in p-q plane (IDTask=1, IDTask=2)
! xNu          : Poissons ratio (IDTask=1, IDTask=2, IDTask=4, IDTask=6)
!
!--- Auxilliaty Subroutines
!
! CalculateStressInvariants : Calculates p,q,eta and theta
! CalculateStressDeviator  : Calculates array of deviatoric stress
! CalculateVoidRatio       : Calculates current void ratio
! CalculateStrainInvariants : Calculates volumetric strain, deviatoric strain
invariant and shear strain invariant
! CalculateStrainDeviator  : Calculates array of deviatoric strain
! CalculateElasticModuli   : Calculates bulk and shear modulus at start of step
! CalculateOmega           : Calculates hardening coefficient
! MZEROR                   : Fills real array with zeros
! COPYRVEC                 : Copies a Double array R1 with Dimension K to R2
!
Dimension Stresst(6), Straint(6), dStrain(6), Straintt(6),
*          Stressdt(6), dStraind(6), dStress(6), Stresstt(6),
*          Stressdtt(6), dSig(6), Stresstr(6)
Parameter (ftol=1.D-9, Htol=1.D-6,
*omegalim=15000, Rmax=100.0, Rmin=1.0,
*Pie=3.14159 26535 89793 23846 26433 83279 50288 41971 69399d0)

! Expected contents of Props(1..?)
! 1 : lambda   slope of ICL
! 2 : kappa    slope of U/R line
! 3 : M        slope of CSL in p-q plane
! 4 : nu       Poisson ratio
! 5 : evoid0   Initial void ration
! 6 : OCR      overconsolidation ratio
! 7 : POP      preoverburden pressure
! 8 : K0NC     ratio of horizontal and vertical stresses

```

```

nStatV = 12
If (IDTask .Eq. 1) Then ! Initialize state variables
  if (StVar0(1).eq.0) then !Reset state variables
    ! Get Props
    xl=Props(1)
    xk=Props(2)
    xM=Props(3)
    xNu=Props(4)
    e0=Props(5)
    OCR=Props(6)
    POP=Props(7)
    xKNC=Props(8)
    ! Calculate Gama from OCR, POP
    Symax=-OCR*Sig0(2)+POP
    Sxmax=xKNC*Symax
    Szmax=xKNC*Symax
    pmax=(Symax+Sxmax+Szmax)/3
    qmax=abs(Symax-Sxmax)
    etamax=qmax/pmax
    pi=-(Sig0(1)+Sig0(2)+Sig0(3))/3.0
    qi=sqrt(3.0*(1.0/6.0*((Sig0(1)-Sig0(2))**2.0+
* (Sig0(2)-Sig0(3))**2.0+(Sig0(3)-Sig0(1))**2.0)+Sig0(4)**2.0+
* Sig0(5)**2.0+Sig0(6)**2.0))
    etai=qi/pi
    POM1=(pmax*(etamax**2.0+xM**2.0)/2.0/xM**2.0)**((xl-xk)/xl)
    POM2=pi**((xk/xl)
    Gama=1.0+e0+xl*log(POM1*POM2)
    PSIOC=1.0+e0+xl*log(pi)-Gama
    PSINC=(xl-xk)*log(2.0*xM**2.0/(xM**2.0+etai**2.0))
    R=exp((PSINC-PSIOC)/(xl-xk))
    If (R.gt.Rmax) then
      Gama=1.0+e0+xl*log(pi)-PSINC+(xl-xk)*log(Rmax)
      R=Rmax
    end if
    if (R.lt.Rmin) then
      Gama=1.0+e0+xl*log(pi)-PSINC+(xl-xk)*log(Rmin)
      R=Rmin
    end if
    omega=(1.0+(PSINC-PSIOC)/PSINC)*R
    p0=qi**2.0/(pi*xM**2.0)+pi
    p0bs=R*p0
    ! Check if value of Poissons ratio is ok
    If (xNu.Gt.0.5d0-1d-6 .Or. xNu.Lt.-1d-6) Then
      Call OK_MessageBox(' Invalid value for "nu" (param 4)')
      iAbort = 1
      Return
    End If
    if (xNu.Lt.0) then
      xNu = 0.0
      Props(4) = xNu
    end if
    if (xNu.Gt.0.495d0) then
      xNu = 0.495
      Props(4) = xNu
    end if
    ! Fill state variable array
    do i=1,nStatV
      StVar0(i)=0.0
    end do
    StVar0(1)=1.0
    StVar0(2)=Gama

```

```

StVar0(3)=R
StVar0(4)=PSIOC
StVar0(5)=PSINC
StVar0(6)=omega
StVar0(7)=p0
StVar0(8)=p0bs
StVar0(9)=Symax
StVar0(10)=Symax/(-Sig0(2))
StVar0(12)=2.0
end if
If (iel+int.eq.2) Call WriVec(1, 'Props', Props, 8)
End If ! IDTask = 1

If (IDTask .Eq. 2) Then ! Calculate stresses
If (IsUndr.Eq.1) Then
dEpsV = dEps(1) + dEps(2) + dEps(3)
dSwp = BulkW * dEpsV
Swp = Swp0 + dSwp
Else
Swp = Swp0
End If
If (iEl+Int+iter.Eq.3 .And. iStep.Lt.10) Then
Call WriMat( -1, 'D66', D, 6, 6, 6 )
End If
! Material properties
xl=Props(1)
xk=Props(2)
xM=Props(3)
xNu=Props(4)
e0=Props(5)
! State variables
Gama=StVar0(2)
! Reversing sign for input stresses and strains
do i=1,6
Stresst(i)=-Sig0(i)
Straint(i)=-dEps(i+6)
dStrain(i)=-dEps(i)
Straintt(i)=Straint(i)+dStrain(i)
end do
!Stress and strain invariants
Call CalculateStressInvariants(Stresst,pt,qt,etat,thetat)
Call CalculateStressDeviator(Stresst,pt,Stressdt)
Call CalculateVoidRatio(Straint,e0,et)
Call CalculateStrainInvariants(dStrain,dEpsv,dEpsq,dShearStrain)
Call CalculateStrainDeviator(dStrain,dEpsv,dStraind)
!Material modules
Call CalculateElasticModuli(et,pt,xk,xNu,Bulk,G)
!Hardening parameter for curent yield surface
p0t=qt**2.0/pt/xM**2.0+pt
!Calculate OMEGA - hardening coefficient at the beginning of increment
Call CalculateOmega(xl,xk,xM,Gama,pt,qt,et,omegalim,Rmax,Rmin,
*omega,R,PSIOC,PSINC)
! Stress determination Borja and Lee algorithm
!
!Trial stresses
ptr=pt+Bulk*dEpsv
qtr=qt+3.0*G*dEpsq
AUX=(1.0+et)*omega/(xl-xk)
ipl=4
if (omega.eq.omegalim.or.omega.eq.-omegalim) goto 150
!Find plastic multiplier, dlam
dlam=0.0

```

```

    il=1
    iH=1
140  p0tt=p0t
130  H=p0t*exp(AUX*dlam*(2.0*ptr-p0tt)/(1.0+2.0*Bulk*dlam))-p0tt
    if (abs(H).lt.Htol.or.iH.gt.25) goto 110
    dH=-AUX*dlam/(1.0+2.0*Bulk*dlam)*p0t*exp(AUX*dlam*
* (2*ptr-p0tt)/(1.0+2.0*Bulk*dlam))-1.0
    p0tt=p0tt-H/dH
    iH=iH+1
110  goto 130
    iH=1
    ptt=(ptr+Bulk*dlam*p0tt)/(1.0+2.0*Bulk*dlam)
    qtt=qtr/(1.0+6.0*G*dlam/xM**2.0)
    f=qtt**2.0/xM**2.0+ptt*(ptt-p0tt)
    if (abs(f).lt.ftol.or.il.gt.25) goto 120
    df=-(2.0*ptt-p0tt)*(Bulk*(2.0*ptt-p0tt)+AUX*ptt*p0tt)/
* (1.0+dlam*(2.0*Bulk+AUX*p0tt))-2.0*qtt**2.0/(xM**2.0*
* (xM**2.0/6.0/G+dlam))
    dlam=dlam-f/df
    il=il+1
    goto 140
150  p0tt=2.0*ptr
    ptt=ptr
    qtt=xM*ptt
    dq=qtt-qt
    dEpsqp=(dEpsq-dq/(3.0*G))/omega
    dlam=xM*dEpsqp/(2.0*ptt)
120  if (dlam.lt.0.0) dlam=0.0
    if (isnan(dlam)) dlam=0.0
    if (dlam.eq.0.0) then
        ptt=ptr
        p0tt=p0t
    end if
    do i=1,6
        Stressdtt(i)=(dStraind(i)+Stressdt(i)/2.0/G)/(3.0*dlam/xM**
* 2.0+1.0/2.0/G)
    end do
    do i=1,3
        Stresstt(i)=Stressdtt(i)+ptt
        Stresstt(i+3)=Stressdtt(i+3)
    end do
!
!Return stresses
do i=1,6
    Sig(i)=-Stresstt(i)
end do
!Update state variables
StVar(1)=StVar0(1)
StVar(2)=StVar0(2)
StVar(3)=R
StVar(4)=PSIOC
StVar(5)=PSINC
StVar(6)=omega
StVar(7)=p0t
StVar(8)=p0t*R
If (StVar0(9).lt.Stresstt(2)) then
    StVar(9)=Stresstt(2)
else
    StVar(9)=StVar0(9)
end if
StVar(10)=StVar(9)/Stresstt(2)
StVar(11)=G

```

```

StVar(12)=2.0
End If ! IDTask = 2

If ( IDTask .Eq. 3 .Or.
*   IDTask .Eq. 6 ) Then ! Calculate D-Matrix
  xk=Props(2)
  xNu=Props(4)
  e0=Props(5)
  pt=-(Sig0(1)+Sig0(2)+Sig0(3))/3.0
  EpsV=-(dEps(7)+dEps(8)+dEps(9))
  et=(1.0+e0)*exp(-EpsV)-1.0
  Call CalculateElasticModuli(et,pt,xk,xNu,Bulk,G)
  D1=Bulk+4.0*G/3.0
  D2=Bulk-2.0*G/3.0
  D3=G
  Call MZeroR(D,36)
  D(1,1)=D1
  D(2,2)=D1
  D(3,3)=D1
  D(1,2)=D2
  D(1,3)=D2
  D(2,1)=D2
  D(2,3)=D2
  D(3,1)=D2
  D(3,2)=D2
  D(4,4)=D3
  D(5,5)=D3
  D(6,6)=D3
  BulkW = 0
  If (IsUndr.Eq.1) Then
    ! BulkW = ...
    xNu_U = 0.495d0
    Fac=(1+xNu_U)/(1-2*xNu_U) - (1+xNu)/(1-2*xNu)
    Fac=2D0*G/3D0 * Fac
    BulkW = Fac
  End If
End If ! IDTask = 3, 6

If (IDTask .Eq. 4) Then ! Number of state parameters
  nStat = nStatV
End If ! IDTask = 4

If (IDTask .Eq. 5) Then ! matrix type
  NonSym = 0 ! 1 for non-symmetric D-matrix
  iStrsDep = 1 ! 1 for stress dependent D-matrix
  iTang = 0 ! 1 for tangent D-matrix
  iTimeDep = 0 ! 1 for time dependent D-matrix
End If ! IDTask = 5

Return
End ! MyModel2

```


MYMODEL3.for

```
Subroutine MyModel3 ( IDTask, iMod, IsUndr,
**                               iStep, iTer, iEl, Int,
**                               X, Y, Z,
**                               Time0, dTime,
**                               Props, Sig0, Swp0, StVar0,
**                               dEps, D, BulkW,
**                               Sig, Swp, StVar, ipl,
**                               nStat,
**                               NonSym, iStrsDep, iTimeDep, iTang,
**                               iAbort )
!
! Purpose: User supplied soil model, basic HASP, explicit modified Euler with
automatic subincrementation and error control
! Depending on IDTask, 1 : Initialize state variables
!                               2 : calculate stresses,
!                               3 : calculate material stiffness matrix
!                               4 : return number of state variables
!                               5 : inquire matrix properties
!                               return switch for non-symmetric D-matrix
!                               stress/time dependent matrix
!                               6 : calculate elastic material stiffness matrix
!
! Arguments:
!           I/O  Type
! IDTask     I   I   : see above
! iMod       I   I   : model number (1..10)
! IsUndr     I   I   : =1 for undrained, 0 otherwise
! iStep      I   I   : Global step number
! iter       I   I   : Global iteration number
! iel        I   I   : Global element number
! Int        I   I   : Global integration point number
! X          I   R   : X-Position of integration point
! Y          I   R   : Y-Position of integration point
! Z          I   R   : Z-Position of integration point
! Time0     I   R   : Time at start of step
! dTime     I   R   : Time increment
! Props     I   R( ) : List with model parameters
! Sig0      I   R( ) : Stresses at start of step
! Swp0      I   R   : Excess pore pressure start of step
! StVar0    I   R( ) : State variable at start of step
! dEps      I   R( ) : Strain increment
! D         I/O  R(, ) : Material stiffness matrix
! BulkW     I/O  R   : Bulkmodulus for water (undrained only)
! Sig       0   R( ) : Resulting stresses
! Swp       0   R   : Resulting excess pore pressure
! StVar     0   R( ) : Resulting values state variables
! ipl       0   I   : Plasticity indicator
! nStat     0   I   : Number of state variables
! NonSym    0   I   : Non-Symmetric D-matrix ?
! iStrsDep  0   I   : =1 for stress dependent D-matrix
! iTimeDep  0   I   : =1 for time dependent D-matrix
! iAbort    0   I   : =1 to force stopping of calculation
!
!       Implicit Double Precision (A-H, 0-Z)
!
!       Dimension Props(*), Sig0(*), StVar0(*), dEps(*), D(6,6),
**           Sig(*), StVar(*)
!
!---- Local variables (Arrays)
!
```

```

! AUXV11(6)      : Array which contains the product of Va and De (transpose(Va) * De)
(modEuler) (IDTask=2)
! AUXV12(6)      : Array which contains the product of Va and De (transpose(Va) * De)
(modEuler) (IDTask=2)
! AUXV21(6)      : Array which contains the product of De and Va (De * Va) (modEuler)
(IDTask=2)
! AUXV22(6)      : Array which contains the product of De and Va (De * Va) (modEuler)
(IDTask=2)
! AUXV3(6)       : Array which contains the product (transpose(Va0) * De) (IDTask=2)
! AUXV4(6)       : Array which contains the product (De * Va0) (IDTask=2)
! dSig(6)        : Array of stress increment (IDTask=2)
! dStresse1(6)   : Array of elastic stress increments (modEuler) (IDTask=2)
! dStresse2(6)   : Array of elastic stress increments (modEuler) (IDTask=2)
! dStrain(6)     : Array of strain increments (compression is positive) (IDTask=2)
! dStress1(6)    : Array of stress increments (compression is positive) (modEuler)
(IDTask=2)
! dStress2(6)    : Array of stress increments (compression is positive) (modEuler)
(IDTask=2)
! dTstrain      : Array of strain increments in one subincrement (IDTask=2)
! Straint(6)    : Array of strains at start of step (compression is positive)
(IDTask=2)
! Straintt(6)   : Array of strains at the end of step (compression is positive)
(IDTask=2)
! Stresst(6)    : Array of stresses at start of step (compression is positive)
(IDTask=2)
! Stresstt(6)   : Array of stresses at the end of step (compression is positive)
(IDTask=2)
! Stress0(6)    : Array of uncorrected stresses - drift correction (IDTask=2)
! Tstrain(6)    : Array of strains at start of subincrement (IDTask=2)
! Tstress(6)    : Array of stresses at start of subincrements (IDTask=2)
! Tstress1(6)   : Array of stresses at start of subincrements (modEuler) (IDTask=2)
! TTstrain(6)   : Array of strains at the end of subincrement (IDTask=2)
! TTstress(6)   : Array of stresset at the end of subincrement (IDTask=2)
! Va1(6)        : Array of yield function derivatives (modEuler) (IDTask=2)
! Va2(6)        : Array of yield function derivatives (modEuler) (IDTask=2)
! Va0(6)        : Array of yield function derivatives for uncorrected stresses -
drift correction (IDTask=2)

```

```

!
!
!--- Local variables (Matrices)
!
! De0(6,6)       : Elastic De matrix in drift correction algorithm (IDTask=2)
! De1(6,6)       : Elastic De matrix (modEuler) (IDTask=2)
! De2(6,6)       : Elastic De matrix (modEuler) (IDTask=2)
!
!--- Local variables
!
! AUX11          : Auxilliary real variable, transpose(Va) * dStresse (modEuler)
(IDTask=2)
! AUX12          : Auxilliary real variable, transpose(Va) * dStresse (modEuler)
(IDTask=2)
! AUX21          : Auxilliary real variable, transpose(Va) * De * Va (modEuler)
(IDTask=2)
! AUX22          : Auxilliary real variable, transpose(Va) * De * Va (modEuler)
(IDTask=2)
! AUX3           : Auxilliary real variable, transpose(Va0) * De * Va0 (IDTask=2)
! AUX4           : Auxilliary real variable, transpose(Va0) * Va0
! Bulk           : Bulk modulus at start of step (IDTask=2, IDTask=4, IDTask=6)
! BulkW          : Bulk modulus of water (IDTask=4, IDTask=6)
! Bulk0          : Bulk modulus in drift correction algorithm (IDTask=2)

```

```

! dEpsV      : Increment of volumetric strain (IDTask=2)
! dlam       : Plastic multiplier (IDTask=2)
! dlam1      : Plastic multiplier (modEuler) (IDTask=2)
! dlam2      : Plastic multiplier (modEuler) (IDTask=2)
! dp01       : Increment of hardening parameter (modEuler) (IDTask=2)
! dp02       : Increment of hardening parameter (modEuler) (IDTask=2)
! dShearStrain : Increment of shear strain invariant (IDTask=2)
! dsubtime   : size of subincrement (IDTask=2)
! dSwp       : Increment of excess pore water pressure (IDTask=2)
! D1,D2 and D3 : Auxilliary variables for calculation od D matrix (IDTask=4,
IDTask=6)
! EM         : Machine Error tolerance (IDTask=2)
! EpsV       : Volumetric strain (IDTask=3, IDTask=6)
! Error      : Error estimation in automatic subincrementation (IDTask=2)
! Error1     : Error estimation in stresses determination in automatic
subincrementation (IDTask=2)
! Error2     : Error estimation in hardening parameter determination in automatic
subincrementation (IDTask=2)
! et        : Void ratio at start of step (IDTask=2, IDTask=4, IDTask=6)
! eta       : Stress ratio (IDTask=2)
! etai      : Initial stress ratio (IDTask=1)
! etamax    : Max previous stress ratio (IDTask=1)
! etat      : Shear ratio at start of step (IDTask=2)
! e0        : Initial void ratio (IDTask=1, IDTask=2, IDTask=4, IDTask=6)
! ftol      : Tolerance for zero value of yield function (IDTask=2)
! f0        : Value of yield function that must be corrected - drift correction
(IDTask=2)
! G         : Shear modulus at start of step (IDTask=2, IDTask=4, IDTask=6)
! G0        : Shear modulus in drift correction algorithm (IDTask=2)
! Gama      : Specific volume for CLS at p'=1 kPa (IDTask=1, IDTask=2)
! iDrift    : Counter in drift correction loop (IDTask=2)
! isuc      : =0 if previous subincrement is successful, =1 if not (IDTask=2)
! iss       : Counts the number of successful subincrements (IDTask=2)
! its       : Counts the number of total subincrements (IDTask=2)
! Nsubs     : Number of subincrements (IDTask=2)
! nStatV    : Number of used state variables
! OCR       : Input value of overconsolidation ratio (IDTask=1)
! omega     : Hardening coefficient (IDTask=1, IDTask=2)
! omegalim  : Limiting value for hardening coefficient in order to avoid
numerical instability (IDTask=2)
! omega0    : Hardening coefficient in drift correction algorithm (IDTask=2)
! p         : mean effective stress (IDTask=2)
! pi        : Initial value of mean effective stress
! Pie       : Value of Pi (IDTask=1)
! pmax      : Max previous mean effective stress (IDTask=1)
! POM1, POM2, ... : Auxilliary real variables
! POP       : Input value of preoverburden pressure (IDTask=1)
! pt        : Mean effective stress at start of step (IDTask=2, IDTask=4,
IDTask=6)
! ptt      : mean effective stress at the end of step (IDTask=2)
! PSINC     : State parameter for NC soil (IDTask=1, IDTask=2)
! PSINC0    : State parameter for NC soil in drift correction algorithm
(IDTask=2)
! PSIOC     : State parameter for OC soil ((IDTask=1, IDTask=2)
! PSIOC0    : State parameter for OC soil in drift correction algorithm
(IDTask=2)
! p0        : Hardening parameter for yield surface (IDTask=1, IDTask=2)
! p0bs     : Hardening parameter for bounding surface (IDTask=1, IDTask=2)
! p0t      : Hardening parameter for current yield surface at start of step
(IDTask=2)
! p0tt     : Hardening parameter for current yield surface at start of step
(IDTask=2)

```

```

! q          : Shear deviator stress (IDTask=2)
! qi         : Initial value of shear deviator stress (IDTask=1)
! qmax      : Max previous shear deviator stress (IDTask=1)
! qt        : Shear deviator stress at start of step (IDTask=2)
! qtt       : Shear deviator stress at the end of step (IDTask=2)
! R         : Isotropic OCR ratio (IDTask=1, IDTask=2)
! Rmax      : Max value of R in order to avoid numerical instability (IDTask=1,
IDTask=2)
! Rmin      : Min value of R in order to avoid non-possible values (IDTask=1,
IDTask=2)
! R0        : Isotropic OCR ratio in drift correction algorithm (IDTask=2)
! stepsize  : Relative size of next subincrement size to the previous
subincrement size (IDTask=2)
! STOL      : Error tolerance for evaluation of subincrement size (IDTask=2)
! subtime1  : Time at the beginning of subincrement
! subtime2  : Time at the end of subincrement
! Sxmax    : Max previous horizontal stress in x direction (IDTask=1)
! Symax    : Max previous vertical stress (IDTask=1)
! Szmax    : Max previous horizontal stress in y direction (IDTask=1)
! TBulk1   : Bulk modulus at the start of subincrement (modEuler) (IDTask=2)
! TBulk2   : Bulk modulus at the start of subincrement (modEuler) (IDTask=2)
! Te       : Void ratio at start of subincrement
! Teta     : Stress ratio at start of increment (IDTask=2)
! theta    : Lodes angle (IDTask=2)
! thetat   : Lodes angle at start of step (IDTask=2)
! TG1      : Shear modulus at start of subincrement (modEuler) (IDTask=2)
! TG2      : Shear modulus at start of subincrement (modEuler) (IDTask=2)
! Tomega1  : hardening coefficient at start of step (modEuler) (IDTask=2)
! Tomega2  : hardening coefficient at start of step (modEuler) (IDTask=2)
! Tp       : Mean effective stress at start of subincrement (IDTask=2)
! TPSINC1  : State parameter for NC soil at start of subincrement (modEuler)
(IDTask=2)
! TPSINC2  : State parameter for NC soil at start of subincrement (modEuler)
(IDTask=2)
! TPSIOC1  : State parameter for OC soil at start of subincrement (modEuler)
(IDTask=2)
! TPSIOC2  : State parameter for OC soil at start of subincrement (modEuler)
(IDTask=2)
! Tp0      : Hardening parameter at start of subincrement (IDTask=2)
! Tp01     : Hardening parameter at start of subincrement (modEuler) (IDTask=2)
! Tp1      : Mean effective stress (modEuler) (IDTask=2)
! Tq       : Deviator stress at start of subincrement (IDTask=2)
! Tq1      : Deviator stress at start of subincrement (modEuler) (IDTask=2)
! TR1      : Isotropic OCR ratio at start of subincrement (modEuler) (IDTask=2)
! TR2      : Isotropic OCR ratio at start of subincrement (modEuler) (IDTask=2)
! TTe      : Void ratio at the end of subincrement (IDTask=2)
! Teta     : Stress ratio at the end of subincrement (IDTask=2)
! Teta1    : Stress ratio at the end of subincrement (modEuler) (IDTask=2)
! TTeta    : Stress ratio at the end of subincrement (IDTask=2)
! TTf      : Yield function at the end of subincrement (IDTask=2)
! Ttheta   : Lodes angle at start of subincrement (IDTask=2)
! Ttheta1  : Lodes angle at start of subincrement (modEuler) (IDTask=2)
! TTp      : Mean effective stress at start of subincrement (IDTask=2)
! TTp0     : Hardening parameter at the end of subincrement (IDTask=2)
! TTq      : Deviator stress at the end of subincrement (IDTask=2)
! TTtheta  : Lodes angle at the end of subincrement (IDTask=2)
! VarA0    : Variable A in stress integration procedure (drift correction)
(IDTask=2)
! VarA1    : Variable A in stress integration procedure (modEuler) (IDTask=2)
! VarA2    : Variable A in stress integration procedure (modEuler) (IDTask=2)
! VarB0    : Variable B in stress integration procedure (drift correction)
(IDTask=2)

```

```

! VarB1      : Variable B in stress integration procedure (modEuler) (IDTask=2)
! VarB2      : Variable B in stress integration procedure (modEuler) (IDTask=2)
! xk         : kappa, slope of U/R line (IDTask=1, IDTask=2, IDTask=4, IDTask=6)
! xKNC       : Ratio of horizontal and vertical stresses at NC state (IDTask=1)
! xL         : Lambda, slope of ICL (IDTask=1, IDTask=2, IDTask=4, IDTask=6))
! xM         : M, slope of CSL in p-q plane (IDTask=1, IDTask=2)
! xNu        : Poissons ratio (IDTask=1, IDTask=2, IDTask=4, IDTask=6)
!

```

```
!--- Auxilliary Subroutines
```

```

! CalculateStressInvariants : Calculates p,q,eta and theta
! CalculateVoidRatio       : Calculates current void ratio
! CalculateElasticModuli   : Calculates bulk and shear modulus at start of step
! CalculateOmega           : Calculates hardening coefficient
! CalculateVectorA         : Calculates vector A which contains derivatives of
yield function
! CalculateDeMatric        : Calculates elastic D matrix
! MZEROR                   : Fills real array with zeros
! MatVec                   : Multiplies matrix with a vector. Result is vector
! DInProd                  : Inner product of two vectors. Result is scalar
! AddVec                   : Calculates sum of two vectors. result is vector
! COPYRVEC                 : Copies a Double array R1 with Dimension K to R2
!

```

```

Dimension Stresst(6), Straint(6), dStrain(6), Straintt(6),
*          dStress1(6), Stresstt(6), dSig(6), Va1(6), De1(6,6),
*          dStresse1(6), AUXV11(6), AUXV21(6), Tstress(6),
*          Tstrain(6), dTstrain(6), TTstress(6), TTstrain(6),
*          Stress0(6), Va0(6), AUXV3(6), AUXV4(6), De0(6,6),
*          Tstress1(6), Va2(6), dStresse2(6), De2(6,6),
*          AUXV12(6), AUXV22(6), dStress2(6), dStress(6)
Parameter (ftol=1.D-9, Htol=1.D-6, EM=1.D-16,
*omegalim=15000, Rmax=100.0, Rmin=1.0,
*Pie=3.14159 26535 89793 23846 26433 83279 50288 41971 69399d0)

```

```

! Expected contents of Props(1..?)
! 1 : lambda   slope of ICL
! 2 : kappa    slope of U/R line
! 3 : M        slope of CSL in p-q plane
! 4 : nu       Poisson ratio
! 5 : evoid0   Initial void ration
! 6 : OCR      overconsolidation ratio
! 7 : POP      preoverburden pressure
! 8 : K0NC     ratio of horizontal and vertical stresses
! 9 : STOL     error tolerance in subincrementation algorithm

```

```
nStatV = 16
```

```

If (IDTask .Eq. 1) Then ! Initialize state variables
  if (StVar0(1).eq.0) then !Reset state variables
    ! Get Props
    xL=Props(1)
    xk=Props(2)
    xM=Props(3)
    xNu=Props(4)
    e0=Props(5)
    OCR=Props(6)
    POP=Props(7)
    xKNC=Props(8)
    ! Calculate Gama from OCR, POP
    Symax=-OCR*Sig0(2)+POP
    Sxmax=xKNC*Symax
    Szmax=xKNC*Symax
    pmax=(Symax+Sxmax+Szmax)/3
  
```

```

qmax=abs(Symax-Sxmax)
etamax=qmax/pmax
pi=-((Sig0(1)+Sig0(2)+Sig0(3))/3.0
qi=sqrt(3.0*(1.0/6.0*((Sig0(1)-Sig0(2))**2.0+
*(Sig0(2)-Sig0(3))**2.0+(Sig0(3)-Sig0(1))**2.0)+Sig0(4)**2.0+
*(Sig0(5)**2.0+Sig0(6)**2.0))
etai=qi/pi
POM1=(pmax*(etamax**2.0+xM**2.0)/2.0/xM**2.0)**((xl-xk)/xl)
POM2=pi**((xk/xl)
Gama=1.0+e0+xl*log(POM1*POM2)
PSIOC=1.0+e0+xl*log(pi)-Gama
PSINC=(xl-xk)*log(2.0*xM**2.0/(xM**2.0+etai**2.0))
R=exp((PSINC-PSIOC)/(xl-xk))
If (R.gt.Rmax) then
    Gama=1.0+e0+xl*log(pi)-PSINC+(xl-xk)*log(Rmax)
    R=Rmax
end if
if (R.lt.Rmin) then
    Gama=1.0+e0+xl*log(pi)-PSINC+(xl-xk)*log(Rmin)
    R=Rmin
end if
omega=(1.0+(PSINC-PSIOC)/PSINC)*R
p0=qi**2.0/(pi*xM**2.0)+pi
p0bs=R*p0
! Check if value of Poissons ratio is ok
If (xNu.Gt.0.5d0-1d-6 .Or. xNu.Lt.-1d-6) Then
    Call OK_MessageBox(' Invalid value for "nu" (param 4)')
    iAbort = 1
    Return
End If
if (xNu.Lt.0) then
    xNu = 0.0
    Props(4) = xNu
end if
if (xNu.Gt.0.495d0) then
    xNu = 0.495
    Props(4) = xNu
end if
! Fill state variable array
do i=1,nStatV
    StVar0(i)=0.0
end do
StVar0(1)=1.0
StVar0(2)=Gama
StVar0(3)=R
StVar0(4)=PSIOC
StVar0(5)=PSINC
StVar0(6)=omega
StVar0(7)=p0
StVar0(8)=p0bs
StVar0(9)=Symax
StVar0(10)=Symax/(-Sig0(2))
StVar0(12)=3.0
end if
If (iel+int.eq.2) Call WriVec(1, 'Props', Props, 8)
End If ! IDTask = 1

If (IDTask .Eq. 2) Then ! Calculate stresses
    If (IsUndr.Eq.1) Then
        dEpsV = dEps(1) + dEps(2) + dEps(3)
        dSwp = BulkW * dEpsV
        Swp = Swp0 + dSwp
    
```

```

Else
  Swp = Swp0
End If
If (iEL+Int+iter.Eq.3 .And. iStep.Lt.10) Then
  Call WriMat( -1, 'D66', D, 6, 6, 6 )
End If
! Material properties
xl=Props(1)
xk=Props(2)
xM=Props(3)
xNu=Props(4)
e0=Props(5)
! State variables
Gama=StVar0(2)
! Reversing sign for input stresses and strains
do i=1,6
  Stresst(i)=-Sig0(i)
  Straint(i)=-dEps(i+6)
  dStrain(i)=-dEps(i)
  Straintt(i)=Straint(i)+dStrain(i)
end do
!Stress and strain invariants
Call CalculateStressInvariants(Stresst,pt,qt,etat,thetat)
Call CalculateVoidRatio(Straint,e0,et)
!Material modules
Call CalculateElasticModuli(et,pt,xk,xNu,Bulk,G)
!Hardening parameter for curent yield surface
p0t=qt**2.0/pt/xM**2.0+pt
!Calculate OMEGA - hardening coefficient at the beginning of increment
Call CalculateOmega(xl,xk,xM,Gama,pt,qt,et,omegalim,Rmax,Rmin,
*omega,R,PSIOC,PSINC)
! Stress determination explicit modified euler with automatic subincrementation and
error control
!
! Initialization
STOL=Props(9)
if (STOL.lt.1.D-9) STOL=1.D-9
ipl=4
isuc=0
its=0
iss=0
subtime1=0.0
dsubtime=1.0
Call MZEROR(Tstress,6)
Call MZEROR(Tstrain,6)
Call MZEROR(dTstrain,6)
Call MZEROR(TTstress,6)
Call MZEROR(TTstrain,6)
Call COPYRVEC(Stresst,Tstress,6)
Call COPYRVEC(Straint,Tstrain,6)
do i=1,6
  dTstrain(i)=dStrain(i)*dsubtime
end do
Call AddVec(Tstrain,dTstrain,1.0,1.0,6,TTstrain)
! Stress calculation
110 continue
its=its+1
! Calculation of dStress1 and dp01
Call CalculateStressInvariants(Tstress,Tp,Tq,Teta,Ttheta)
Call CalculateVoidRatio(Tstrain,e0,Te)
Call CalculateVoidRatio(TTstrain,e0,TTe)
Call CalculateElasticModuli(Te,Tp,xk,xNu,TBulk1,TG1)

```



```

Tp0=Tq**2.0/Tp/xM**2.0+Tp
Call CalculateOmega(xl,xk,xM,Gama,Tp,Tq,Te,omegalim,Rmax,
*Rmin,Tomega1,TR1,TPSIOC1,TPSINC1)
Call MZEROR(Va1,6)
Call CalculateVectorA(Tp,Tp0,xM,Tstress,Va1)
Call MZEROR(De1,36)
Call CalculateDeMatrix(TBulk1,TG1,De1)
VarA1=(1.0+Te)*Tp0*Tomega1*Tp*(2.0*Tp-Tp0)/(xl-xk)
VarB1=(1.0+Te)*Tp0*Tomega1*(2.0*Tp-Tp0)/(xl-xk)
Call MZEROR(dStresse1,6)
Call MatVec(De1,6,dTstrain,6,dStresse1)
AUX11=0.0
AUX11=DInProd(Va1,dStresse1,6)
Call MZEROR(AUXV11,6)
Call MatVec(De1,6,Va1,6,AUXV11)
AUX21=0.0
AUX21=DInProd(AUXV11,Va1,6)
dlam1=AUX11/(VarA1+AUX21)
if (isnan(dlam1).or.dlam1.lt.0.0) dlam1=0.0
Call MZEROR(AUXV21,6)
Call MatVec(De1,6,Va1,6,AUXV21)
Call MZEROR(dStress1,6)
Call AddVec(dStresse1,AUXV21,1.0,-dlam1,6,dStress1)
Call MZEROR(Tstress1,6)
Call AddVec(TStress,dStress1,1.0,1.0,6,Tstress1)
dp01=dlam1*VarB1
Tp01=Tp0+dp01
! Calculation of dStress2 and dp02
Call CalculateStressInvariants(Tstress1,Tp1,Tq1,Teta1,
*Ttheta1)
Call CalculateElasticModuli(TTe,Tp1,xk,xNu,TBulk2,TG2)
Call CalculateOmega(xl,xk,xM,Gama,Tp1,Tq1,TTe,omegalim,Rmax,
*Rmin,Tomega2,TR2,TPSIOC2,TPSINC2)
Call MZEROR(Va2,6)
Call CalculateVectorA(Tp1,Tp01,xM,Tstress1,Va2)
Call MZEROR(De2,36)
Call CalculateDeMatrix(TBulk2,TG2,De2)
VarA2=(1.0+TTe)*Tp01*Tomega2*Tp1*(2.0*Tp1-Tp01)/(xl-xk)
VarB2=(1.0+TTe)*Tp01*Tomega2*(2.0*Tp1-Tp01)/(xl-xk)
Call MZEROR(dStresse2,6)
Call MatVec(De2,6,dTstrain,6,dStresse2)
AUX12=0.0
AUX12=DInProd(Va2,dStresse2,6)
Call MZEROR(AUXV12,6)
Call MatVec(De2,6,Va2,6,AUXV12)
AUX22=0.0
AUX22=DInProd(AUXV12,Va2,6)
dlam2=AUX12/(VarA2+AUX22)
if (isnan(dlam2).or.dlam2.lt.0.0) dlam2=0.0
Call MZEROR(AUXV22,6)
Call MatVec(De2,6,Va2,6,AUXV22)
Call MZEROR(dStress2,6)
Call AddVec(dStresse2,AUXV22,1.0,-dlam2,6,dStress2)
dp02=dlam2*VarB2
! Stresses and hardening parameter for modified Euler
Call MZEROR(dStress,6)
Call AddVec(dStress1,dStress2,0.5,0.5,6,dStress)
Call MZEROR(TTstress,6)
Call AddVec(Tstress,dStress,1.0,1.0,6,TTstress)
TTp0=Tp0+(dp01+dp02)/2.0
! Error estimation
Error1=sqrt((dStress2(1)-dStress1(1))**2.0+(dStress2(2)-

```



```

*dStress1(2)**2.0+(dStress2(3)-dStress1(3))**2.0+(dStress2(4)-
*dStress1(4))**2.0+(dStress2(5)-dStress1(5))**2.0+(dStress2(6)-
*dStress1(6))**2.0)/(2.0*sqrt(TTstress(1)**2.0+TTstress(2)**2.0+
*TTstress(2)**2.0+TTstress(3)**2.0+TTstress(4)**2.0+TTstress(6)**
*2.0))
Error2=abs(dp02-dp01)/(2.0*TTp0)
Error=Error1
if (Error2.gt.Error) Error=Error2
if (EM.gt.Error) Error=EM
if (Error.gt.STOL.and.dsubtime.gt.0.000001) then
stepsize=0.9*sqrt(STOL/Error)
if (stepsize.lt.0.1) stepsize=0.1
dsubtime=stepsize*dsubtime
if (dsubtime.lt.0.000001) dsubtime=0.000001
Call MZEROR(dTstrain,6)
do i=1,6
dTstrain(i)=dStrain(i)*dsubtime
end do
Call MZEROR(TTstrain,6)
Call AddVec(Tstrain,dTstrain,1.0,1.0,6,TTstrain)
isuc=1
goto 110
end if

Call CalculateStressInvariants(TTstress,TTp,TTq,TTeta,
*TTtheta)
TTf=TTq**2.0/xM**2.0+TTp*(TTp-TTp0)
if (dlam1.ne.0.0.and.dlam2.ne.0.0.and.abs(TTf).gt.ftol) then
do iDrift=1,10
Call MZEROR(Stress0,6)
Call COPYRVEC(TTstress,Stress0,6)
Call CalculateStressInvariants(Stress0,p,q,eta,
*theta)

p0=TTp0
f0=TTf
Call CalculateElasticModuli(TTe,p,xk,xNu,Bulk0,G0)
Call CalculateDeMatrix(Bulk0,G0,De0)
Call CalculateOmega(xl,xk,xM,Gama,p,q,TTe,omegalim,
*Rmax,Rmin,omega0,R0,PSIOC0,PSINC0)
Call MZEROR(Va0,6)
Call CalculateVectorA(p,p0,xM,Stress0,Va0)
VarA0=(1.0+TTe)*p0*omega0*p*(2.0*p-p0)/(xl-xk)
VarB0=(1.0+TTe)*p0*omega0*(2.0*p-p0)/(xl-xk)
Call MZEROR(AUXV3,6)
Call MatVec(De0,6,Va0,6,AUXV3)
AUX3=DInProd(AUXV3,Va0,6)
dlam=f0/(VarA0+AUX3)
Call MZEROR(AUXV4,6)
Call MatVec(De0,6,Va0,6,AUXV4)
Call MZEROR(TTstress,6)
Call AddVec(Stress0,AUXV4,1.0,-dlam,6,TTstress)
TTp0=p0+dlam*VarB0
Call CalculateStressInvariants(TTstress,TTp,TTq,
*TTeta,TTtheta)
TTf=TTq**2.0/xM**2.0+TTp*(TTp-TTp0)
If (abs(TTf).gt.abs(f0)) then
AUX4=DInProd(Va0,Va0,6)
dlam=f0/AUX4
Call MZEROR(TTstress,6)
Call AddVec(Stress0,Va0,1.0,-dlam,6,TTstress)
TTp0=p0
Call CalculateStressInvariants(TTstress,TTp,TTq,

```

```

*TTetatt, TTtheta)
      TTf=TTq**2.0/xM**2.0+TTp*(TTp-TTp0)
      end if
      if (abs(TTf).lt.ftol) exit
    end do
  end if
  iss=iss+1
  subtime2=subtime1+dsubtime
  stepsize=0.9*sqrt(STOL/Error)
  if (stepsize.gt.1.1) stepsize=1.1
  if (isuc.eq.1) then
    if (stepsize.gt.1.0) stepsize=1.0
  end if
  dsubtime=stepsize*dsubtime
  if (dsubtime.lt.0.000001) dsubtime=0.000001
  if ((subtime2+dsubtime).gt.1.0) dsubtime=1.0-subtime2
  isuc=0
  Call MZEROR(Tstress,6)
  Call COPYRVEC(TTstress,Tstress,6)
  Call MZEROR(Tstrain,6)
  Call COPYRVEC(TTstrain,Tstrain,6)
  Call MZEROR(dTstrain,6)
  do i=1,6
    dTstrain(i)=dStrain(i)*dsubtime
  end do
  Call MZEROR(TTstrain,6)
  Call AddVec(Tstrain,dTstrain,1.0,1.0,6,TTStrain)
  subtime1=subtime2
  if (subtime2.lt.1.0) goto 110

! Final stresses and hardening parameter
Call MZEROR(Stresstt,6)
Call COPYRVEC(TTstress,Stresstt,6)
p0tt=TTp0
!
!Return stresses
do i=1,6
  Sig(i)=-Stresstt(i)
end do
!Update state variables
StVar(1)=StVar0(1)
StVar(2)=StVar0(2)
StVar(3)=R
StVar(4)=PSIOC
StVar(5)=PSINC
StVar(6)=omega
StVar(7)=p0t
StVar(8)=p0t*R
If (StVar0(9).lt.Stresstt(2)) then
  StVar(9)=Stresstt(2)
else
  StVar(9)=StVar0(9)
end if
StVar(10)=StVar(9)/Stresstt(2)
StVar(11)=G
StVar(12)=3.0
StVar(13)=its
StVar(14)=iss
StVar(15)=subtime2
StVar(16)=Error
End If ! IDTask = 2
If ( IDTask .Eq. 3 .Or.

```

```

* IDTask .Eq. 6 ) Then ! Calculate D-Matrix
xk=Props(2)
xNu=Props(4)
e0=Props(5)
pt=-(Sig0(1)+Sig0(2)+Sig0(3))/3.0
EpsV=-(dEps(7)+dEps(8)+dEps(9))
et=(1.0+e0)*exp(-EpsV)-1.0
Call CalculateElasticModuli(et,pt,xk,xNu,Bulk,G)
D1=Bulk+4.0*G/3.0
D2=Bulk-2.0*G/3.0
D3=G
Call MZeroR(D,36)
D(1,1)=D1
D(2,2)=D1
D(3,3)=D1
D(1,2)=D2
D(1,3)=D2
D(2,1)=D2
D(2,3)=D2
D(3,1)=D2
D(3,2)=D2
D(4,4)=D3
D(5,5)=D3
D(6,6)=D3
BulkW = 0
If (IsUndr.Eq.1) Then
! BulkW = ...
xNu_U = 0.495d0
Fac=(1+xNu_U)/(1-2*xNu_U) - (1+xNu)/(1-2*xNu)
Fac=2D0*G/3D0 * Fac
BulkW = Fac
End If
End If ! IDTask = 3, 6

If (IDTask .Eq. 4) Then ! Number of state parameters
nStat = nStatV
End If ! IDTask = 4

If (IDTask .Eq. 5) Then ! matrix type
NonSym = 0 ! 1 for non-symmetric D-matrix
iStrsDep = 1 ! 1 for stress dependent D-matrix
iTang = 0 ! 1 for tangent D-matrix
iTimeDep = 0 ! 1 for time dependent D-matrix
End If ! IDTask = 5

Return
End ! MyModel3

```

MYMODEL4.for

```
Subroutine MyModel4 ( IDTask, iMod, IsUndr,
*                    iStep, iTer, iEl, Int,
*                    X, Y, Z,
*                    Time0, dTime,
*                    Props, Sig0, Swp0, StVar0,
*                    dEps, D, BulkW,
*                    Sig, Swp, StVar, ipl,
*                    nStat,
*                    NonSym, iStrsDep, iTimeDep, iTang,
*                    iAbort )
!
! Purpose: User supplied soil model, basic HASP, explicit fifth-order accurate (Runge-
Kutta-Dormand-Prince) with automatic subincrementation and error control
! Depending on IDTask, 1 : Initialize state variables
!                       2 : calculate stresses,
!                       3 : calculate material stiffness matrix
!                       4 : return number of state variables
!                       5 : inquire matrix properties
!                           return switch for non-symmetric D-matrix
!                           stress/time dependent matrix
!                       6 : calculate elastic material stiffness matrix
!
! Arguments:
!           I/O  Type
! IDTask      I   I   : see above
! iMod        I   I   : model number (1..10)
! IsUndr      I   I   : =1 for undrained, 0 otherwise
! iStep       I   I   : Global step number
! iter        I   I   : Global iteration number
! iel         I   I   : Global element number
! Int         I   I   : Global integration point number
! X           I   R   : X-Position of integration point
! Y           I   R   : Y-Position of integration point
! Z           I   R   : Z-Position of integration point
! Time0       I   R   : Time at start of step
! dTime       I   R   : Time increment
! Props       I   R( ) : List with model parameters
! Sig0        I   R( ) : Stresses at start of step
! Swp0        I   R   : Excess pore pressure start of step
! StVar0      I   R( ) : State variable at start of step
! dEps        I   R( ) : Strain increment
! D           I/O  R(, ) : Material stiffness matrix
! BulkW       I/O  R   : Bulkmodulus for water (undrained only)
! Sig         0   R( ) : Resulting stresses
! Swp         0   R   : Resulting excess pore pressure
! StVar       0   R( ) : Resulting values state variables
! ipl         0   I   : Plasticity indicator
! nStat       0   I   : Number of state variables
! NonSym      0   I   : Non-Symmetric D-matrix ?
! iStrsDep    0   I   : =1 for stress dependent D-matrix
! iTimeDep    0   I   : =1 for time dependent D-matrix
! iAbort      0   I   : =1 to force stopping of calculation
!
!           Implicit Double Precision (A-H, O-Z)
!
!           Dimension Props(*), Sig0(*), StVar0(*), dEps(*), D(6,6),
*                   Sig(*), StVar(*)
!
!----- Local variables (Arrays)
!
```

```

! AUXV1X(6)      : Array which contains the product of Va and De (transpose(Va) * De)
(RKDP) (X=1,2,3,4,5,6) (IDTask=2)
! AUXV2X(6)      : Array which contains the product of De and Va (De * Va) (RKDP)
(X=1,2,3,4,5,6) (IDTask=2)
! AUXV3(6)       : Array which contains the product (transpose(Va0) * De) (IDTask=2)
! AUXV4(6)       : Array which contains the product (De * Va0) (IDTask=2)
! dStresseX(6)   : Array of elastic stress increments (RKDP) (X=1,2,3,4,5,6)
(IDTask=2)
! dStrain(6)     : Array of strain increments (compression is positive) (IDTask=2)
! dStressX(6)    : Array of stress increments (compression is positive)
(X=1,2,3,4,5,6) (modEuler) (IDTask=2)
! dTstrain      : Array of strain increments in one subincrement (IDTask=2)
! Straint(6)    : Array of strains at start of step (compression is positive)
(IDTask=2)
! Straintt(6)   : Array of strains at the end of step (compression is positive)
(IDTask=2)
! Stresst(6)    : Array of stresses at start of step (compression is positive)
(IDTask=2)
! Stresstt(6)   : Array of stresses at the end of step (compression is positive)
(IDTask=2)
! Stress0(6)    : Array of uncorrected stresses - drift correction (IDTask=2)
! Tstrain(6)    : Array of strains at start of subincrement (IDTask=2)
! Tstress(6)    : Array of stresses at start of subincrements (IDTask=2)
! StressX(6)    : Array of stresses (X=1,2,3,4,5,6) (RKDP) (IDTask=2)
! TTstrain(6)   : Array of strains at the end of subincrement (IDTask=2)
! TTstress(6)   : Array of stresset at the end of subincrement (IDTask=2)
! VaX(6)        : Array of yield function derivatives (RKDP) (X=1,2,3,4,5,6)
(IDTask=2)
! Va0(6)        : Array of yield function derivatives for uncorrected stresses -
drift correction (IDTask=2)

!
!
!--- Local variables (Matrices)
!
! De0(6,6)      : Elastic De matrix in drift correction algorithm (IDTask=2)
! DeX(6,6)      : Elastic De matrix (RKDP) (X=1,2,3,4,5,6) (IDTask=2)
!
!--- Local variables
!
! AUX1X         : Auxilliary real variable, transpose(Va) * dStresse (RKDP)
(X=1,2,3,4,5,6) (IDTask=2)
! AUX2X         : Auxilliary real variable, transpose(Va) * De * Va (RKDP)
(X=1,2,3,4,5,6) (IDTask=2)
! AUX3         : Auxilliary real variable, transpose(Va0) * De * Va0 (IDTask=2)
! AUX4         : Auxilliary real variable, transpose(Va0) * Va0
! Bulk         : Bulk modulus at start of step (IDTask=2, IDTask=4, IDTask=6)
! BulkX        : Bulk modulus in RKDP algorithm (IDTask=2)
! BulkW        : Bulk modulus of water (IDTask=4, IDTask=6)
! Bulk0        : Bulk modulus in drift correction algorithm (IDTask=2)
! dEv          : Increment of volumetric strain during one subincrement
! dEpsV        : Increment of volumetric strain (IDTask=2)
! dlam         : Plastic multiplier (IDTask=2)
! dlamX        : Plastic multiplier (RKDP) (X=1,2,3,4,5,6) (IDTask=2)
! dp0X        : Increment of hardening parameter (RKDP) (X=1,2,3,4,5,6) (IDTask=2)
! dShearStrain : Increment of shear strain invariant (IDTask=2)
! dsubtime     : size of subincrement (IDTask=2)
! dSwp         : Increment of excess pore water pressure (IDTask=2)
! D1,D2 and D3 : Auxilliary variables for calculation od D matrix (IDTask=4,
IDTask=6)
! EM          : Machine Error tolerance (IDTask=2)

```

```

! EpsV      : Volumetric strain (IDTask=3, IDTask=6)
! Error     : Error estimation in automatic subincrementation (IDTask=2)
! Error1    : Error estimation in stresses determination in automatic
subincrementation (IDTask=2)
! Error2    : Error estimation in hardening parameter determination in automatic
subincrementation (IDTask=2)
! et        : Void ratio at start of step (IDTask=2, IDTask=4, IDTask=6)
! eta       : Stress ratio (IDTask=2)
! etai      : Initial stress ratio (IDTask=1)
! etamax    : Max previous stress ratio (IDTask=1)
! etat      : Shear ratio at start of step (IDTask=2)
! etaX      : Shear ratio in RKDP algorithm (IDTask=2)
! e0        : Initial void ratio (IDTask=1, IDTask=2, IDTask=4, IDTask=6)
! eX        : Void ratio in RKDP algorithm (X=2,3,4,5,6) (IDTask=2)
! ftol      : Tolerance for zero value of yield function (IDTask=2)
! f0        : Value of yield function that must be corrected - drift correction
(IDTask=2)
! G         : Shear modulus at start of step (IDTask=2, IDTask=4, IDTask=6)
! GX        : Shear modulus in RKDM algorithm (X=1,2,3,4,5,6) (IDTask=2)
! G0        : Shear modulus in drift correction algorithm (IDTask=2)
! Gama      : Specific volume for CLS at p'=1 kPa (IDTask=1, IDTask=2)
! iDrift    : Counter in drift correction loop (IDTask=2)
! isuc      : =0 if previous subincrement is successful, =1 if not (IDTask=2)
! iss       : Counts the number of successful subincrements (IDTask=2)
! its       : Counts the number of total subincrements (IDTask=2)
! Nsubs     : Number of subincrements (IDTask=2)
! nStatV    : Number of used state variables
! OCR       : Input value of overconsolidation ratio (IDTask=1)
! omega     : Hardening coefficient (IDTask=1, IDTask=2)
! omegaX    : Hardening coefficient in RKDP (IDTask=2)
! omegalm   : Limiting value for hardening coefficient in order to avoid
numerical instability (IDTask=2)
! omega0    : Hardening coefficient in drift correction algorithm (IDTask=2)
! p         : mean effective stress (IDTask=2)
! pX        : mean effective stress in RKDP algorithm (X=1,2,3,4,5,6) (IDTask=2)
! pi        : Initial value of mean effective stress
! Pie       : Value of Pi (IDTask=1)
! pmax      : Max previous mean effective stress (IDTask=1)
! POM1, POM2, ... : Auxilliary real variables
! POP       : Input value of preoverburden pressure (IDTask=1)
! pt        : Mean effective stress at start of step (IDTask=2, IDTask=4,
IDTask=6)
! ptt       : mean effective stress at the end of step (IDTask=2)
! PSINC     : State parameter for NC soil (IDTask=1, IDTask=2)
! PSINCX    : State parameter for NC soil in RKDP algorithm (X=1,2,3,4,5,6)
(IDTask=2)
! PSINC0    : State parameter for NC soil in drift correction algorithm
(IDTask=2)
! PSIOC     : State parameter for OC soil ((IDTask=1, IDTask=2)
! PSIOCX    : State parameter for OC soil in RKDP algorithm (X=1,2,3,4,5,6)
(IDTask=2)
! PSIOC0    : State parameter for OC soil in drift correction algorithm
(IDTask=2)
! p0        : Hardening parameter for yield surface (IDTask=1, IDTask=2)
! p0bs      : Hardening parameter for bounding surface (IDTask=1, IDTask=2)
! p0t       : Hardening parameter for current yield surface at start of step
(IDTask=2)
! p0tt      : Hardening parameter for current yield surface at start of step
(IDTask=2)
! p0X       : Hardening parameter for current yield surface in RKDP algorithm
(X=1,2,3,4,5,6) (IDTask=2)
! q         : Shear deviator stress (IDTask=2)

```

```

! qX      : Shear deviator stress in RKDP algorithm (X=1,2,3,4,5,6) (IDTask=2)
! qi      : Initial value of shear deviator stress (IDTask=1)
! qmax    : Max previous shear deviator stress (IDTask=1)
! qt      : Shear deviator stress at start of step (IDTask=2)
! qtt     : Shear deviator stress at the end of step (IDTask=2)
! R       : Isotropic OCR ratio (IDTask=1, IDTask=2)
! Rmax    : Max value of R in order to avoid numerical instability (IDTask=1,
IDTask=2)
! Rmin    : Min value of R in order to avoid non-possible values (IDTask=1,
IDTask=2)
! RX      : Isotropic OCR ratio in RKDP algorithm (X=1,2,3,4,5,6) (IDTask=2)
! R0      : Isotropic OCR ratio in drift correction algorithm (IDTask=2)
! stepsize : Relative size of next subincrement size to the previous
subincrement size (IDTask=2)
! subtime1 : Time at the begining of subincrement
! subtime2 : Time at the end of subincrement
! STOL     : Error tolerance for evaluation of subincrement size (IDTask=2)
! Sxmax   : Max previous horizontal stress in x direction (IDTask=1)
! Symax   : Max previous vertical stress (IDTask=1)
! Szmax   : Max previous horizontal stress in y direction (IDTask=1)
! Te      : Void ratio at start of subincrement
! theta   : Lodes angle (IDTask=2)
! thetat  : Lodes angle at start of step (IDTask=2)
! thetaX  : Lodes angle in RKDP algorithm (X=1,2,3,4,5,6) (IDTask=2)
! TTe     : Void ratio at the end of subincrement (IDTask=2)
! TTf     : Yield function at the end of subincrement (IDTask=2)
! TTp     : Mean effective stress at start of subincrement (IDTask=2)
! TTp0    : Hardening parameter at the end of subincrement (IDTask=2)
! TTq     : Deviator stress at the end of subincrement (IDTask=2)
! TTtheta : Lodes angle at the end of subincrement (IDTask=2)
! VarA0   : Variable A in stress integration procedure (drift correction)
(IDTask=2)
! VarAX   : Variable A in stress integration procedure (RKDP) (X=1,2,3,4,5,6)
(IDTask=2)
! VarB0   : Variable B in stress integration procedure (drift correction)
(IDTask=2)
! VarBX   : Variable B in stress integration procedure (RKDP) (X=1,2,3,4,5,6)
(IDTask=2)
! xk      : kappa, slope of U/R line (IDTask=1, IDTask=2, IDTask=4, IDTask=6)
! xKNC    : Ratio of horizontal and vertical stresses at NC state (IDTask=1)
! xl      : Lambda, slope of ICL (IDTask=1, IDTask=2, IDTask=4, IDTask=6))
! xM      : M, slope of CSL in p-q plane (IDTask=1, IDTask=2)
! xNu     : Poissons ratio (IDTask=1, IDTask=2, IDTask=4, IDTask=6)
!
!--- Auxilliary Subroutines
!
! CalculateStressInvariants : Calculates p,q,eta and theta
! CalculateVoidRatio       : Calculates current void ratio
! CalculateElasticModuli   : Calculates bulk and shear modulus at start of step
! CalculateOmega           : Calculates hardening coefficient
! CalculateVectorA         : Calculates vector A which contains derivatives of
yield function
! CalculateDeMatric        : Calculates elastic D matrix
! MZEROR                   : Fills real array with zeros
! MatVec                   : Multiplies matrix with a vector. Result is vector
! DInProd                  : Inner product of two vectors. Result is scalar
! AddVec                   : Calculates sum of two vectors. result is vector
! COPYRVEC                 : Copies a Double array R1 with Dimension K to R2
!
! Dimension Stresst(6), Straint(6), dStrain(6), Straintt(6),
* dStress1(6), Stresstt(6), dSig(6), Va1(6), De1(6,6),
* dStresse1(6), AUXV11(6), AUXV21(6), Tstress(6),

```



```

*      Tstrain(6), dTstrain(6), TTstress(6), TTstrain(6),
*      Stress0(6), Va0(6), AUXV3(6), AUXV4(6), De0(6,6),
*      Tstress1(6), Va2(6), dStresse2(6), De2(6,6),
*      AUXV12(6), AUXV22(6), dStress2(6), dStress(6),
*      Stress1(6), Stress2(6), Stress3(6), Va3(6), De3(6,6),
*      dStresse3(6), AUXV13(6), AUXV23(6), dStress3(6),
*      Stress4(6), Va4(6), De4(6,6), dStresse4(6), AUXV14(6),
*      AUXV24(6), dStress4(6), Stress5(6), Va5(6), De5(6,6),
*      dStresse5(6), AUXV15(6), AUXV25(6), dStress5(6),
*      Stress6(6), Va6(6), De6(6,6), dStresse6(6), AUXV16(6),
*      AUXV26(6), dStress6(6), StressError(6)
Parameter (ftol=1.D-9, Htol=1.D-5, EM=1.D-16,
*omegalim=15000, Rmax=100.0, Rmin=1.0,
*Pie=3.14159 26535 89793 23846 26433 83279 50288 41971 69399d0)

```

```

! Expected contents of Props(1..?)
! 1 : lambda   slope of ICL
! 2 : kappa    slope of U/R line
! 3 : M        slope of CSL in p-q plane
! 4 : nu       Poisson ratio
! 5 : evoid0   Initial void ration
! 6 : OCR      overconsolidation ratio
! 7 : POP      preoverburden pressure
! 8 : K0NC     ratio of horizontal and vertical stresses
! 9 : STOL     Error tolerance for substepping

```

```

nStatV = 16
If (IDTask .Eq. 1) Then ! Initialize state variables
  if (StVar0(1).eq.0) then !Reset state variables
    ! Get Props
    xl=Props(1)
    xk=Props(2)
    xM=Props(3)
    xNu=Props(4)
    e0=Props(5)
    OCR=Props(6)
    POP=Props(7)
    xKNC=Props(8)
    ! Calculate Gama from OCR, POP
    Symax=-OCR*Sig0(2)+POP
    Sxmax=xKNC*Symax
    Szmax=xKNC*Symax
    pmax=(Symax+Sxmax+Szmax)/3
    qmax=abs(Symax-Sxmax)
    etamax=qmax/pmax
    pi=-(Sig0(1)+Sig0(2)+Sig0(3))/3.0
    qi=sqrt(3.0*(1.0/6.0*((Sig0(1)-Sig0(2))**2.0+
*(Sig0(2)-Sig0(3))**2.0+(Sig0(3)-Sig0(1))**2.0)+Sig0(4)**2.0+
*Sig0(5)**2.0+Sig0(6)**2.0))
    etai=qi/pi
    POM1=(pmax*(etamax**2.0+xM**2.0)/2.0/xM**2.0)**((xl-xk)/xl)
    POM2=pi**((xk/xl)
    Gama=1.0+e0+xl*log(POM1*POM2)
    PSIOC=1.0+e0+xl*log(pi)-Gama
    PSINC=(xl-xk)*log(2.0*xM**2.0/(xM**2.0+etai**2.0))
    R=exp((PSINC-PSIOC)/(xl-xk))
    If (R.gt.Rmax) then
      Gama=1.0+e0+xl*log(pi)-PSINC+(xl-xk)*log(Rmax)
      R=Rmax
    end if
    if (R.lt.Rmin) then

```



```

        Gama=1.0+e0*xl*log(pi)-PSINC+(xl-xk)*log(Rmin)
        R=Rmin
    end if
    omega=(1.0+(PSINC-PSIOC)/PSINC)*R
    p0=qi**2.0/(pi*xM**2.0)+pi
    p0bs=R*p0
    ! Check if value of Poissons ratio is ok
    If (xNu.Gt.0.5d0-1d-6 .Or. xNu.Lt.-1d-6) Then
        Call OK_MessageBox(' Invalid value for "nu" (param 4)')
        iAbort = 1
        Return
    End If
    if (xNu.Lt.0) then
        xNu = 0.0
        Props(4) = xNu
    end if
    if (xNu.Gt.0.495d0) then
        xNu = 0.495
        Props(4) = xNu
    end if
    ! Fill state variable array
    do i=1,nStatV
        StVar0(i)=0.0
    end do
    StVar0(1)=1.0
    StVar0(2)=Gama
    StVar0(3)=R
    StVar0(4)=PSIOC
    StVar0(5)=PSINC
    StVar0(6)=omega
    StVar0(7)=p0
    StVar0(8)=p0bs
    StVar0(9)=Symax
    StVar0(10)=Symax/(-Sig0(2))
    StVar0(12)=4.0
    end if
    If (iel+int.eq.2) Call WriVec(1, 'Props', Props, 8)
End If ! IDTask = 1

If (IDTask .Eq. 2) Then ! Calculate stresses
    If (IsUndr.Eq.1) Then
        dEpsV = dEps(1) + dEps(2) + dEps(3)
        dSwp = BulkW * dEpsV
        Swp = Swp0 + dSwp
    Else
        Swp = Swp0
    End If
    If (iEL+Int+iter.Eq.3 .And. iStep.Lt.10) Then
        Call WriMat( -1, 'D66', D, 6, 6, 6 )
    End If
! Material properties
    xl=Props(1)
    xk=Props(2)
    xM=Props(3)
    xNu=Props(4)
    e0=Props(5)
! State variables
    Gama=StVar0(2)
! Reversing sign for input stresses and strains
    do i=1,6
        Stresst(i)=-Sig0(i)
        Straint(i)=-dEps(i+6)
    end do

```

```

        dStrain(i)=-dEps(i)
        Straintt(i)=Straint(i)+dStrain(i)
    end do
!Stress and strain invariants
    Call CalculateStressInvariants(Stresst,pt,qt,etat,thetat)
    Call CalculateVoidRatio(Straint,e0,et)
!Material modules
    Call CalculateElasticModuli(et,pt,xk,xNu,Bulk,G)
!Hardening parameter for curent yield surface
    p0t=qt**2.0/pt/xM**2.0+pt
!Calculate OMEGA - hardening coefficient at the beginning of increment
    Call CalculateOmega(xl,xk,xM,Gama,pt,qt,et,omegalim,Rmax,Rmin,
*omega,R,PSIOC,PSINC)
! Stress determination explicit fifth-order accurate (Runge-Kutta-Dormand-Price) with
automatic subincrementation and error control
!
! Initialization
    STOL=Props(9)
    if (STOL.lt.1.D-9) STOL=1.D-9
    ipl=4
    isuc=0
    its=0
    iss=0
    subtime1=0.0
    dsubtime=1.0
    Call MZEROR(Tstress,6)
    Call MZEROR(Tstrain,6)
    Call MZEROR(dTstrain,6)
    Call MZEROR(TTstress,6)
    Call MZEROR(TTstrain,6)
    Call COPYRVEC(Stresst,Tstress,6)
    Call COPYRVEC(Straint,Tstrain,6)
    do i=1,6
        dTstrain(i)=dStrain(i)*dsubtime
    end do
    Call AddVec(Tstrain,dTstrain,1.0,1.0,6,TTstrain)
    dEv=dTstrain(1)+dTstrain(2)+dTstrain(3)
! Stress calculation for one subincrement
110 continue
    its=its+1
        ! Calculation of dStress1 and dp01
        Call MZEROR(Stress1,6)
        Call COPYRVEC(Tstress,Stress1,6)
        Call CalculateStressInvariants(Stress1,p1,q1,eta1,theta1)
        Call CalculateVoidRatio(Tstrain,e0,Te)
        Call CalculateVoidRatio(TTstrain,e0,TTe)
        Call CalculateElasticModuli(Te,p1,xk,xNu,Bulk1,G1)
        p01=q1**2.0/p1/xM**2.0+p1
        Call CalculateOmega(xl,xk,xM,Gama,p1,q1,Te,omegalim,Rmax,
*Rmin,omega1,R1,PSIOC1,PSINC1)
        Call MZEROR(Va1,6)
        Call CalculateVectorA(p1,p01,xM,Stress1,Va1)
        Call MZEROR(De1,36)
        Call CalculateDeMatrix(Bulk1,G1,De1)
        VarA1=(1.0+Te)*p01*omega1*p1*(2.0*p1-p01)/(xl-xk)
        VarB1=(1.0+Te)*p01*omega1*(2.0*p1-p01)/(xl-xk)
        Call MZEROR(dStresse1,6)
        Call MatVec(De1,6,dTstrain,6,dStresse1)
        AUX11=0.0
        AUX11=DInProd(Va1,dStresse1,6)
        Call MZEROR(AUXV11,6)
        Call MatVec(De1,6,Va1,6,AUXV11)

```

```

AUX21=0.0
AUX21=DInProd(AUXV11,Va1,6)
dlam1=AUX11/(VarA1+AUX21)
if (isnan(dlam1).or.dlam1.lt.0.0) dlam1=0.0
Call MZEROR(AUXV21,6)
Call MatVec(De1,6,Va1,6,AUXV21)
Call MZEROR(dStress1,6)
Call AddVec(dStresse1,AUXV21,1.0,-dlam1,6,dStress1)
dp01=dlam1*VarB1
Call MZEROR(Stress2,6)
Call AddVec(Stress1,dStress1,1.0,0.2,6,Stress2)
p02=p01+0.2*dp01
e2=(1.0+Te)*exp(-0.2*dEv)-1.0
if (e2.lt.0.0) e2=0.0
if (e2.gt.1000.0) e2=1000.0
! Calculation of dStress2 and dp02
Call CalculateStressInvariants(Stress2,p2,q2,eta2,
*theta2)
Call CalculateElasticModuli(e2,p2,xk,xNu,Bulk2,G2)
Call CalculateOmega(xl,xk,xM,Gama,p2,q2,e2,omegalim,Rmax,
*Rmin,omega2,R2,PSIOC2,PSINC2)
Call MZEROR(Va2,6)
Call CalculateVectorA(p2,p02,xM,Stress2,Va2)
Call MZEROR(De2,36)
Call CalculateDeMatrix(Bulk2,G2,De2)
VarA2=(1.0+e2)*p02*omega2*p2*(2.0*p2-p02)/(xl-xk)
VarB2=(1.0+e2)*p02*omega2*(2.0*p2-p02)/(xl-xk)
Call MZEROR(dStresse2,6)
Call MatVec(De2,6,dTstrain,6,dStresse2)
AUX12=0.0
AUX12=DInProd(Va2,dStresse2,6)
Call MZEROR(AUXV12,6)
Call MatVec(De2,6,Va2,6,AUXV12)
AUX22=0.0
AUX22=DInProd(AUXV12,Va2,6)
dlam2=AUX12/(VarA2+AUX22)
if (isnan(dlam2).or.dlam2.lt.0.0) dlam2=0.0
Call MZEROR(AUXV22,6)
Call MatVec(De2,6,Va2,6,AUXV22)
Call MZEROR(dStress2,6)
Call AddVec(dStresse2,AUXV22,1.0,-dlam2,6,dStress2)
dp02=dlam2*VarB2
Call MZEROR(Stress3,6)
do j=1,6
Stress3(j)=Stress1(j)+3.0/40.0*dStress1(j)+9.0/40.0*
*dStress2(j)
end do
p03=p01+3.0/40.0*dp01+9.0/40.0*dp02
e3=(1.0+Te)*exp(-(3.0/40+9.0/40)*dEv)-1.0
if (e3.lt.0.0) e3=0.0
if (e3.gt.1000.0) e3=1000.0
! Calculation of dStress3 and dp03
Call CalculateStressInvariants(Stress3,p3,q3,eta3,
*theta3)
Call CalculateElasticModuli(e3,p3,xk,xNu,Bulk3,G3)
Call CalculateOmega(xl,xk,xM,Gama,p3,q3,e3,omegalim,Rmax,
*Rmin,omega3,R3,PSIOC3,PSINC3)
Call MZEROR(Va3,6)
Call CalculateVectorA(p3,p03,xM,Stress3,Va3)
Call MZEROR(De3,36)
Call CalculateDeMatrix(Bulk3,G3,De3)
VarA3=(1.0+e3)*p03*omega3*p3*(2.0*p3-p03)/(xl-xk)

```

```

VarB3=(1.0+e3)*p03*omega3*(2.0*p3-p03)/(xl-xk)
Call MZEROR(dStresse3,6)
Call MatVec(De3,6,dTstrain,6,dStresse3)
AUX13=0.0
AUX13=DInProd(Va3,dStresse3,6)
Call MZEROR(AUXV13,6)
Call MatVec(De3,6,Va3,6,AUXV13)
AUX23=0.0
AUX23=DInProd(AUXV13,Va3,6)
dlam3=AUX13/(VarA3+AUX23)
if (isnan(dlam3).or.dlam3.lt.0.0) dlam3=0.0
Call MZEROR(AUXV23,6)
Call MatVec(De3,6,Va3,6,AUXV23)
Call MZEROR(dStress3,6)
Call AddVec(dStresse3,AUXV23,1.0,-dlam3,6,dStress3)
dp03=dlam3*VarB3
Call MZEROR(Stress4,6)
do j=1,6
    Stress4(j)=Stress1(j)+3.0/10.0*dStress1(j)-9.0/10.0*
*dStress2(j)+6.0/5.0*dStress3(j)
end do
p04=p01+3.0/10.0*dp01-9.0/10.0*dp02+6.0/5.0*dp03
e4=(1.0+Te)*exp(-(3.0/10-9.0/10+6.0/5.0)*dEv)-1.0
if (e4.lt.0.0) e4=0.0
if (e4.gt.1000.0) e4=1000.0
! Calculation of dStress4 and dp04
Call CalculateStressInvariants(Stress4,p4,q4,eta4,
*theta4)
Call CalculateElasticModuli(e4,p4,xk,xNu,Bulk4,G4)
Call CalculateOmega(xl,xk,xM,Gama,p4,q4,e4,omegaLim,Rmax,
*Rmin,omega4,R4,PSIOC4,PSINC4)
Call MZEROR(Va4,6)
Call CalculateVectorA(p4,p04,xM,Stress4,Va4)
Call MZEROR(De4,36)
Call CalculateDeMatrix(Bulk4,G4,De4)
VarA4=(1.0+e4)*p04*omega4*p4*(2.0*p4-p04)/(xl-xk)
VarB4=(1.0+e4)*p04*omega4*(2.0*p4-p04)/(xl-xk)
Call MZEROR(dStresse4,6)
Call MatVec(De4,6,dTstrain,6,dStresse4)
AUX14=0.0
AUX14=DInProd(Va4,dStresse4,6)
Call MZEROR(AUXV14,6)
Call MatVec(De4,6,Va4,6,AUXV14)
AUX24=0.0
AUX24=DInProd(AUXV14,Va4,6)
dlam4=AUX14/(VarA4+AUX24)
if (isnan(dlam4).or.dlam4.lt.0.0) dlam4=0.0
Call MZEROR(AUXV24,6)
Call MatVec(De4,6,Va4,6,AUXV24)
Call MZEROR(dStress4,6)
Call AddVec(dStresse4,AUXV24,1.0,-dlam4,6,dStress4)
dp04=dlam4*VarB4
Call MZEROR(Stress5,6)
do j=1,6
    Stress5(j)=Stress1(j)+226.0/729.0*dStress1(j)-
*25.0/27.0*dStress2(j)+880.0/729.0*dStress3(j)+55.0/729.0*
*dStress4(j)
end do
p05=p01+226.0/729.0*dp01-25.0/27.0*dp02+880.0/729.0*dp03+
*55.0/729.0*dp04
e5=(1.0+Te)*exp(-(226.0/729.0-25.0/27.0+880.0/729.0+55.0/
*729.0)*dEv)-1.0

```

```

    if (e5.lt.0.0) e5=0.0
    if (e5.gt.1000.0) e5=1000.0
    ! Calculation of dStress5 and dp05
    Call CalculateStressInvariants(Stress5,p5,q5,eta5,
*theta5)
    Call CalculateElasticModuli(e5,p5,xk,xNu,Bulk5,G5)
    Call CalculateOmega(xl,xk,xM,Gama,p5,q5,e5,omegalim,Rmax,
*Rmin,omega5,R5,PSIOC5,PSINC5)
    Call MZEROR(Va5,6)
    Call CalculateVectorA(p5,p05,xM,Stress5,Va5)
    Call MZEROR(De5,36)
    Call CalculateDeMatrix(Bulk5,G5,De5)
    VarA5=(1.0+e5)*p05*omega5*p5*(2.0*p5-p05)/(xl-xk)
    VarB5=(1.0+e5)*p05*omega5*(2.0*p5-p05)/(xl-xk)
    Call MZEROR(dStresse5,6)
    Call MatVec(De5,6,dTstrain,6,dStresse5)
    AUX15=0.0
    AUX15=DInProd(Va5,dStresse5,6)
    Call MZEROR(AUXV15,6)
    Call MatVec(De5,6,Va5,6,AUXV15)
    AUX25=0.0
    AUX25=DInProd(AUXV15,Va5,6)
    dlam5=AUX15/(VarA5+AUX25)
    if (isnan(dlam5).or.dlam5.lt.0.0) dlam5=0.0
    Call MZEROR(AUXV25,6)
    Call MatVec(De5,6,Va5,6,AUXV25)
    Call MZEROR(dStress5,6)
    Call AddVec(dStresse5,AUXV25,1.0,-dlam5,6,dStress5)
    dp05=dlam5*VarB5
    Call MZEROR(Stress6,6)
    do j=1,6
        Stress6(j)=Stress1(j)-181.0/270.0*dStress1(j)+
*5.0/2.0*dStress2(j)-266.0/297.0*dStress3(j)-91.0/27.0*
*dStress4(j)+189.0/55.0*dStress5(j)
    end do
    p06=p01-181.0/270.0*dp01+5.0/2.0*dp02-266.0/297.0*dp03-
*91.0/27.0*dp04+189.0/55.0*dp05
    e6=(1.0+Te)*exp(-(-181.0/270.0+5.0/2.0-266.0/297.0-91.0/27.0
*+189.0/55.0)*dEv)-1.0
    if (e6.lt.0.0) e6=0.0
    if (e6.gt.1000.0) e6=1000.0
    !Calculation of dStress6 and dp06
    Call CalculateStressInvariants(Stress6,p6,q6,eta6,
*theta6)
    Call CalculateElasticModuli(e6,p6,xk,xNu,Bulk6,G6)
    Call CalculateOmega(xl,xk,xM,Gama,p6,q6,e6,omegalim,Rmax,
*Rmin,omega6,R6,PSIOC6,PSINC6)
    Call MZEROR(Va6,6)
    Call CalculateVectorA(p6,p06,xM,Stress6,Va6)
    Call MZEROR(De6,36)
    Call CalculateDeMatrix(Bulk6,G6,De6)
    VarA6=(1.0+e6)*p06*omega6*p6*(2.0*p6-p06)/(xl-xk)
    VarB6=(1.0+e6)*p06*omega6*(2.0*p6-p06)/(xl-xk)
    Call MZEROR(dStresse6,6)
    Call MatVec(De6,6,dTstrain,6,dStresse6)
    AUX16=0.0
    AUX16=DInProd(Va6,dStresse6,6)
    Call MZEROR(AUXV16,6)
    Call MatVec(De6,6,Va6,6,AUXV16)
    AUX26=0.0
    AUX26=DInProd(AUXV16,Va6,6)
    dlam6=AUX16/(VarA6+AUX26)

```

```

    if (isnan(dlam6).or.dlam6.lt.0.0) dlam6=0.0
    Call MZEROR(AUXV26,6)
    Call MatVec(De6,6,Va6,6,AUXV26)
    Call MZEROR(dStress6,6)
    Call AddVec(dStresse6,AUXV26,1.0,-dlam6,6,dStress6)
    dp06=dlam6*VarB6
! Stresses and hardening parameter for 5th order
    Call MZEROR(TTstress,6)
    do j=1,6
        TTstress(j)=Stress1(j)+19.0/216.0*dStress1(j)+
*1000.0/2079.0*dStress3(j)-125.0/216.0*dStress4(j)+
*81.0/88.0*dStress5(j)+5.0/56.0*dStress6(j)
    end do
    TTp0=p01+19.0/216.0*dp01+1000.0/2079.0*dp03-
*125.0/216.0*dp04+81.0/88.0*dp05+5.0/56.0*dp06
    ! Error estimation
    Call MZEROR(StressError,6)
    do j=1,6
        StressError(j)=11.0/360.0*dStress1(j)-10.0/63.0*
*dStress3(j)+55.0/72.0*dStress4(j)-27.0/40.0*dStress5(j)+
*11.0/280.0*dStress6(j)
    end do
    p0Error=11.0/360.0*dp01-11.0/63.0*dp03+55.0/72.0*dp04-
*27.0/40.0*dp05+11.0/280.0*dp06
    Error1=sqrt(StressError(1)**2.0+StressError(2)**2.0+
*StressError(3)**2.0+StressError(4)**2.0+StressError(5)**2.0+
*StressError(6)**2.0)/sqrt(TTstress(1)**2.0+TTstress(2)**2.0+
*TTstress(3)**2.0+TTstress(4)**2.0+TTstress(5)**2.0+
*TTstress(6)**2.0)
    Error2=abs(p0Error)/TTp0
    Error=Error1
    if (Error2.gt.Error) Error=Error2
    if (EM.gt.Error) Error=EM
    if (Error.gt.STOL.and.dsubtime.gt.0.000001) then
        stepsizem=0.9*((STOL/Error)**(1.0/5.0))
        if (stepsizem.lt.0.1) stepsizem=0.1
        dsubtime=stepsizem*dsubtime
        if (dsubtime.lt.0.000001) dsubtime=0.000001
        Call MZEROR(dTstrain,6)
        do i=1,6
            dTstrain(i)=dStrain(i)*dsubtime
        end do
        Call MZEROR(TTstrain,6)
        Call AddVec(Tstrain,dTstrain,1.0,1.0,6,TTstrain)
        dEv=dTstrain(1)+dTstrain(2)+dTstrain(3)
        isuc=1
        goto 110
    end if
    Call CalculateStressInvariants(TTstress,TTp,TTq,TTeta,
*TTtheta)
    TTf=TTq**2.0/xM**2.0+TTp*(TTp-TTp0)
    if (dlam1.ne.0.0.and.dlam2.ne.0.0.and.dlam3.ne.0.0.and.
*dlam4.ne.0.0.and.dlam5.ne.0.0.and.dlam6.ne.0.0.and.
*abs(TTf).gt.ftol) then
        do iDrift=1,10
            Call MZEROR(Stress0,6)
            Call COPYRVEC(TTstress,Stress0,6)
            Call CalculateStressInvariants(Stress0,p,q,eta,
*theta)
            p0=TTp0
            f0=TTf
            Call CalculateElasticModuli(TTe,p,xk,xNu,Bulk0,G0)

```

```

        Call CalculateDeMatrix(Bulk0,G0,De0)
        Call CalculateOmega(xl,xk,xM,Gama,p,q,TTe,omegalim,
* Rmax,Rmin,omega0,R0,PSIOC0,PSINC0)
        Call MZEROR(Va0,6)
        Call CalculateVectorA(p,p0,xM,Stress0,Va0)
        VarA0=(1.0+TTe)*p0*omega0*p*(2.0*p-p0)/(xl-xk)
        VarB0=(1.0+TTe)*p0*omega0*(2.0*p-p0)/(xl-xk)
        Call MZEROR(AUXV3,6)
        Call MatVec(De0,6,Va0,6,AUXV3)
        AUX3=DInProd(AUXV3,Va0,6)
        dlam=f0/(VarA0+AUX3)
        Call MZEROR(AUXV4,6)
        Call MatVec(De0,6,Va0,6,AUXV4)
        Call MZEROR(TTstress,6)
        Call AddVec(Stress0,AUXV4,1.0,-dlam,6,TTstress)
        TTp0=p0+dlam*VarB0
        Call CalculateStressInvariants(TTstress,TTp,TTq,
* TTeta,TTtheta)
        TTf=TTq**2.0/xM**2.0+TTp*(TTp-TTp0)
        If (abs(TTf).gt.abs(f0)) then
            AUX4=DInProd(Va0,Va0,6)
            dlam=f0/AUX4
            Call MZEROR(TTstress,6)
            Call AddVec(Stress0,Va0,1.0,-dlam,6,TTstress)
            TTp0=p0
            Call CalculateStressInvariants(TTstress,TTp,TTq,
* TTetatt,TTtheta)
            TTf=TTq**2.0/xM**2.0+TTp*(TTp-TTp0)
            end if
            if (abs(TTf).lt.ftol) exit
        end do
    end if
    iss=iss+1
    subtime2=subtime1+dsubtime
    stepsizem=0.9*((STOL/Error)**(1.0/5.0))
    if (stepsizem.gt.1.1) stepsizem=1.1
    if (isuc.eq.1) then
        if (stepsizem.gt.1.0) stepsizem=1.0
    end if
    dsubtime=stepsizem*dsubtime
    if (dsubtime.lt.0.000001) dsubtime=0.000001
    if ((subtime2+dsubtime).gt.1.0) dsubtime=1.0-subtime2
    isuc=0
    Call MZEROR(Tstress,6)
    Call COPYRVEC(TTstress,Tstress,6)
    Call MZEROR(Tstrain,6)
    Call COPYRVEC(TTstrain,Tstrain,6)
    Call MZEROR(dTstrain,6)
        do i=1,6
            dTstrain(i)=dStrain(i)*dsubtime
        end do
    Call MZEROR(TTstrain,6)
    Call AddVec(Tstrain,dTstrain,1.0,1.0,6,TTStrain)
    dEv=dTstrain(1)+dTstrain(2)+dTstrain(3)
    subtime1=subtime2
    if (subtime2.lt.1.0) goto 110
! Final stresses and hardening parameter
    Call MZEROR(Stresstt,6)
    Call COPYRVEC(TTstress,Stresstt,6)
    p0tt=TTp0
!
```

```

!Return stresses
do i=1,6
  Sig(i)=-Stresstt(i)
end do
!Update state variables
StVar(1)=StVar0(1)
StVar(2)=StVar0(2)
StVar(3)=R
StVar(4)=PSIOC
StVar(5)=PSINC
StVar(6)=omega
StVar(7)=p0t
StVar(8)=p0t*R
If (StVar0(9).lt.Stresstt(2)) then
  StVar(9)=Stresstt(2)
else
  StVar(9)=StVar0(9)
end if
StVar(10)=StVar(9)/Stresstt(2)
StVar(11)=G
StVar(12)=4.0
StVar(13)=its
StVar(14)=iss
StVar(15)=subtime2
StVar(16)=Error
End If ! IDTask = 2

If ( IDTask .Eq. 3 .Or.
*   IDTask .Eq. 6 ) Then ! Calculate D-Matrix
  xk=Props(2)
  xNu=Props(4)
  e0=Props(5)
  pt=-(Sig0(1)+Sig0(2)+Sig0(3))/3.0
  EpsV=-(dEps(7)+dEps(8)+dEps(9))
  et=(1.0+e0)*exp(-EpsV)-1.0
  Call CalculateElasticModuli(et,pt,xk,xNu,Bulk,G)
  D1=Bulk+4.0*G/3.0
  D2=Bulk-2.0*G/3.0
  D3=G
  Call MZeroR(D,36)
  D(1,1)=D1
  D(2,2)=D1
  D(3,3)=D1
  D(1,2)=D2
  D(1,3)=D2
  D(2,1)=D2
  D(2,3)=D2
  D(3,1)=D2
  D(3,2)=D2
  D(4,4)=D3
  D(5,5)=D3
  D(6,6)=D3
  BulkW = 0
  If (IsUndr.Eq.1) Then
    ! BulkW = ...
    xNu_U = 0.495d0
    Fac=(1+xNu_U)/(1-2*xNu_U) - (1+xNu)/(1-2*xNu)
    Fac=2D0*G/3D0 * Fac
    BulkW = Fac
  End If
End If ! IDTask = 3, 6

```



```
If (IDTask .Eq. 4) Then ! Number of state parameters
  nStat = nStatV
End If ! IDTask = 4

If (IDTask .Eq. 5) Then ! matrix type
  NonSym = 0 ! 1 for non-symmetric D-matrix
  iStrsDep = 1 ! 1 for stress dependent D-matrix
  iTang = 0 ! 1 for tangent D-matrix
  iTimeDep = 0 ! 1 for time dependent D-matrix
End If ! IDTask = 5

Return
End ! MyModel4
```

Прилог 3 UDSM for PLAXIS – Fortran code for HASP-E model

У овом прилогу је дат рачунарски код у Fortran-у за имплементацију HASP-E модела у PLAXIS (2D и 3D). Садржај субрутина USRMOD.for, USRLIB.for, USRADD.F.for и OKMSGIVF.for је описан у Прилогу 1. Субрутина MYMODEL1.for садржи рачунарски код за имплементацију HASP-E модела преко имплицитног алгоритма методе водећег параметра. Субрутина MYMODEL2.for садржи рачунарски код за имплементацију HASP-E модела преко експлицитног алгоритма – модификован Euler-ов поступак са аутоматском субинкрементацијом и контролом грешке. Субрутина MYMODEL3.for садржи рачунарски код за имплементацију HASP-E модела преко експлицитног алгоритма – Runge-Kutta-Dormand-Prince-ов поступак са аутоматском субинкрементацијом и контролом грешке.

Приликом учењања dll, различити алгоритми за имплементацију конститутивних релација су доступни следећим редом:

- iMod=1 : имплицитни алгоритам методе водећег параметра
- iMod=2 : експлицитни алгоритам – модификовани Euler-ов поступак са аутоматском субинкрементацијом и контролом грешке
- iMod=3 : експлицитни алгоритам - Runge-Kutta-Dormand-Prince-ов поступак са аутоматском субинкрементацијом и контролом грешке

Улазни параметри које корисник задаје су:

- Props(1) : параметар λ
- Props(2) : параметар κ
- Props(3) : параметар M_c
- Props(4) : параметар M_e
- Props(5) : Poisson-ов коефицијент μ
- Props(6) : иницијални коефицијент порозности e_0
- Props(7) : степен преконсолидације OCR
- Props(8) : параметар POP
- Props(9) : коефицијент притиска тла у стању мировања за нормално консолидовано тло K_0^{NC}

За методу водећег параметра

- Props(10) : референтни модул смицања при веома малим деформацијама G_0^{ref}
- Props(11) : смичућа деформација при којој је секантни модул смицања редукован на 70% иницијалне вредности

За експлицитне алгоритме

- Props(10) : толеранција за локалну грешку STOL (само за експлицитне алгоритме)
- Props(11) : референтни модул смицања при веома малим деформацијама G_0^{ref}

- Props(12) : смичућа деформација при којој је секантни модул смицања редукован на 70% иницијалне вредности

Као параметри стања, за сваку напонску тачку, чувају се следеће величине:

- StVar(1) : индикатор да ли је потребно поново срачунати параметре који описују иницијално стање (параметар Γ – израз (3-48)) и ресетовати претходну историју оптерећењивања (вратити све цигле у координатни почетак у простору деформација) на почетку нове фазе. Има вредност 0 ако се поново рачуна Γ и ресетује претходна историја оптерећења (увек у иницијалној фази када се генерише почетно стање напона и када је активна наредба Reset State Variables) или вредност 1 када не треба да се поново рачуна Γ и ресетује претходна историја оптерећења
- StVar(2) : параметар Γ
- StVar(3) : изотропни степен преконсолидације R
- StVar(4) : параметар стања за тренутну напонску тачку Ψ
- StVar(5) : параметар стања за имагинарну напонску тачку $\bar{\Psi}$
- StVar(6) : коефицијент ојачања ω
- StVar(7) : параметар ојачања за тренутну површ течења p'_0
- StVar(8) : параметар ојачања за граничну површ \bar{p}'_0
- StVar(9) : максимални претходни вертикални ефективни напон
- StVar(10) : тренутни степен преконсолидације, рачуна се као количник тренутног вертикалног ефективног напона и максималног претходног вертикалног ефективног напона
- StVar(11) : контролни индикатор, указује који алгоритам је коришћен и има исту вредност као iMod
- StVar(12) : референтни тангентни модул смицања G_t^{ref}
- StVar(13) : референтни модул смицања при великим деформацијама G_{ur}^{ref}
- StVar(14) : тангентни модул смицања G
- StVar(15) : проценат деградације смичуће крутости који одговара једној цигли $\Delta\omega_G^b$
- StVar(16) : тренутна вредност Lode-овог угла
- StVar(17) : константа X која описује облик површи течења у девијаторској равни
- StVar(18) : константа Y која описује облик површи течења у девијаторској равни
- StVar(19) : тренутна вредност параметра M
- StVar(20) : локална грешка након последњег субинкремента (само за експлицитне алгоритме)
- StVar(21) – StVar(40) : дужине канапа за сваку циглу
- StVar(41) – StVar(160) : координате цигли у простору деформација
- StVar(161) – StVar(180) : индикатор да ли је канап једне цигле затегнут (= 1) или не (= 0)

USRMOD.for

```

Subroutine User_Mod ( IDTask, iMod, IsUndr,
*                          iStep, iTer, iEl, Int,
*                          X, Y, Z,
*                          Time0, dTime,
*                          Props, Sig0, Swp0, StVar0,
*                          dEps, D, BulkW,
*                          Sig, Swp, StVar, ipl,
*                          nStat, NonSym, iStrsDep, iTimeDep, iTang,
*                          iPrjDir, iPrjLen, iAbort )
!
! Purpose: User supplied soil model, HASP-E model
! Types : iMod=1 : implicit gpm with mean effective stress as governing
parameter
!            iMod=2 : explicit modified Euler with automatic subincrementation
and error control
!            iMod=3 : explicit fifth-order accurate (Runge-Kutta-Dormand-
Prince) with automatic subincrementation and error control
!
! Depending on IDTask, 1 : Initialize state variables
!                     2 : calculate stresses,
!                     3 : calculate material stiffness matrix
!                     4 : return number of state variables
!                     5 : inquire matrix properties
!                       return switch for non-symmetric D-matrix
!                       stress/time dependent matrix
!                     6 : calculate elastic material stiffness matrix
! Arguments:
! I/O Type
! IDTask I I : see above
! iMod I I : model number (1..10)
! IsUndr I I : =1 for undrained, 0 otherwise
! iStep I I : Global step number
! iter I I : Global iteration number
! iel I I : Global element number
! Int I I : Global integration point number
! X I R : X-Position of integration point
! Y I R : Y-Position of integration point
! Z I R : Z-Position of integration point
! Time0 I R : Time at start of step
! dTime I R : Time increment
! Props I R( ) : List with model parameters
! Sig0 I R( ) : Stresses at start of step
! Swp0 I R : Excess pore pressure start of step
! StVar0 I R( ) : State variable at start of step
! dEps I R( ) : Strain increment
! D I/O R(,) : Material stiffness matrix
! BulkW I/O R : Bulkmodulus for water (undrained only)
! Sig 0 R( ) : Resulting stresses
! Swp 0 R : Resulting excess pore pressure
! StVar 0 R( ) : Resulting values state variables
! ipl 0 I : Plasticity indicator
! nStat 0 I : Number of state variables
! NonSym 0 I : Non-Symmetric D-matrix ?
! iStrsDep 0 I : =1 for stress dependent D-matrix
! iTimeDep 0 I : =1 for time dependent D-matrix
! iTang 0 I : =1 for tangent matrix
! iAbort 0 I : =1 to force stopping of calculation
!
Implicit Double Precision (A-H, 0-Z)

```

```

!
  Dimension Props(*), Sig0(*), StVar0(*), dEps(*), D(6,6),
  *      Sig(*), StVar(*), iPrjDir(*)

  Data iounit / 0 /
  Save iounit

!
!--- Local variables
!
  Character*100 BaseName

!DEC$ ATTRIBUTES DLLEXPORT, STDCALL, REFERENCE :: User_Mod

  BaseName = 'HASP-E'
  ! Possibly open a file for debugging purposes
  If (iounit.Eq.0) Then

    Call Open_Dbg_File( iPrjDir, iPrjLen, BaseName )

    Write(1,*)'File 1 opened: ', Trim(BaseName)
    ! maybe write some more info on version to debug file ?
    write(1,*)'Compiled : ',__DATE__, ' ', __TIME__
!DEC$ IF DEFINED(_X86_)
  ! this 32-bit ??
  Write(1,1050)'IF32', __INTEL_COMPILER
!DEC$ ELSE
  ! this 64-bit ??
  Write(1,1050)'IF64', __INTEL_COMPILER
!DEC$ ENDIF
  1050 format ( 1X,A,1x,I0 )
  iounit = 1
  Call WriVec(1, 'Props', Props, 50)
  Call Flush(1)
  End If

  Call WriIvl( -1, 'iounit', iounit )
  Call WriIvl( -1, 'IDTask', IDTask )
  Select Case (iMod)
Case (1) ! gpm integration with mean effective stress as governing parameter
  Call MyModel1( IDTask, iMod, IsUndr, iStep, iTer, iEl, Int,
  *      X, Y, Z, Time0, dTime,
  *      Props, Sig0, Swp0, StVar0,
  *      dEps, D, BulkW, Sig, Swp, StVar, ipl,
  *      nStat, NonSym, iStrsDep, iTimeDep, iTang,
  *      iAbort )
  Case (2) ! explicit modified Euler with automatic subincrementation and
error control
  Call MyModel2( IDTask, iMod, IsUndr, iStep, iTer, iEl, Int,
  *      X, Y, Z, Time0, dTime,
  *      Props, Sig0, Swp0, StVar0,
  *      dEps, D, BulkW, Sig, Swp, StVar, ipl,
  *      nStat, NonSym, iStrsDep, iTimeDep, iTang,
  *      iAbort )
  Case (3) ! explicit fifth-order accurate (Runge-Kutta-Dormand-Prince) with
automatic subincrementation and error control
  Call MyModel3( IDTask, iMod, IsUndr, iStep, iTer, iEl, Int,
  *      X, Y, Z, Time0, dTime,
  *      Props, Sig0, Swp0, StVar0,
  *      dEps, D, BulkW, Sig, Swp, StVar, ipl,
  *      nStat, NonSym, iStrsDep, iTimeDep, iTang,
  *      iAbort )
  Case Default

```

```

        Write(1,*) 'invalid model number in UsrMod', iMod
        Write(1,*) 'IDTask: ',IDTask
        Stop 'invalid model number in UsrMod'
        iAbort=1
        Return
    End Select ! iMod
    If (IDTask .Eq. 5.And.iel+int.Eq.2) Then
        Write(1,*)'nStat : ',nStat
        Write(1,*)'NonSym : ',NonSym
        Write(1,*)'StrsDep : ',iStrsDep
        Write(1,*)'TimeDep : ',iTimeDep
        Write(1,*)'Tangent : ',iTang
    End If
    If (IDTask == -333 .And. iel+int == -1234) Then
!       Write(1,*)'IDTask: ',IDTask,' iStep,iTer',iStep,iTer
!       Call Flush(1)
    End If
    Call WriIvl( -1, 'IDTask end',IDTask )
    Return
End ! User_Mod

! *****

Subroutine Open_Dbg_File( iPrjDir, iPrjLen, BaseName )
Implicit None

Integer, intent(in) :: iPrjLen, iPrjDir(*)
Character*(*), intent(in):: BaseName

Character*255 PrjDir, Dbg_Name

Integer i, nErr, ios

PrjDir=' '
Do i=1,iPrjLen
    PrjDir(i:i) = Char( iPrjDir(i) )
End Do

Dbg_Name=PrjDir(:iPrjLen)//'data.'//trim(BaseName)//'.rr0'
nErr=0
1 Continue
    Open( Unit= 1, File= Dbg_Name,iostat=ios)
    If (ios.Eq.0) Close(Unit=1,Status='delete',iostat=ios)

    If (ios.Ne.0) Then
        !
        ! in case of error try ...udsmex1 or uds mex2 or ..
        !
        nErr=nErr+1
        Dbg_Name=PrjDir(:iPrjLen)//'data.'//
*       trim(BaseName)//char(48+nErr)//'.rr0'

        If (nErr.Lt.10) Goto 1
    End If

    Open( Unit= 1, File= Dbg_Name,blocksize=4096)

End Subroutine Open_Dbg_File

! *****

!
```

```

! Interfaces to routines to report parameter names, counts etc.
!

Subroutine Add_Str_Length( aString )
Implicit None
Character*(*) aString
Character *255 tString
Integer Lt          ! length of incoming string

! routine should add the length of the string as the first character

tString = aString
Lt      = Len_Trim(tString)
aString = Char(Lt) // tString(1:Lt)

End Subroutine Add_Str_Length

Subroutine GetModelCount( nMod )
!
! Return the maximum model number (nMod) in this DLL
!
Implicit None
Integer (Kind=4) nMod

!DEC$ ATTRIBUTES DLLEExport, StdCall, reference :: GetModelCount

Call Get_Model_Count( nMod )

Return
End ! GetModelCount

Subroutine GetModelName( iMod , ModelName )
!
! Return the name of the different models
!
Implicit None
Integer iMod
Character (Len= * ) ModelName
Character (Len=255) tName
!DEC$ ATTRIBUTES DLLEExport, StdCall, reference :: GetModelName

Call Get_Model_Name( iMod , ModelName )
Call Add_Str_Length( ModelName )

End ! GetModelName

Subroutine GetParamCount( iMod , nParam )
!
! Return the number of parameters of the different models
!
Implicit None
Integer iMod, nParam

!DEC$ ATTRIBUTES DLLEExport, StdCall, reference :: GetParamCount

Call Get_Param_Count( iMod , nParam )

End ! GetParamCount

Subroutine GetParamName( iMod , iParam, ParamName )
!
! Return the parameters name of the different models

```

```

!
Implicit None
Integer iMod, iParam
Character (Len=255) ParamName, Units

!DEC$ ATTRIBUTES DLLEExport, StdCall, reference :: GetParamName

Call GetParamAndUnit(iMod,iParam,ParamName,Units)
Call Add_Str_Length( ParamName )

End ! GetParamName

Subroutine GetParamUnit( iMod , iParam, Units )
!
! Return the units of the different parameters of the different models
!
Implicit None
Integer iMod, iParam
Character (Len=255) ParamName, Units

!DEC$ ATTRIBUTES DLLEExport, StdCall, reference :: GetParamUnit

Call GetParamAndUnit(iMod,iParam,ParamName,Units)
Call Add_Str_Length( Units )

End ! GetParamUnit

Subroutine GetStateVarCount( iMod , nVar )
!
! Return the number of state variables of the different models
!
Implicit None
Integer iMod, nVar

!DEC$ ATTRIBUTES DLLEExport, StdCall, reference :: GetStateVarCount

Call Get_StateVar_Count( iMod , nVar )

End ! GetStateVarCount

Subroutine GetStateVarName( iMod , iVar, Name )
!
! Return the name of the different state variables
! of the different models
!
Implicit None
Integer iMod, iVar
Character (Len=255) Name, Unit

!DEC$ ATTRIBUTES DLLEExport, StdCall, reference :: GetStateVarName

Call GetStateVarNameAndUnit( iMod , iVar, Name, Unit )
Call Add_Str_Length( Name )

End ! GetStateVarName

Subroutine GetStateVarUnit( iMod , iVar, Unit )
!
! Return the units of the different state variables of the different models
!
Implicit None
Integer iMod, iVar

```



```
Character (Len=255) Name, Unit
!DEC$ ATTRIBUTES DLLEExport, StdCall, reference :: GetStateVarUnit
Call GetStateVarNameAndUnit( iMod , iVar, Name, Unit )
Call Add_Str_Length( Unit )
End ! GetStateVarUnit
```

USRLIB.for

```
Subroutine MZEROR(R,K)
C
C*****
C
C   Function: To make a real array R with dimension K to zero
C
C*****
C
C   Implicit Double Precision (A-H,O-Z)
C   Dimension R(*)

C   Do J=1,K
C     R(J) = 0.0D0
C   End Do

C   Return
C   End
Subroutine COPYRVEC(R1,R2,K)
C
C*****
C
C   Function: To copy a Double array R1 with Dimension K to R2
C
C*****
C
C   Implicit Double Precision (A-H,O-Z)
C   Dimension R1(*),R2(*)

C   Do J=1,K
C     R2(J)=R1(J)
C   End Do

C   Return
C   End
C*****
Subroutine MatVec(xMat,IM,Vec,N,VecR)
C*****
C
C   Calculate VecR = xMat*Vec
C
C I  xMat  : (Square) Matrix (IM,*)
C I  Vec   : Vector
C I  N     : Number of rows/columns
C O  VecR  : Resulting vector
C
C*****
C   Implicit Double Precision (A-H,O-Z)
C   Dimension xMat(IM,*),Vec(*),VecR(*)
C*****
C   Do I=1,N
C     X=0
C     Do J=1,N
C       X=X+xMat(I,J)*Vec(J)
C     End Do
C     VecR(I)=X
C   End Do
C   Return
C   End    ! Subroutine MatVec
```

```

C*****
Subroutine AddVec(Vec1,Vec2,R1,R2,N,VecR)
C*****
C
C   Calculate VecR() = R1*Vec1()+R2*Vec2()
C
C I  Vec1,
C I  Vec2  : Vectors
C I  R1,R2 : Multipliers
C I  N     : Number of rows
C O  VecR  : Resulting vector
C
C*****
Implicit Double Precision (A-H,O-Z)
Dimension Vec1(*),Vec2(*),VecR(*)
C*****
Do I=1,N
  X=R1*Vec1(I)+R2*Vec2(I)
  VecR(I)=X
End Do
Return
End      ! Subroutine AddVec
C
C*****
Double Precision Function DInProd(A,B,N)
C*****
C
C   Returns the Inproduct of two vectors
C
C I  A,B  : Two vectors
C I  N    : Used length of vectors
C*****
Implicit Double Precision (A-H,O-Z)
Dimension A(*),B(*)
C*****

X = 0
Do I=1,N
  X = X + A(I)*B(I)
End Do
DInProd = X
Return
End      ! Function DInProd
C*****
subroutine CalculateStressInvariants(Stress,p,q,eta,theta)
C*****
! Stress(6)      : Array of stresses
! p              : Mean effective stress
! q              : Deviatoric stress
! eta            : Stress ratio
! theta          : Lodes angle
!
! ---Local variables---
!
! D3             : Third invariant of deviator stress
! Pie            : Value of Pi
!
Implicit Double Precision (A-H,O-Z)
Dimension Stress(6)
Parameter(
*Pie=3.14159 26535 89793 23846 26433 83279 50288 41971 69399d0)
p=(Stress(1)+Stress(2)+Stress(3))/3.0

```

```

if (p.le.0.0) p=0.001
q=sqrt(3.0*(1.0/6.0*((Stress(1)-Stress(2))**2.0+(Stress(2)-
*Stress(3))**2.0+(Stress(3)-Stress(1))**2.0)+Stress(4)**2.0+
*Stress(5)**2.0+Stress(6)**2.0))
eta=q/p
D3=(Stress(1)-p)*(Stress(2)-p)*(Stress(3)-p)+2.0*Stress(4)*
*Stress(5)*Stress(6)-(Stress(1)-p)*Stress(5)**2.0-(Stress(2)-
*p)*Stress(6)**2.0-(Stress(3)-p)*Stress(4)**2.0
theta=-1.0/3.0*asin(27/2*D3/q**3.0)
if (isnan(theta)) theta=-Pie/6.0
if (theta.gt.Pie/6.0) theta=Pie/6.0
if (theta.lt.-Pie/6.0) theta=-Pie/6.0
return
end ! CalculateStressInvariants
C*****
subroutine CalculateStressDeviator(Stress,p,Stressd)
C*****
! Stress(6)      : Array of stresses
! p              : Mean effective stress
! Stressd(6)     : Array of deviatoric stress
!
Implicit Double Precision (A-H,0-Z)
Dimension Stress(6), Stressd(6)
do i=1,3
    Stressd(i)=Stress(i)-p
    Stressd(i+3)=Stress(i+3)
end do
return
end ! CalculateStressDeviator
C*****
subroutine CalculateVoidRatio(Strain,e0,e)
C*****
! Strain(6)      : Array of strains
! e0             : Initial void ratio
! e              : Current void ratio
!
! ---Local variables---
!
! Epsv           : Current volumetric strain
!
Implicit Double Precision (A-H,0-Z)
Dimension Strain(6)
Epsv=Strain(1)+Strain(2)+Strain(3)
e=(1+e0)*exp(-Epsv)-1
if (e.lt.0.0) e=0.0
if (e.gt.1000) e=1000
return
end ! CalculateVoidRatio
C*****
subroutine CalculateStrainInvariants(Strain,EpsV,Epsq,ShearStrain)
C*****
! Strain(6)      : Array of strains
! EpsV           : Volumetric strain
! Epsq           : Deviatoric strain
! ShearStrain    : Shear strain invariant
!
Implicit Double Precision (A-H,0-Z)
Dimension Strain(6)
EpsV=Strain(1)+Strain(2)+Strain(3)
Epsq=sqrt(2.0/3.0*((Strain(1)-EpsV/3.0)**2.0+(Strain(2)-EpsV/3.0)
**2.0+(Strain(3)-EpsV/3.0)**2.0+1.0/2.0*(Strain(4)**2.0+
*Strain(5)**2.0+Strain(6)**2.0))

```

```

ShearStrain=3.0/2.0*Epsq
return
end ! CalculateStrainInvariants
C*****
subroutine CalculateStrainDeviator(Strain,EpsV,Straind)
C*****
! Strain(6)      : Array of strains
! EpsV          : Volumetric strain
! Straind(6)    : Array of deviatoric strain
!
Implicit Double Precision (A-H,O-Z)
Dimension Strain(6), Straind(6)
do i=1,3
  Straind(i)=Strain(i)-EpsV/3.0
  Straind(i+3)=Strain(i+3)/2.0
end do
return
end ! CalculateStrainDeviator
C*****
subroutine CalculateElasticModuli(e,p,xk,xNu,Bulk,G)
C*****
! e              : Void ratio
! p              : Mean effective stress
! xk             : Kappa, slope of URL
! xNu           : Poissons ratio
! Bulk          : Bulk modulus
! G             : Shear modulus
!
Implicit Double Precision (A-H,O-Z)
Bulk=(1.0+e)*p/xk
If (Bulk.lt.1.0) Bulk=1.0
G=3.0*(1.0-2.0*xNu)*Bulk/(2.0*(1.0+xNu))
return
end ! CalculateStrainDeviator
C*****
subroutine CalculateOmega(xl,xk,xM,Gama,p,q,e,omegalim,Rmax,Rmin,
*omega,R,PSIOC,PSINC)
C*****
! xl            : Lambda, slope oc ICL
! xk            : Kappa, slope of URL
! xM            : M, slope of CSL
! Gama          : Specific volume at CSL for p'=1 kPa
! p             : Mean effective stress
! q             : Deviator stress
! e             : Void ratio
! omegalim     : Limiting value of omega in order to avoid numerical
instability
! Rmax         : Max value of R in order to avoid numerical instability
! Rmin         : Min value of R in order to avoid non-possible values
! omega        : Hardening coefficient
! R            : R, isotropic OCR
! PSIOC        : State parameter for OC
! PSINC        : State parameter for NC
!
Implicit Double Precision (A-H,O-Z)
eta=q/p
PSIOC=1.0+e+xl*log(p)-Gama
PSINC=(xl-xk)*log(2.0*xM**2.0/(xM**2.0+eta**2.0))
R=exp((PSINC-PSIOC)/(xl-xk))
If (R.gt.Rmax) then
  PSIOC=PSINC-(xl-xk)*log(Rmax)
  R=Rmax

```

```

end if
if (R.lt.Rmin) then
  PSIOC=PSINC-(xl-xk)*log(Rmin)
  R=Rmin
end if
omega=(1.0+(PSINC-PSIOC)/PSINC)*R
If (omega.gt.omegalim) omega=omegalim
If (omega.lt.-omegalim) omega=-omegalim
if (omega.gt.0.0.and.omega.lt.0.0001) omega=0.0001
if (omega.gt.-0.0001.and.omega.lt.0.0) omega=-0.0001
return
end ! CalculateOmega
C*****
subroutine CalculateGMPYield(xl,xk,xM,et,ett,G,pt,qt,p0t,dEpsq,
dEpsd,Sdt,CStol,omega,ptt,f)
C*****
! xl      : Slope of ICL
! xk      : Slope of URL
! xM      : Slope of CSL
! et      : Void ratio at start of step
! ett     : Void ratio at end of step
! G       : Shear modulus
! pt      : Mean effective stress at start of step
! qt      : Shear stress at start of step
! p0t     : Hardening parameter at start of step
! dEpsq   : Increment of deviatoric strain
! dEpsd(6) : Array of deviatoric strain increments
! Sdt(6)  : Array of deviatoric stresses at the end of step
! CStol   : Tolerance for assessing whether the critical state is reached
! omega   : Hardening coefficient
! ptt     : Mean effective stress at the end of step
! f       : Value of yield function for governing parameter ptt
!
! ---Local variables---
!
! Sdtt(6) : Array of deviatoric stresses at the end of increment
! Stt(6)  : Array of stresses at the end of step
! p0tt    : Hardening parameter at the end of step
! qtt     : Shear stress at the end of step
! dq      : increment of shear stress
! dEpsqp : Increment of plastic shear strain
! dlam    : Plastic multiplier
! dEpsvp  : Increment of plastic volumetric strain
!
Implicit Double Precision (A-H,O-Z)
Dimension dEpsd(6), Sdt(6), Sdtt(6), Stt(6)
p0tt=p0t*exp((et-ett)*omega/(xl-xk))*(pt/ptt)**(xk*omega/(xl-xk))
if (p0tt.gt.huge(1.0d0)) p0tt=2*pt ! check if p0tt is bigger than max possible
real number
if (isnan(p0tt)) p0tt=p0t ! check if p0tt is a number
if (p0tt.lt.tiny(1.0d0)) p0tt=0.001 ! check if p0tt is not too small
if (abs(2*ptt-p0tt).lt.CStol) then ! check if critical state is reached
  qtt=xM*ptt
  dq=qtt-qt
  dEpsqp=(dEpsq-dq/3.0/G)/omega
  dlam=xM*dEpsqp/2.0/ptt
else
  dEpsvp=(xl-xk)/(1.0+et)*log(p0tt/p0t)/omega
  dlam=dEpsvp/(2*ptt-p0tt)
end if
do i=1,6
  Sdtt(i)=(dEpsd(i)+Sdt(i)/2.0/G)/(3.0*dlam/xM**2.0+1.0/2.0/G)

```

```

end do
do i=1,3
  Stt(i)=Sdtt(i)+ptt
  Stt(i+3)=Sdtt(i+3)
end do
qtt=sqrt(3.0*(1.0/6.0*((Stt(1)-Stt(2))**2.0+(Stt(2)-Stt(3))**2.0+
*(Stt(3)-Stt(1))**2.0)+Stt(4)**2.0+Stt(5)**2.0+Stt(6)**2.0))
f=qtt**2.0/xM**2.0+ptt*(ptt-p0tt)
return
end ! CalculateGMPYield
C*****
subroutine CalculateGMPStress(xl,xk,xM,et,ett,G,pt,qt,p0t,dEpsq,
*dEpsd,Sdt,CStol,omega,ptt,dlam,p0tt,Stt)
C*****
! xl      : Slope of ICL
! xk      : Slope of URL
! xM      : Slope of CSL
! et      : Void ratio at start of step
! ett     : Void ratio at end of step
! G       : Shear modulus
! pt      : Mean effective stress at start of step
! qt      : Shear stress at start of step
! p0t     : Hardening parameter at start of step
! dEpsq   : Increment of deviatoric strain
! dEpsd(6) : Array of deviatoric strain increments
! Sdt(6)  : Array of deviatoric stresses at the end of step
! CStol   : Tolerance for assessing whether the critical state is reached
! omega   : Hardening coefficient
! ptt     : Mean effective stress at the end of step
! dlam    : Plastic multiplier
! p0tt    : Hardening parameter at the end of step
! Stt(6)  : Array of stresses at the end of step
!
! ---Local variables---
!
! Sdtt(6)  : Array of deviatoric stresses at the end of increment
! qtt      : Shear stress at the end of step
! dq       : increment of shear stress
! dEpsqp   : Increment of plastic shear strain
! dEpsvp   : Increment of plastic volumetric strain
! f        : Value of yield function for governing parameter ptt
!
Implicit Double Precision (A-H,O-Z)
Dimension dEpsd(6), Sdt(6), Sdtt(6), Stt(6)
p0tt=p0t*exp((et-ett)*omega/(xl-xk))*(pt/ptt)**(xk*omega/(xl-xk))
if (p0tt.gt.huge(1.0d0)) p0tt=2*pt ! check if p0tt is bigger then max possible
real number
if (isnan(p0tt)) p0tt=p0t ! check if p0tt is a number
if (p0tt.lt.tiny(1.0d0)) p0tt=0.001 ! check if p0tt is not too small
if (abs(2*ptt-p0tt).lt.CStol) then ! check if critical state is reached
  qtt=xM*ptt
  dq=qtt-qt
  dEpsqp=(dEpsq-dq/3.0/G)/omega
  dlam=xM*dEpsqp/2.0/ptt
  dEpsvp=0.0
else
  dEpsvp=(xl-xk)/(1+et)*log(p0tt/p0t)/omega
  dlam=dEpsvp/(2*ptt-p0tt)
end if
do i=1,6
  Sdtt(i)=(dEpsd(i)+Sdt(i)/2.0/G)/(3.0*dlam/xM**2.0+1.0/2.0/G)
end do

```

```

do i=1,3
  Stt(i)=Sdtt(i)+ptt
  Stt(i+3)=Sdtt(i+3)
end do
qtt=sqrt(3.0*(1.0/6.0*((Stt(1)-Stt(2))**2.0+(Stt(2)-Stt(3))**2.0+
*(Stt(3)-Stt(1))**2.0)+Stt(4)**2.0+Stt(5)**2.0+Stt(6)**2.0))
f=qtt**2.0/xM**2.0+ptt*(ptt-p0tt)
eta=qtt/ptt
return
end ! CalculateGMPPStress
C*****
subroutine CalculateVectorA(p,p0,xM,Stress,Va)
C*****
! p          : Mean effective stress
! p0         : Hardening parameter
! xM         : Slope of CSL
! Stress(6)  : Array of stresses
! Va(6)      : Array of yield function derivatives
!
Implicit Double Precision (A-H,0-Z)
Dimension Stress(6), Va(6)
Va(1)=(2.0*p-p0)/3.0+3.0*(Stress(1)-p)/xM**2.0
Va(2)=(2.0*p-p0)/3.0+3.0*(Stress(2)-p)/xM**2.0
Va(3)=(2.0*p-p0)/3.0+3.0*(Stress(3)-p)/xM**2.0
Va(4)=6.0*Stress(4)/xM**2.0
Va(5)=6.0*Stress(5)/xM**2.0
Va(6)=6.0*Stress(6)/xM**2.0
return
end ! CalculateVectorA
C*****
subroutine CalculateDeMatrix(Bulk,G,De)
C*****
! K          : Bulk modulus
! G          : Shear modulus
! De(6)      : Elastic De matrix
! Stress(6)  : Array of stresses
! Va(6)      : Array of yield function derivatives
!
! ---Local variables---
!
! D1,D2,D3   : Auxilliary variables for filling of De matrix
!
Implicit Double Precision (A-H,0-Z)
Dimension De(6,6)
D1=Bulk+4.0*G/3.0
D2=Bulk-2.0*G/3.0
D3=G
Call MZeroR(De,36)
De(1,1)=D1
De(2,2)=D1
De(3,3)=D1
De(1,2)=D2
De(1,3)=D2
De(2,1)=D2
De(2,3)=D2
De(3,1)=D2
De(3,2)=D2
De(4,4)=D3
De(5,5)=D3
De(6,6)=D3
return
end ! CalculateDeMatrix

```



```

C*****
  subroutine CalculateElasticModuliBrick(BT,Strain,SH,SLen,G1,
    *G2,Gur,G0,G,e,p,xNu,xka,dBG,Bulk)
C*****
  ! BT(20)      : Array which contains whether the string is slack or taut
  ! Strain(6)   : Strain at the end of step/increment/subincrement
  ! SH(20,6)   : Matrix with positions of bricks
  ! SLen(20)   : Array with string lengths
  ! G1         : Shear modulus at the start of step for pref=100 kPa
  ! G2         : Shear modulus at the end of step for pref=100 kPa
  ! Gur        : Shear modulus for URL at pref=100 kPa
  ! G0         : Small strain shear modulus for pref=100 kPa
  ! G          : Shear modulus for stress calculation
  ! e          : Void ratio
  ! p          : Mean effective stress
  ! xNu        : Poisson's ratio
  ! xka        : Small strain kappa
  ! dBG        : Proportion of shear stiffness belonging to one brick
  ! Bulk       : Bulk modulus for stress calculation
  !
  ! ---Local variables---
  !
  ! dG         : Total reduction of small strain shear stiffness
  ! SD(6)      : Strain distance
  ! SDVol      : Volumetric strain for strain distance
  ! SDq        : Deviatoric strain for strain distance
  ! strainq    : Shear strain for strain distance
  ! rd         : relative strain distance
  ! Gt         : Shear modulus for stress calculation at pref=100 kPa
  !
  Implicit Double Precision (A-H,O-Z)
  Dimension BT(20), Strain(6), SH(20,6), SHr(20,6), SLen(20), SD(6)
  dG=0.0
  do i=1,20
    BT(i)=0.0
    do j=1,6
      SD(j)=Strain(j)-SH(i,j)
    end do
    SDvol=0.0
    SDvol=SD(1)+SD(2)+SD(3)
    SDq=sqrt(2.0/3.0*((SD(1)-SDvol/3.0)**2.0+(SD(2)-SDvol/
    *3.0)**2.0+(SD(3)-SDvol/3.0)**2.0+1.0/2.0*(SD(4)**2.0+SD(5)**2.0+
    *SD(6)**2.0)))
    strainq=3.0/2.0*SDq
    if (strainq.gt.SLen(i)) then
      ! string i is taut
      BT(i)=1.0
      rd=(strainq-SLen(i))/strainq
      dG=dG+dBG
      do j=1,6
        SH(i,j)=SH(i,j)+rd*(Strain(j)-SH(i,j))
      end do
    end if
  end do
  G2=G0*(1.0-dG)
  if (G1.gt.G2) then
    Gt=G1
  else
    Gt=G2
  end if
  if (Gt.gt.G0) Gt=G0
  if (Gt.lt.Gur) Gt=Gur

```

```

! Elastic moduli for calculation
G=Gt*p/100.0
if (G.lt.1.0) G=1.0
xka=(1+e)*p*1.0/G*3.0*(1.0-2.0*xNu)/(2.0*(1.0+xNu))
Bulk=2.0*G*(1.0+xNu)/(3.0*(1.0-2.0*xNu))
return
end ! CalculateElasticModuliBrick
C*****
subroutine CalculateGMPYield2(xl,xk,xM,et,ett,G,pt,qt,p0t,dEpsq,
dEpsd,Sdt,CStol,omega,ptt,f,xX,yY,zZ)
C*****
! xl          : Slope of ICL
! xk          : Slope of URL
! xM          : Slope of CSL
! et          : Void ratio at start of step
! ett         : Void ratio at end of step
! G           : Shear modulus
! pt          : Mean effective stress at start of step
! qt          : Shear stress at start of step
! p0t         : Hardening parametar at start of step
! dEpsq       : Increment of deviatoric strain
! dEpsd(6)    : Array of deviatoric strain increments
! Sdt(6)      : Array of deviatoric stresses at the end of step
! CStol       : Tolerance for assessing whether the critical state is reached
! omega       : Hardening coefficient
! ptt         : Mean effective stress at the end of step
! f           : Value of yield function for governing parameter ptt
! xX,yY,zZ    : Parameters that describe the dependency of M on Lode's angle
!
! ---Local variables---
!
! Sdtt(6)     : Array of deviatoric stresses at the end of increment
! Stt(6)      : Array of stresses at the end of step
! p0tt        : Hardening parameter at the end of step
! qtt         : Shear stress at the end of step
! dq          : increment of shear stress
! dEpsqp      : Increpent of plastic shear strain
! dlam        : Plastic multiplier
! dEpsvp      : Incrrement of plastic volumetric strain
! D3          : Third invariant of deviator stress
! Pie         : Value of Pi
! theta       : Lodes angle
! xMtt        : Value of M at the end of step
!
Implicit Double Precision (A-H,O-Z)
Dimension dEpsd(6), Sdt(6), Sdtt(6), Stt(6)
Parameter(
*Pie=3.14159 26535 89793 23846 26433 83279 50288 41971 69399d0)
p0tt=p0t*exp((et-ett)*omega/(xl-xk))*(pt/ptt)**(xk*omega/(xl-xk))
if (p0tt.gt.huge(1.0d0)) p0tt=2*pt ! check if p0tt is bigger then max possible
real number
if (isnan(p0tt)) p0tt=p0t ! check if p0tt is a number
if (p0tt.lt.tiny(1.0d0)) p0tt=0.001 ! check if p0tt is not too small
if (abs(2*ptt-p0tt).lt.CStol) then ! check if critical state is reached
qtt=xM*ptt
dq=qtt-qt
dEpsqp=(dEpsq-dq/3.0/G)/omega
dlam=xM*dEpsqp/2.0/ptt
else
dEpsvp=(xl-xk)/(1.0+et)*log(p0tt/p0t)/omega
dlam=dEpsvp/(2*ptt-p0tt)
end if

```

```

do i=1,6
  Sdtt(i)=(dEpsd(i)+Sdt(i)/2.0/G)/(3.0*dLam/xM**2.0+1.0/2.0/G)
end do
do i=1,3
  Stt(i)=Sdtt(i)+ptt
  Stt(i+3)=Sdtt(i+3)
end do
qtt=sqrt(3.0*(1.0/6.0*((Stt(1)-Stt(2))**2.0+(Stt(2)-Stt(3))**2.0+
*(Stt(3)-Stt(1))**2.0)+Stt(4)**2.0+Stt(5)**2.0+Stt(6)**2.0))
D3=(Stt(1)-ptt)*(Stt(2)-ptt)*(Stt(3)-ptt)+2.0*Stt(4)*
*Stt(5)*Stt(6)-(Stt(1)-ptt)*Stt(5)**2.0-(Stt(2)-
*ptt)*Stt(6)**2.0-(Stt(3)-ptt)*Stt(4)**2.0
theta=-1.0/3.0*asin(27.0/2.0*D3/qtt**3.0)
if (isnan(theta)) theta=-Pie/6.0
if (theta.gt.Pie/6.0) theta=Pie/6.0
if (theta.lt.-Pie/6.0) theta=-Pie/6.0
xMtt=xX*(1.0+yY*sin(3.0*theta))**zZ
if (isnan(xMtt)) xMtt=xX*(1.0-yY)**zZ
f=qtt**2.0/xMtt**2.0+ptt*(ptt-p0tt)
return
end ! CalculateGPMPYield2
C*****
subroutine CalculateGPMPStress2(xl,xk,xM,et,ett,G,pt,qt,p0t,dEpsq,
*dEpsd,Sdt,CStol,omega,ptt,dLam,p0tt,Stt,xX,yY,zZ)
C*****
! xl          : Slope of ICL
! xk          : Slope of URL
! xM          : Slope of CSL
! et          : Void ratio at start of step
! ett         : Void ratio at end of step
! G           : Shear modulus
! pt          : Mean effective stress at start of step
! qt          : Shear stress at start of step
! p0t         : Hardening parameter at start of step
! dEpsq       : Increment of deviatoric strain
! dEpsd(6)    : Array of deviatoric strain increments
! Sdt(6)      : Array of deviatoric stresses at the end of step
! CStol       : Tolerance for assessing whether the critical state is reached
! omega       : Hardening coefficient
! ptt         : Mean effective stress at the end of step
! dLam        : Plastic multiplier
! p0tt        : Hardening parameter at the end of step
! Stt(6)      : Array of stresses at the end of step
! xX,yY,zZ    : Parameters that describe the dependency of M on Lode's angle
!
! ---Local variables---
!
! Sdtt(6)     : Array of deviatoric stresses at the end of increment
! qtt         : Shear stress at the end of step
! dq          : increment of shear stress
! dEpsqp      : Increment of plastic shear strain
! dEpsvp      : Increment of plastic volumetric strain
! f           : Value of yield function for governing parameter ptt
! D3          : Third invariant of deviator stress
! Pie         : Value of Pi
! theta       : Lodes angle
! xMtt        : Value of M at the end of step
!
Implicit Double Precision (A-H,O-Z)
Dimension dEpsd(6), Sdt(6), Sdtt(6), Stt(6)
Parameter(
*Pie=3.14159 26535 89793 23846 26433 83279 50288 41971 69399d0)

```

```

p0tt=p0t*exp((et-ett)*omega/(xl-xk))*(pt/ptt)**(xk*omega/(xl-xk))
if (p0tt.gt.huge(1.0d0)) p0tt=2*pt ! check if p0tt is bigger then max possible
real number
if (isnan(p0tt)) p0tt=p0t ! check if p0tt is a number
if (p0tt.lt.tiny(1.0d0)) p0tt=0.001 ! check if p0tt is not too small
if (abs(2*ptt-p0tt).lt.CStol) then ! check if critical state is reached
  qtt=xM*ptt
  dq=qtt-qt
  dEpsqp=(dEpsq-dq/3.0/G)/omega
  dlam=xM*dEpsqp/2.0/ptt
  dEpsvp=0.0
else
  dEpsvp=(xl-xk)/(1+et)*log(p0tt/p0t)/omega
  dlam=dEpsvp/(2*ptt-p0tt)
end if
do i=1,6
  Sdtt(i)=(dEpsd(i)+Sdt(i)/2.0/G)/(3.0*dlam/xM**2.0+1.0/2.0/G)
end do
do i=1,3
  Stt(i)=Sdtt(i)+ptt
  Stt(i+3)=Sdtt(i+3)
end do
qtt=sqrt(3.0*(1.0/6.0*((Stt(1)-Stt(2))**2.0+(Stt(2)-Stt(3))**2.0+
*(Stt(3)-Stt(1))**2.0)+Stt(4)**2.0+Stt(5)**2.0+Stt(6)**2.0))
D3=(Stt(1)-ptt)*(Stt(2)-ptt)*(Stt(3)-ptt)+2.0*Stt(4)*
*Stt(5)*Stt(6)-(Stt(1)-ptt)*Stt(5)**2.0-(Stt(2)-
*ptt)*Stt(6)**2.0-(Stt(3)-ptt)*Stt(4)**2.0
theta=-1.0/3.0*asin(27.0/2.0*D3/qtt**3.0)
if (isnan(theta)) theta=-Pie/6.0
if (theta.gt.Pie/6.0) theta=Pie/6.0
if (theta.lt.-Pie/6.0) theta=-Pie/6.0
xMtt=xX*(1.0+yY*sin(3.0*theta))**zZ
if (isnan(xMtt)) xMtt=xX*(1.0-yY)**zZ
f=qtt**2.0/xMtt**2.0+ptt*(ptt-p0tt)
eta=qtt/ptt
return
end ! CalculateGMPPStress2
C*****
subroutine CalculateVectorA2(p,p0,q,theta,xM,Stress,xX,yY,zZ,Va)
C*****
! p          : Mean effective stress
! p0         : Hardening parameter
! xM         : Slope of CSL
! Stress(6)  : Array of stresses
! Va(6)      : Array of yield function derivatives
! q          : Deviatoric stress
! theta      : Lodes angle
! xX,yY,zZ   : Parameters that describe the dependency of M on Lode's angle
!
! ---Local variables---
!
! D3         : Third invariant of deviator stress
! DD3(6)     : Derivatives of third invariant of deviator stress
!
Implicit Double Precision (A-H,0-Z)
Dimension Stress(6), Va(6), DD3(6)
D3=(Stress(1)-p)*(Stress(2)-p)*(Stress(3)-p)+2.0*Stress(4)*
*Stress(5)*Stress(6)-(Stress(1)-p)*Stress(5)**2.0-(Stress(2)-
*p)*Stress(6)**2.0-(Stress(3)-p)*Stress(4)**2.0
! Derivatives of third stress invariant od deviator stress
DD3(1)=4.0/9.0*Stress(2)*Stress(3)-2.0/9.0*Stress(1)*Stress(2)-
*2.0/9.0*Stress(1)*Stress(3)+2.0/9.0*Stress(1)**2.0+

```

```

*1.0/3.0*Stress(4)**2.0+1.0/3.0*Stress(6)**2.0-
*2.0/3.0*Stress(5)**2.0-1.0/9.0*Stress(2)**2.0-
*1.0/9.0*Stress(3)**2.0
DD3(2)=4.0/9.0*Stress(1)*Stress(3)-2.0/9.0*Stress(1)*Stress(2)-
*2.0/9.0*Stress(2)*Stress(3)+2.0/9.0*Stress(2)**2.0+
*1.0/3.0*Stress(4)**2.0+1.0/3.0*Stress(5)**2.0-
*2.0/3.0*Stress(6)**2.0-1.0/9.0*Stress(1)**2.0-
*1.0/9.0*Stress(3)**2.0
DD3(3)=4.0/9.0*Stress(1)*Stress(2)-2.0/9.0*Stress(1)*Stress(3)-
*2.0/9.0*Stress(2)*Stress(3)+2.0/9.0*Stress(3)**2.0+
*1.0/3.0*Stress(5)**2.0+1.0/3.0*Stress(6)**2.0-
*2.0/3.0*Stress(4)**2.0-1.0/9.0*Stress(1)**2.0-
*1.0/9.0*Stress(2)**2.0
DD3(4)=2.0/3.0*Stress(1)*Stress(4)+2.0/3.0*Stress(2)*Stress(4)-
*4.0/3.0*Stress(3)*Stress(4)+2.0*Stress(5)*Stress(6)
DD3(5)=2.0/3.0*Stress(2)*Stress(5)+2.0/3.0*Stress(3)*Stress(5)-
*4.0/3.0*Stress(1)*Stress(5)+2.0*Stress(4)*Stress(6)
DD3(6)=2.0/3.0*Stress(1)*Stress(6)+2.0/3.0*Stress(3)*Stress(6)-
*4.0/3.0*Stress(2)*Stress(6)+2.0*Stress(4)*Stress(5)
!
if (q.lt.0.001) then
  Va(1)=(2.0*p-p0)/3.0+3.0*(Stress(1)-p)/xM**2.0
  Va(2)=(2.0*p-p0)/3.0+3.0*(Stress(2)-p)/xM**2.0
  Va(3)=(2.0*p-p0)/3.0+3.0*(Stress(3)-p)/xM**2.0
  Va(4)=6.0*Stress(4)/xM**2.0
  Va(5)=6.0*Stress(5)/xM**2.0
  Va(6)=6.0*Stress(6)/xM**2.0
else
  Va(1)=(2.0*p-p0)/3.0+3.0*(Stress(1)-p)/xM**2.0
  *-2.0*q**2.0*3.0**xX*yY*zZ*(1.0+yY*sin(3.0*theta))**(zZ-1.0)
  **cos(3.0*theta)/(xX*(1.0+yY*sin(3.0*theta))**zZ)**3.0*
  *(-1.0/3.0)*(1.0/sqrt(1.0-(27.0/2.0*D3/q**3.0)**2.0))*27.0/2.0*
  *((DD3(1)*q-3.0*D3*3.0*
  *(Stress(1)-p)/(2.0*q))/(q**4.0))
  if (isnan(Va(1)).or.abs(Va(1)).gt.huge(1.0d0).or.
  *abs(Va(1)).lt.tiny(1.0d0)) Va(1)=(2.0*p-p0)/3.0+3.0*(Stress(1)-p)/
  *xM**2.0
  Va(2)=(2.0*p-p0)/3.0+3.0*(Stress(2)-p)/xM**2.0
  *-2.0*q**2.0*3.0**xX*yY*zZ*(1.0+yY*sin(3.0*theta))**(zZ-1.0)
  **cos(3.0*theta)/(xX*(1.0+yY*sin(3.0*theta))**zZ)**3.0*
  *(-1.0/3.0)*(1.0/sqrt(1.0-(27.0/2.0*D3/q**3.0)**2.0))*27.0/2.0*
  *((DD3(2)*q-3.0*D3*3.0*
  *(Stress(2)-p)/(2.0*q))/(q**4.0))
  if (isnan(Va(2)).or.abs(Va(2)).gt.huge(1.0d0).or.
  *abs(Va(2)).lt.tiny(1.0d0)) Va(2)=(2.0*p-p0)/3.0+3.0*(Stress(2)-p)/
  *xM**2.0
  Va(3)=(2.0*p-p0)/3.0+3.0*(Stress(3)-p)/xM**2.0
  *-2.0*q**2.0*3.0**xX*yY*zZ*(1.0+yY*sin(3.0*theta))**(zZ-1.0)
  **cos(3.0*theta)/(xX*(1.0+yY*sin(3.0*theta))**zZ)**3.0*
  *(-1.0/3.0)*(1.0/sqrt(1.0-(27.0/2.0*D3/q**3.0)**2.0))*27.0/2.0*
  *((DD3(3)*q-3.0*D3*3.0*
  *(Stress(3)-p)/(2.0*q))/(q**4.0))
  if (isnan(Va(3)).or.abs(Va(3)).gt.huge(1.0d0).or.
  *abs(Va(3)).lt.tiny(1.0d0)) Va(3)=(2.0*p-p0)/3.0+3.0*(Stress(3)-p)/
  *xM**2.0
  Va(4)=6.0*Stress(4)/xM**2.0
  *-2.0*q**2.0*3.0**xX*yY*zZ*(1.0+yY*sin(3.0*theta))**(zZ-1.0)
  **cos(3.0*theta)/(xX*(1.0+yY*sin(3.0*theta))**zZ)**3.0*
  *(-1.0/3.0)*(1.0/sqrt(1.0-(27.0/2.0*D3/q**3.0)**2.0))*27.0/2.0*
  *((DD3(4)*q-3.0*D3*
  *3.0*Stress(4)/q)/(q**4.0))
  if (isnan(Va(4)).or.abs(Va(4)).gt.huge(1.0d0).or.

```

```

*abs(Va(4)).lt.tiny(1.0d0)) Va(4)=6.0*Stress(4)/xM**2.0
  Va(5)=6.0*Stress(5)/xM**2.0
*-2.0*q**2.0*3.0*xX*yY*zZ*(1.0+yY*sin(3.0*theta))**(zZ-1.0)
**cos(3.0*theta)/(xX*(1.0+yY*sin(3.0*theta))**zZ)**3.0*
*(-1.0/3.0)*(1.0/sqrt(1.0-(27.0/2.0*D3/q**3.0)**2.0))*27.0/2.0*
*((DD3(5)*q-3.0*D3*
*3.0*Stress(5)/q)/(q**4.0))
  if (isnan(Va(5)).or.abs(Va(5)).gt.huge(1.0d0).or.
*abs(Va(5)).lt.tiny(1.0d0)) Va(5)=6.0*Stress(5)/xM**2.0
  Va(6)=6.0*Stress(6)/xM**2.0
*-2.0*q**2.0*3.0*xX*yY*zZ*(1.0+yY*sin(3.0*theta))**(zZ-1.0)
**cos(3.0*theta)/(xX*(1.0+yY*sin(3.0*theta))**zZ)**3.0*
*(-1.0/3.0)*(1.0/sqrt(1.0-(27.0/2.0*D3/q**3.0)**2.0))*27.0/2.0*
*((DD3(6)*q-3.0*D3*
*3.0*Stress(6)/q)/(q**4.0))
  if (isnan(Va(6)).or.abs(Va(6)).gt.huge(1.0d0).or.
*abs(Va(6)).lt.tiny(1.0d0)) Va(6)=6.0*Stress(6)/xM**2.0
end if
return
end ! CalculateVectorA2

```

USRADDF.for

```
! Subroutines in this file:
!  
! Subroutine GetModelCount( nMod )  
! Subroutine GetModelName ( iMod , ModelName )  
! Subroutine GetParamCount( iMod , nParam )  
! Subroutine GetParamName ( iMod , iParam, ParamName )  
! Subroutine GetParamUnit ( iMod , iParam, Units )  
! Subroutine GetStateVarCount( iMod , nVar )  
! Subroutine GetStateVarName ( iMod , iVar, Name )  
! Subroutine GetStateVarUnit ( iMod , iVar, Unit )  
!  
! Local:  
! Subroutine GetParamAndUnit( iMod , iParam, ParamName, Units )  
! Subroutine GetStateVarNameAndUnit( iMod , iVar, Name, Unit )
```

```
Subroutine Get_Model_Count(nMod)  
!  
! Return the maximum model number (iMod) in this DLL  
!  
Integer (Kind=4) nMod  
  
nMod = 3 ! Maximum model number (iMod) in current DLL  
  
Return  
End ! GetModelCount
```

```
Subroutine Get_Model_Name( iMod , ModelName )  
!  
! Return the name of the different models  
!  
Integer iMod  
Character (Len= * ) ModelName
```

```
Select Case (iMod)  
Case (1)  
ModelName = 'GPM P'  
Case (2)  
ModelName = 'modEuler'  
Case (3)  
ModelName = 'RKDP'  
Case Default  
ModelName = 'not in DLL'  
End Select
```

```
Return  
End ! Get_Model_Name
```

```
Subroutine Get_Param_Count( iMod , nParam )  
!  
! Return the number of parameters of the different models  
!
```

```
Select Case (iMod)  
Case ( 1 )  
nParam = 11  
Case ( 2 )
```

```

    nParam = 12
Case ( 3 )
    nParam = 12
Case Default
    nParam = 0
End Select
Return
End ! Get_Param_Count

Subroutine GetParamAndUnit( iMod , iParam, ParamName, Units )
!
! Return the parameters name and units of the different models
!
! Units: use F          for force unit
!           L          for length unit
!           T          for time unit
!           %          for percentage
!           DEG        for degrees
!           TEMP       for temperature
!           TEMPINKELVIN for temperature in Kelvin
!
Character (Len=255) ParamName, Units, tName
Select Case (iMod)
Case (1)
    ! ModName = 'GPM P'
    Select Case (iParam)
    Case (1)
        ParamName = 'lambda' ; Units = '-'
    Case (2)
        ParamName = 'kappa' ; Units = '-'
    Case (3)
        ParamName = 'Mc' ; Units = '-'
    Case (4)
        ParamName = 'Me' ; Units = '-'
    Case (5)
        ParamName = 'nu' ; Units = '-'
    Case (6)
        ParamName = 'evoid0' ; Units = '-'
    Case (7)
        ParamName = 'OCR' ; Units = '-'
    Case (8)
        ParamName = 'POP' ; Units = 'F/L^2#'
    Case (9)
        ParamName = 'K0NC' ; Units = '-'
    Case (10)
        ParamName = 'G0,ref' ; Units = 'F/L^2#'
    Case (11)
        ParamName = 'gama0,7' ; Units = '-'
    Case Default
        ParamName = '???' ; Units = '???'
    End Select
Case (2)
    ! ModName = 'modEuler'
    Select Case (iParam)
    Case (1)
        ParamName = 'lambda' ; Units = '-'
    Case (2)
        ParamName = 'kappa' ; Units = '-'
    Case (3)
        ParamName = 'Mc' ; Units = '-'
    Case (4)
        ParamName = 'Me' ; Units = '-'

```



```

Case (5)
  ParamName = 'nu'      ; Units    = '-'
Case (6)
  ParamName = 'evold0'  ; Units    = '-'
Case (7)
  ParamName = 'OCR'     ; Units    = '-'
Case (8)
  ParamName = 'POP'     ; Units    = 'F/L^2#'
Case (9)
  ParamName = 'K0NC'    ; Units    = '-'
Case (10)
  ParamName = 'STOL'    ; Units    = '-'
Case (11)
  ParamName = 'G0,ref'  ; Units    = 'F/L^2#'
Case (12)
  ParamName = 'gama0,7' ; Units    = '-'
Case Default
  ParamName = '???'    ; Units    = '???'
End Select
Case (3)
! ModName = 'RKDP'
Select Case (iParam)
Case (1)
  ParamName = 'lambda'  ; Units    = '-'
Case (2)
  ParamName = 'kappa'   ; Units    = '-'
Case (3)
  ParamName = 'Mc'      ; Units    = '-'
Case (4)
  ParamName = 'Me'      ; Units    = '-'
Case (5)
  ParamName = 'nu'      ; Units    = '-'
Case (6)
  ParamName = 'evold0'  ; Units    = '-'
Case (7)
  ParamName = 'OCR'     ; Units    = '-'
Case (8)
  ParamName = 'POP'     ; Units    = 'F/L^2#'
Case (9)
  ParamName = 'K0NC'    ; Units    = '-'
Case (10)
  ParamName = 'STOL'    ; Units    = '-'
Case (11)
  ParamName = 'G0,ref'  ; Units    = 'F/L^2#'
Case (12)
  ParamName = 'gama0,7' ; Units    = '-'
Case Default
  ParamName = '???'    ; Units    = '???'
End Select
Case Default
! model not in DLL
ParamName = ' N/A '    ; Units    = ' N/A '
End Select

Return
End ! GetParamAndUnit

Subroutine Get_StateVar_Count( iMod , nVar )
!
! Return the number of state variables of the different models
!

```

```

Select Case (iMod)
Case (1)
  nVar = 180
Case (2)
  nVar = 180
Case (3)
  nVar = 180
Case Default
  nVar = 0
End Select

Return
End
Subroutine GetStateVarNameAndUnit( iMod , iVar, Name, Unit )
!
! Return the name and unit of the different state variables of the different
models
!
Character (Len=255) Name, Unit

Select Case (iMod)
Case (1)
  Select Case (iVar)
  Case (1)
    Name = 'RSV'          ; Unit = '-'
  Case (2)
    Name = 'Gama'        ; Unit = '-'
  Case (3)
    Name = 'R'           ; Unit = '-'
  Case (4)
    Name = 'PSIOC'       ; Unit = '-'
  Case (5)
    Name = 'PSINC'       ; Unit = '-'
  Case (6)
    Name = 'omega'       ; Unit = '-'
  Case (7)
    Name = 'p0'          ; Unit = 'F/L^2#'
  Case (8)
    Name = 'p0bs'        ; Unit = 'F/L^2#'
  Case (9)
    Name = 'Sigma,v,max' ; Unit = 'F/L^2#'
  Case (10)
    Name = 'OCR'         ; Unit = '-'
  Case (11)
    Name = 'IMC'         ; Unit = '-'
  Case (12)
    Name = 'Gt,ref'      ; Unit = 'F/L^2#'
  Case (13)
    Name = 'Gur,ref'     ; Unit = 'F/L^2#'
  Case (14)
    Name = 'G'           ; Unit = 'F/L^2#'
  Case (15)
    Name = 'deltaBG'     ; Unit = '-'
  Case (16)
    Name = 'theta'       ; Unit = 'DEG'
  Case (17)
    Name = 'X'           ; Unit = '-'
  Case (18)
    Name = 'Y'           ; Unit = '-'
  Case (19)
    Name = 'M'           ; Unit = '-'

```

```
Case (20)
  Name = 'NDF20'          ; Unit = '-'
Case (21)
  Name = 'B1 length'     ; Unit = '-'
Case (22)
  Name = 'B2 length'     ; Unit = '-'
Case (23)
  Name = 'B3 length'     ; Unit = '-'
Case (24)
  Name = 'B4 length'     ; Unit = '-'
Case (25)
  Name = 'B5 length'     ; Unit = '-'
Case (26)
  Name = 'B6 length'     ; Unit = '-'
Case (27)
  Name = 'B7 length'     ; Unit = '-'
Case (28)
  Name = 'B8 length'     ; Unit = '-'
Case (29)
  Name = 'B9 length'     ; Unit = '-'
Case (30)
  Name = 'B10 length'    ; Unit = '-'
Case (31)
  Name = 'B11 length'    ; Unit = '-'
Case (32)
  Name = 'B12 length'    ; Unit = '-'
Case (33)
  Name = 'B13 length'    ; Unit = '-'
Case (34)
  Name = 'B14 length'    ; Unit = '-'
Case (35)
  Name = 'B15 length'    ; Unit = '-'
Case (36)
  Name = 'B16 length'    ; Unit = '-'
Case (37)
  Name = 'B17 length'    ; Unit = '-'
Case (38)
  Name = 'B18 length'    ; Unit = '-'
Case (39)
  Name = 'B19 length'    ; Unit = '-'
Case (40)
  Name = 'B20 length'    ; Unit = '-'
Case (41)
  Name = 'B1 xx'         ; Unit = '-'
Case (42)
  Name = 'B2 xx'         ; Unit = '-'
Case (43)
  Name = 'B3 xx'         ; Unit = '-'
Case (44)
  Name = 'B4 xx'         ; Unit = '-'
Case (45)
  Name = 'B5 xx'         ; Unit = '-'
Case (46)
  Name = 'B6 xx'         ; Unit = '-'
Case (47)
  Name = 'B7 xx'         ; Unit = '-'
Case (48)
  Name = 'B8 xx'         ; Unit = '-'
Case (49)
  Name = 'B9 xx'         ; Unit = '-'
Case (50)
  Name = 'B10 xx'        ; Unit = '-'
```

```
Case (51)
  Name = 'B11 xx' ; Unit = '-'
Case (52)
  Name = 'B12 xx' ; Unit = '-'
Case (53)
  Name = 'B13 xx' ; Unit = '-'
Case (54)
  Name = 'B14 xx' ; Unit = '-'
Case (55)
  Name = 'B15 xx' ; Unit = '-'
Case (56)
  Name = 'B16 xx' ; Unit = '-'
Case (57)
  Name = 'B17 xx' ; Unit = '-'
Case (58)
  Name = 'B18 xx' ; Unit = '-'
Case (59)
  Name = 'B19 xx' ; Unit = '-'
Case (60)
  Name = 'B20 xx' ; Unit = '-'
Case (61)
  Name = 'B1 yy' ; Unit = '-'
Case (62)
  Name = 'B2 yy' ; Unit = '-'
Case (63)
  Name = 'B3 yy' ; Unit = '-'
Case (64)
  Name = 'B4 yy' ; Unit = '-'
Case (65)
  Name = 'B5 yy' ; Unit = '-'
Case (66)
  Name = 'B6 yy' ; Unit = '-'
Case (67)
  Name = 'B7 yy' ; Unit = '-'
Case (68)
  Name = 'B8 yy' ; Unit = '-'
Case (69)
  Name = 'B9 yy' ; Unit = '-'
Case (70)
  Name = 'B10 yy' ; Unit = '-'
Case (71)
  Name = 'B11 yy' ; Unit = '-'
Case (72)
  Name = 'B12 yy' ; Unit = '-'
Case (73)
  Name = 'B13 yy' ; Unit = '-'
Case (74)
  Name = 'B14 yy' ; Unit = '-'
Case (75)
  Name = 'B15 yy' ; Unit = '-'
Case (76)
  Name = 'B16 yy' ; Unit = '-'
Case (77)
  Name = 'B17 yy' ; Unit = '-'
Case (78)
  Name = 'B18 yy' ; Unit = '-'
Case (79)
  Name = 'B19 yy' ; Unit = '-'
Case (80)
  Name = 'B20 yy' ; Unit = '-'
Case (81)
  Name = 'B1 zz' ; Unit = '-'
```

```
Case (82)
  Name = 'B2 zz' ; Unit = '--'
Case (83)
  Name = 'B3 zz' ; Unit = '--'
Case (84)
  Name = 'B4 zz' ; Unit = '--'
Case (85)
  Name = 'B5 zz' ; Unit = '--'
Case (86)
  Name = 'B6 zz' ; Unit = '--'
Case (87)
  Name = 'B7 zz' ; Unit = '--'
Case (88)
  Name = 'B8 zz' ; Unit = '--'
Case (89)
  Name = 'B9 zz' ; Unit = '--'
Case (90)
  Name = 'B10 zz' ; Unit = '--'
Case (91)
  Name = 'B11 zz' ; Unit = '--'
Case (92)
  Name = 'B12 zz' ; Unit = '--'
Case (93)
  Name = 'B13 zz' ; Unit = '--'
Case (94)
  Name = 'B14 zz' ; Unit = '--'
Case (95)
  Name = 'B15 zz' ; Unit = '--'
Case (96)
  Name = 'B16 zz' ; Unit = '--'
Case (97)
  Name = 'B17 zz' ; Unit = '--'
Case (98)
  Name = 'B18 zz' ; Unit = '--'
Case (99)
  Name = 'B19 zz' ; Unit = '--'
Case (100)
  Name = 'B20 zz' ; Unit = '--'
Case (101)
  Name = 'B1 xy' ; Unit = '--'
Case (102)
  Name = 'B2 xy' ; Unit = '--'
Case (103)
  Name = 'B3 xy' ; Unit = '--'
Case (104)
  Name = 'B4 xy' ; Unit = '--'
Case (105)
  Name = 'B5 xy' ; Unit = '--'
Case (106)
  Name = 'B6 xy' ; Unit = '--'
Case (107)
  Name = 'B7 xy' ; Unit = '--'
Case (108)
  Name = 'B8 xy' ; Unit = '--'
Case (109)
  Name = 'B9 xy' ; Unit = '--'
Case (110)
  Name = 'B10 xy' ; Unit = '--'
Case (111)
  Name = 'B11 xy' ; Unit = '--'
Case (112)
  Name = 'B12 xy' ; Unit = '--'
```

```
Case (113)
  Name = 'B13 xy' ; Unit = '-'
Case (114)
  Name = 'B14 xy' ; Unit = '-'
Case (115)
  Name = 'B15 xy' ; Unit = '-'
Case (116)
  Name = 'B16 xy' ; Unit = '-'
Case (117)
  Name = 'B17 xy' ; Unit = '-'
Case (118)
  Name = 'B18 xy' ; Unit = '-'
Case (119)
  Name = 'B19 xy' ; Unit = '-'
Case (120)
  Name = 'B20 xy' ; Unit = '-'
Case (121)
  Name = 'B1 yz' ; Unit = '-'
Case (122)
  Name = 'B2 yz' ; Unit = '-'
Case (123)
  Name = 'B3 yz' ; Unit = '-'
Case (124)
  Name = 'B4 yz' ; Unit = '-'
Case (125)
  Name = 'B5 yz' ; Unit = '-'
Case (126)
  Name = 'B6 yz' ; Unit = '-'
Case (127)
  Name = 'B7 yz' ; Unit = '-'
Case (128)
  Name = 'B8 yz' ; Unit = '-'
Case (129)
  Name = 'B9 yz' ; Unit = '-'
Case (130)
  Name = 'B10 yz' ; Unit = '-'
Case (131)
  Name = 'B11 yz' ; Unit = '-'
Case (132)
  Name = 'B12 yz' ; Unit = '-'
Case (133)
  Name = 'B13 yz' ; Unit = '-'
Case (134)
  Name = 'B14 yz' ; Unit = '-'
Case (135)
  Name = 'B15 yz' ; Unit = '-'
Case (136)
  Name = 'B16 yz' ; Unit = '-'
Case (137)
  Name = 'B17 yz' ; Unit = '-'
Case (138)
  Name = 'B18 yz' ; Unit = '-'
Case (139)
  Name = 'B19 yz' ; Unit = '-'
Case (140)
  Name = 'B20 yz' ; Unit = '-'
Case (141)
  Name = 'B1 zx' ; Unit = '-'
Case (142)
  Name = 'B2 zx' ; Unit = '-'
Case (143)
  Name = 'B3 zx' ; Unit = '-'
```

```

Case (144)
  Name = 'B4 zx' ; Unit = '-'
Case (145)
  Name = 'B5 zx' ; Unit = '-'
Case (146)
  Name = 'B6 zx' ; Unit = '-'
Case (147)
  Name = 'B7 zx' ; Unit = '-'
Case (148)
  Name = 'B8 zx' ; Unit = '-'
Case (149)
  Name = 'B9 zx' ; Unit = '-'
Case (150)
  Name = 'B10 zx' ; Unit = '-'
Case (151)
  Name = 'B11 zx' ; Unit = '-'
Case (152)
  Name = 'B12 zx' ; Unit = '-'
Case (153)
  Name = 'B13 zx' ; Unit = '-'
Case (154)
  Name = 'B14 zx' ; Unit = '-'
Case (155)
  Name = 'B15 zx' ; Unit = '-'
Case (156)
  Name = 'B16 zx' ; Unit = '-'
Case (157)
  Name = 'B17 zx' ; Unit = '-'
Case (158)
  Name = 'B18 zx' ; Unit = '-'
Case (159)
  Name = 'B19 zx' ; Unit = '-'
Case (160)
  Name = 'B20 zx' ; Unit = '-'
Case (161)
  Name = 'B1 taut' ; Unit = '-'
Case (162)
  Name = 'B2 taut' ; Unit = '-'
Case (163)
  Name = 'B3 taut' ; Unit = '-'
Case (164)
  Name = 'B4 taut' ; Unit = '-'
Case (165)
  Name = 'B5 taut' ; Unit = '-'
Case (166)
  Name = 'B6 taut' ; Unit = '-'
Case (167)
  Name = 'B7 taut' ; Unit = '-'
Case (168)
  Name = 'B8 taut' ; Unit = '-'
Case (169)
  Name = 'B9 taut' ; Unit = '-'
Case (170)
  Name = 'B10 taut' ; Unit = '-'
Case (171)
  Name = 'B11 taut' ; Unit = '-'
Case (172)
  Name = 'B12 taut' ; Unit = '-'
Case (173)
  Name = 'B13 taut' ; Unit = '-'
Case (174)
  Name = 'B14 taut' ; Unit = '-'

```

```

Case (175)
  Name = 'B15 taut'      ; Unit = '-'
Case (176)
  Name = 'B16 taut'      ; Unit = '-'
Case (177)
  Name = 'B17 taut'      ; Unit = '-'
Case (178)
  Name = 'B18 taut'      ; Unit = '-'
Case (179)
  Name = 'B19 taut'      ; Unit = '-'
Case (180)
  Name = 'B20 taut'      ; Unit = '-'
Case Default
  Name='N/A'              ; Unit = '?'
End Select

```

Case (2)

```

Select Case (iVar)
Case (1)
  Name = 'RSV'           ; Unit = '-'
Case (2)
  Name = 'Gama'          ; Unit = '-'
Case (3)
  Name = 'R'             ; Unit = '-'
Case (4)
  Name = 'PSIOC'         ; Unit = '-'
Case (5)
  Name = 'PSINC'         ; Unit = '-'
Case (6)
  Name = 'omega'         ; Unit = '-'
Case (7)
  Name = 'p0'            ; Unit = 'F/L^2#'
Case (8)
  Name = 'p0bs'          ; Unit = 'F/L^2#'
Case (9)
  Name = 'Sigma,v,max'   ; Unit = 'F/L^2#'
Case (10)
  Name = 'OCR'           ; Unit = '-'
Case (11)
  Name = 'IMC'           ; Unit = '-'
Case (12)
  Name = 'Gt,ref'        ; Unit = 'F/L^2#'
Case (13)
  Name = 'Gur,ref'       ; Unit = 'F/L^2#'
Case (14)
  Name = 'G'             ; Unit = 'F/L^2#'
Case (15)
  Name = 'deltaBG'       ; Unit = '-'
Case (16)
  Name = 'theta'         ; Unit = 'DEG'
Case (17)
  Name = 'X'             ; Unit = '-'
Case (18)
  Name = 'Y'             ; Unit = '-'
Case (19)
  Name = 'M'             ; Unit = '-'
Case (20)
  Name = 'Error'         ; Unit = '-'
Case (21)
  Name = 'B1 length'     ; Unit = '-'
Case (22)
  Name = 'B2 length'     ; Unit = '-'
Case (23)

```



```
Name = 'B3 length' ; Unit = '-'
Case (24)
Name = 'B4 length' ; Unit = '-'
Case (25)
Name = 'B5 length' ; Unit = '-'
Case (26)
Name = 'B6 length' ; Unit = '-'
Case (27)
Name = 'B7 length' ; Unit = '-'
Case (28)
Name = 'B8 length' ; Unit = '-'
Case (29)
Name = 'B9 length' ; Unit = '-'
Case (30)
Name = 'B10 length' ; Unit = '-'
Case (31)
Name = 'B11 length' ; Unit = '-'
Case (32)
Name = 'B12 length' ; Unit = '-'
Case (33)
Name = 'B13 length' ; Unit = '-'
Case (34)
Name = 'B14 length' ; Unit = '-'
Case (35)
Name = 'B15 length' ; Unit = '-'
Case (36)
Name = 'B16 length' ; Unit = '-'
Case (37)
Name = 'B17 length' ; Unit = '-'
Case (38)
Name = 'B18 length' ; Unit = '-'
Case (39)
Name = 'B19 length' ; Unit = '-'
Case (40)
Name = 'B20 length' ; Unit = '-'
Case (41)
Name = 'B1 xx' ; Unit = '-'
Case (42)
Name = 'B2 xx' ; Unit = '-'
Case (43)
Name = 'B3 xx' ; Unit = '-'
Case (44)
Name = 'B4 xx' ; Unit = '-'
Case (45)
Name = 'B5 xx' ; Unit = '-'
Case (46)
Name = 'B6 xx' ; Unit = '-'
Case (47)
Name = 'B7 xx' ; Unit = '-'
Case (48)
Name = 'B8 xx' ; Unit = '-'
Case (49)
Name = 'B9 xx' ; Unit = '-'
Case (50)
Name = 'B10 xx' ; Unit = '-'
Case (51)
Name = 'B11 xx' ; Unit = '-'
Case (52)
Name = 'B12 xx' ; Unit = '-'
Case (53)
Name = 'B13 xx' ; Unit = '-'
Case (54)
```

```
Name = 'B14 xx' ; Unit = '--'
Case (55)
Name = 'B15 xx' ; Unit = '--'
Case (56)
Name = 'B16 xx' ; Unit = '--'
Case (57)
Name = 'B17 xx' ; Unit = '--'
Case (58)
Name = 'B18 xx' ; Unit = '--'
Case (59)
Name = 'B19 xx' ; Unit = '--'
Case (60)
Name = 'B20 xx' ; Unit = '--'
Case (61)
Name = 'B1 yy' ; Unit = '--'
Case (62)
Name = 'B2 yy' ; Unit = '--'
Case (63)
Name = 'B3 yy' ; Unit = '--'
Case (64)
Name = 'B4 yy' ; Unit = '--'
Case (65)
Name = 'B5 yy' ; Unit = '--'
Case (66)
Name = 'B6 yy' ; Unit = '--'
Case (67)
Name = 'B7 yy' ; Unit = '--'
Case (68)
Name = 'B8 yy' ; Unit = '--'
Case (69)
Name = 'B9 yy' ; Unit = '--'
Case (70)
Name = 'B10 yy' ; Unit = '--'
Case (71)
Name = 'B11 yy' ; Unit = '--'
Case (72)
Name = 'B12 yy' ; Unit = '--'
Case (73)
Name = 'B13 yy' ; Unit = '--'
Case (74)
Name = 'B14 yy' ; Unit = '--'
Case (75)
Name = 'B15 yy' ; Unit = '--'
Case (76)
Name = 'B16 yy' ; Unit = '--'
Case (77)
Name = 'B17 yy' ; Unit = '--'
Case (78)
Name = 'B18 yy' ; Unit = '--'
Case (79)
Name = 'B19 yy' ; Unit = '--'
Case (80)
Name = 'B20 yy' ; Unit = '--'
Case (81)
Name = 'B1 zz' ; Unit = '--'
Case (82)
Name = 'B2 zz' ; Unit = '--'
Case (83)
Name = 'B3 zz' ; Unit = '--'
Case (84)
Name = 'B4 zz' ; Unit = '--'
Case (85)
```

```

    Name = 'B5 zz' ; Unit = '-'
Case (86)
    Name = 'B6 zz' ; Unit = '-'
Case (87)
    Name = 'B7 zz' ; Unit = '-'
Case (88)
    Name = 'B8 zz' ; Unit = '-'
Case (89)
    Name = 'B9 zz' ; Unit = '-'
Case (90)
    Name = 'B10 zz' ; Unit = '-'
Case (91)
    Name = 'B11 zz' ; Unit = '-'
Case (92)
    Name = 'B12 zz' ; Unit = '-'
Case (93)
    Name = 'B13 zz' ; Unit = '-'
Case (94)
    Name = 'B14 zz' ; Unit = '-'
Case (95)
    Name = 'B15 zz' ; Unit = '-'
Case (96)
    Name = 'B16 zz' ; Unit = '-'
Case (97)
    Name = 'B17 zz' ; Unit = '-'
Case (98)
    Name = 'B18 zz' ; Unit = '-'
Case (99)
    Name = 'B19 zz' ; Unit = '-'
Case (100)
    Name = 'B20 zz' ; Unit = '-'
Case (101)
    Name = 'B1 xy' ; Unit = '-'
Case (102)
    Name = 'B2 xy' ; Unit = '-'
Case (103)
    Name = 'B3 xy' ; Unit = '-'
Case (104)
    Name = 'B4 xy' ; Unit = '-'
Case (105)
    Name = 'B5 xy' ; Unit = '-'
Case (106)
    Name = 'B6 xy' ; Unit = '-'
Case (107)
    Name = 'B7 xy' ; Unit = '-'
Case (108)
    Name = 'B8 xy' ; Unit = '-'
Case (109)
    Name = 'B9 xy' ; Unit = '-'
Case (110)
    Name = 'B10 xy' ; Unit = '-'
Case (111)
    Name = 'B11 xy' ; Unit = '-'
Case (112)
    Name = 'B12 xy' ; Unit = '-'
Case (113)
    Name = 'B13 xy' ; Unit = '-'
Case (114)
    Name = 'B14 xy' ; Unit = '-'
Case (115)
    Name = 'B15 xy' ; Unit = '-'
Case (116)

```

```

Name = 'B16 xy' ; Unit = '-'
Case (117)
Name = 'B17 xy' ; Unit = '-'
Case (118)
Name = 'B18 xy' ; Unit = '-'
Case (119)
Name = 'B19 xy' ; Unit = '-'
Case (120)
Name = 'B20 xy' ; Unit = '-'
Case (121)
Name = 'B1 yz' ; Unit = '-'
Case (122)
Name = 'B2 yz' ; Unit = '-'
Case (123)
Name = 'B3 yz' ; Unit = '-'
Case (124)
Name = 'B4 yz' ; Unit = '-'
Case (125)
Name = 'B5 yz' ; Unit = '-'
Case (126)
Name = 'B6 yz' ; Unit = '-'
Case (127)
Name = 'B7 yz' ; Unit = '-'
Case (128)
Name = 'B8 yz' ; Unit = '-'
Case (129)
Name = 'B9 yz' ; Unit = '-'
Case (130)
Name = 'B10 yz' ; Unit = '-'
Case (131)
Name = 'B11 yz' ; Unit = '-'
Case (132)
Name = 'B12 yz' ; Unit = '-'
Case (133)
Name = 'B13 yz' ; Unit = '-'
Case (134)
Name = 'B14 yz' ; Unit = '-'
Case (135)
Name = 'B15 yz' ; Unit = '-'
Case (136)
Name = 'B16 yz' ; Unit = '-'
Case (137)
Name = 'B17 yz' ; Unit = '-'
Case (138)
Name = 'B18 yz' ; Unit = '-'
Case (139)
Name = 'B19 yz' ; Unit = '-'
Case (140)
Name = 'B20 yz' ; Unit = '-'
Case (141)
Name = 'B1 zx' ; Unit = '-'
Case (142)
Name = 'B2 zx' ; Unit = '-'
Case (143)
Name = 'B3 zx' ; Unit = '-'
Case (144)
Name = 'B4 zx' ; Unit = '-'
Case (145)
Name = 'B5 zx' ; Unit = '-'
Case (146)
Name = 'B6 zx' ; Unit = '-'
Case (147)

```

```

Name = 'B7 zx' ; Unit = '-'
Case (148)
Name = 'B8 zx' ; Unit = '-'
Case (149)
Name = 'B9 zx' ; Unit = '-'
Case (150)
Name = 'B10 zx' ; Unit = '-'
Case (151)
Name = 'B11 zx' ; Unit = '-'
Case (152)
Name = 'B12 zx' ; Unit = '-'
Case (153)
Name = 'B13 zx' ; Unit = '-'
Case (154)
Name = 'B14 zx' ; Unit = '-'
Case (155)
Name = 'B15 zx' ; Unit = '-'
Case (156)
Name = 'B16 zx' ; Unit = '-'
Case (157)
Name = 'B17 zx' ; Unit = '-'
Case (158)
Name = 'B18 zx' ; Unit = '-'
Case (159)
Name = 'B19 zx' ; Unit = '-'
Case (160)
Name = 'B20 zx' ; Unit = '-'
Case (161)
Name = 'B1 taut' ; Unit = '-'
Case (162)
Name = 'B2 taut' ; Unit = '-'
Case (163)
Name = 'B3 taut' ; Unit = '-'
Case (164)
Name = 'B4 taut' ; Unit = '-'
Case (165)
Name = 'B5 taut' ; Unit = '-'
Case (166)
Name = 'B6 taut' ; Unit = '-'
Case (167)
Name = 'B7 taut' ; Unit = '-'
Case (168)
Name = 'B8 taut' ; Unit = '-'
Case (169)
Name = 'B9 taut' ; Unit = '-'
Case (170)
Name = 'B10 taut' ; Unit = '-'
Case (171)
Name = 'B11 taut' ; Unit = '-'
Case (172)
Name = 'B12 taut' ; Unit = '-'
Case (173)
Name = 'B13 taut' ; Unit = '-'
Case (174)
Name = 'B14 taut' ; Unit = '-'
Case (175)
Name = 'B15 taut' ; Unit = '-'
Case (176)
Name = 'B16 taut' ; Unit = '-'
Case (177)
Name = 'B17 taut' ; Unit = '-'
Case (178)

```

```

    Name = 'B18 taut'      ; Unit = '-'
Case (179)
    Name = 'B19 taut'      ; Unit = '-'
Case (180)
    Name = 'B20 taut'      ; Unit = '-'
Case Default
    Name='N/A'              ; Unit = '?'
End Select
Case (3)
Select Case (iVar)
Case (1)
    Name = 'RSV'           ; Unit = '-'
Case (2)
    Name = 'Gama'          ; Unit = '-'
Case (3)
    Name = 'R'             ; Unit = '-'
Case (4)
    Name = 'PSIOC'         ; Unit = '-'
Case (5)
    Name = 'PSINC'         ; Unit = '-'
Case (6)
    Name = 'omega'         ; Unit = '-'
Case (7)
    Name = 'p0'            ; Unit = 'F/L^2#'
Case (8)
    Name = 'p0bs'          ; Unit = 'F/L^2#'
Case (9)
    Name = 'Sigma,v,max'   ; Unit = 'F/L^2#'
Case (10)
    Name = 'OCR'           ; Unit = '-'
Case (11)
    Name = 'IMC'           ; Unit = '-'
Case (12)
    Name = 'Gt,ref'        ; Unit = 'F/L^2#'
Case (13)
    Name = 'Gur,ref'       ; Unit = 'F/L^2#'
Case (14)
    Name = 'G'             ; Unit = 'F/L^2#'
Case (15)
    Name = 'deltaBG'       ; Unit = '-'
Case (16)
    Name = 'theta'         ; Unit = 'DEG'
Case (17)
    Name = 'X'             ; Unit = '-'
Case (18)
    Name = 'Y'             ; Unit = '-'
Case (19)
    Name = 'M'             ; Unit = '-'
Case (20)
    Name = 'Error'         ; Unit = '-'
Case (21)
    Name = 'B1 length'     ; Unit = '-'
Case (22)
    Name = 'B2 length'     ; Unit = '-'
Case (23)
    Name = 'B3 length'     ; Unit = '-'
Case (24)
    Name = 'B4 length'     ; Unit = '-'
Case (25)
    Name = 'B5 length'     ; Unit = '-'
Case (26)
    Name = 'B6 length'     ; Unit = '-'

```

```
Case (27)
  Name = 'B7 length' ; Unit = '-'
Case (28)
  Name = 'B8 length' ; Unit = '-'
Case (29)
  Name = 'B9 length' ; Unit = '-'
Case (30)
  Name = 'B10 length' ; Unit = '-'
Case (31)
  Name = 'B11 length' ; Unit = '-'
Case (32)
  Name = 'B12 length' ; Unit = '-'
Case (33)
  Name = 'B13 length' ; Unit = '-'
Case (34)
  Name = 'B14 length' ; Unit = '-'
Case (35)
  Name = 'B15 length' ; Unit = '-'
Case (36)
  Name = 'B16 length' ; Unit = '-'
Case (37)
  Name = 'B17 length' ; Unit = '-'
Case (38)
  Name = 'B18 length' ; Unit = '-'
Case (39)
  Name = 'B19 length' ; Unit = '-'
Case (40)
  Name = 'B20 length' ; Unit = '-'
Case (41)
  Name = 'B1 xx' ; Unit = '-'
Case (42)
  Name = 'B2 xx' ; Unit = '-'
Case (43)
  Name = 'B3 xx' ; Unit = '-'
Case (44)
  Name = 'B4 xx' ; Unit = '-'
Case (45)
  Name = 'B5 xx' ; Unit = '-'
Case (46)
  Name = 'B6 xx' ; Unit = '-'
Case (47)
  Name = 'B7 xx' ; Unit = '-'
Case (48)
  Name = 'B8 xx' ; Unit = '-'
Case (49)
  Name = 'B9 xx' ; Unit = '-'
Case (50)
  Name = 'B10 xx' ; Unit = '-'
Case (51)
  Name = 'B11 xx' ; Unit = '-'
Case (52)
  Name = 'B12 xx' ; Unit = '-'
Case (53)
  Name = 'B13 xx' ; Unit = '-'
Case (54)
  Name = 'B14 xx' ; Unit = '-'
Case (55)
  Name = 'B15 xx' ; Unit = '-'
Case (56)
  Name = 'B16 xx' ; Unit = '-'
Case (57)
  Name = 'B17 xx' ; Unit = '-'
```

```
Case (58)
  Name = 'B18 xx' ; Unit = '-'
Case (59)
  Name = 'B19 xx' ; Unit = '-'
Case (60)
  Name = 'B20 xx' ; Unit = '-'
Case (61)
  Name = 'B1 yy' ; Unit = '-'
Case (62)
  Name = 'B2 yy' ; Unit = '-'
Case (63)
  Name = 'B3 yy' ; Unit = '-'
Case (64)
  Name = 'B4 yy' ; Unit = '-'
Case (65)
  Name = 'B5 yy' ; Unit = '-'
Case (66)
  Name = 'B6 yy' ; Unit = '-'
Case (67)
  Name = 'B7 yy' ; Unit = '-'
Case (68)
  Name = 'B8 yy' ; Unit = '-'
Case (69)
  Name = 'B9 yy' ; Unit = '-'
Case (70)
  Name = 'B10 yy' ; Unit = '-'
Case (71)
  Name = 'B11 yy' ; Unit = '-'
Case (72)
  Name = 'B12 yy' ; Unit = '-'
Case (73)
  Name = 'B13 yy' ; Unit = '-'
Case (74)
  Name = 'B14 yy' ; Unit = '-'
Case (75)
  Name = 'B15 yy' ; Unit = '-'
Case (76)
  Name = 'B16 yy' ; Unit = '-'
Case (77)
  Name = 'B17 yy' ; Unit = '-'
Case (78)
  Name = 'B18 yy' ; Unit = '-'
Case (79)
  Name = 'B19 yy' ; Unit = '-'
Case (80)
  Name = 'B20 yy' ; Unit = '-'
Case (81)
  Name = 'B1 zz' ; Unit = '-'
Case (82)
  Name = 'B2 zz' ; Unit = '-'
Case (83)
  Name = 'B3 zz' ; Unit = '-'
Case (84)
  Name = 'B4 zz' ; Unit = '-'
Case (85)
  Name = 'B5 zz' ; Unit = '-'
Case (86)
  Name = 'B6 zz' ; Unit = '-'
Case (87)
  Name = 'B7 zz' ; Unit = '-'
Case (88)
  Name = 'B8 zz' ; Unit = '-'
```



```
Case (89)
  Name = 'B9 zz' ; Unit = '-'
Case (90)
  Name = 'B10 zz' ; Unit = '-'
Case (91)
  Name = 'B11 zz' ; Unit = '-'
Case (92)
  Name = 'B12 zz' ; Unit = '-'
Case (93)
  Name = 'B13 zz' ; Unit = '-'
Case (94)
  Name = 'B14 zz' ; Unit = '-'
Case (95)
  Name = 'B15 zz' ; Unit = '-'
Case (96)
  Name = 'B16 zz' ; Unit = '-'
Case (97)
  Name = 'B17 zz' ; Unit = '-'
Case (98)
  Name = 'B18 zz' ; Unit = '-'
Case (99)
  Name = 'B19 zz' ; Unit = '-'
Case (100)
  Name = 'B20 zz' ; Unit = '-'
Case (101)
  Name = 'B1 xy' ; Unit = '-'
Case (102)
  Name = 'B2 xy' ; Unit = '-'
Case (103)
  Name = 'B3 xy' ; Unit = '-'
Case (104)
  Name = 'B4 xy' ; Unit = '-'
Case (105)
  Name = 'B5 xy' ; Unit = '-'
Case (106)
  Name = 'B6 xy' ; Unit = '-'
Case (107)
  Name = 'B7 xy' ; Unit = '-'
Case (108)
  Name = 'B8 xy' ; Unit = '-'
Case (109)
  Name = 'B9 xy' ; Unit = '-'
Case (110)
  Name = 'B10 xy' ; Unit = '-'
Case (111)
  Name = 'B11 xy' ; Unit = '-'
Case (112)
  Name = 'B12 xy' ; Unit = '-'
Case (113)
  Name = 'B13 xy' ; Unit = '-'
Case (114)
  Name = 'B14 xy' ; Unit = '-'
Case (115)
  Name = 'B15 xy' ; Unit = '-'
Case (116)
  Name = 'B16 xy' ; Unit = '-'
Case (117)
  Name = 'B17 xy' ; Unit = '-'
Case (118)
  Name = 'B18 xy' ; Unit = '-'
Case (119)
  Name = 'B19 xy' ; Unit = '-'
```

```
Case (120)
  Name = 'B20 xy' ; Unit = '-'
Case (121)
  Name = 'B1 yz' ; Unit = '-'
Case (122)
  Name = 'B2 yz' ; Unit = '-'
Case (123)
  Name = 'B3 yz' ; Unit = '-'
Case (124)
  Name = 'B4 yz' ; Unit = '-'
Case (125)
  Name = 'B5 yz' ; Unit = '-'
Case (126)
  Name = 'B6 yz' ; Unit = '-'
Case (127)
  Name = 'B7 yz' ; Unit = '-'
Case (128)
  Name = 'B8 yz' ; Unit = '-'
Case (129)
  Name = 'B9 yz' ; Unit = '-'
Case (130)
  Name = 'B10 yz' ; Unit = '-'
Case (131)
  Name = 'B11 yz' ; Unit = '-'
Case (132)
  Name = 'B12 yz' ; Unit = '-'
Case (133)
  Name = 'B13 yz' ; Unit = '-'
Case (134)
  Name = 'B14 yz' ; Unit = '-'
Case (135)
  Name = 'B15 yz' ; Unit = '-'
Case (136)
  Name = 'B16 yz' ; Unit = '-'
Case (137)
  Name = 'B17 yz' ; Unit = '-'
Case (138)
  Name = 'B18 yz' ; Unit = '-'
Case (139)
  Name = 'B19 yz' ; Unit = '-'
Case (140)
  Name = 'B20 yz' ; Unit = '-'
Case (141)
  Name = 'B1 zx' ; Unit = '-'
Case (142)
  Name = 'B2 zx' ; Unit = '-'
Case (143)
  Name = 'B3 zx' ; Unit = '-'
Case (144)
  Name = 'B4 zx' ; Unit = '-'
Case (145)
  Name = 'B5 zx' ; Unit = '-'
Case (146)
  Name = 'B6 zx' ; Unit = '-'
Case (147)
  Name = 'B7 zx' ; Unit = '-'
Case (148)
  Name = 'B8 zx' ; Unit = '-'
Case (149)
  Name = 'B9 zx' ; Unit = '-'
Case (150)
  Name = 'B10 zx' ; Unit = '-'
```

```

Case (151)
  Name = 'B11 zx'      ; Unit = '-'
Case (152)
  Name = 'B12 zx'      ; Unit = '-'
Case (153)
  Name = 'B13 zx'      ; Unit = '-'
Case (154)
  Name = 'B14 zx'      ; Unit = '-'
Case (155)
  Name = 'B15 zx'      ; Unit = '-'
Case (156)
  Name = 'B16 zx'      ; Unit = '-'
Case (157)
  Name = 'B17 zx'      ; Unit = '-'
Case (158)
  Name = 'B18 zx'      ; Unit = '-'
Case (159)
  Name = 'B19 zx'      ; Unit = '-'
Case (160)
  Name = 'B20 zx'      ; Unit = '-'
Case (161)
  Name = 'B1 taut'     ; Unit = '-'
Case (162)
  Name = 'B2 taut'     ; Unit = '-'
Case (163)
  Name = 'B3 taut'     ; Unit = '-'
Case (164)
  Name = 'B4 taut'     ; Unit = '-'
Case (165)
  Name = 'B5 taut'     ; Unit = '-'
Case (166)
  Name = 'B6 taut'     ; Unit = '-'
Case (167)
  Name = 'B7 taut'     ; Unit = '-'
Case (168)
  Name = 'B8 taut'     ; Unit = '-'
Case (169)
  Name = 'B9 taut'     ; Unit = '-'
Case (170)
  Name = 'B10 taut'    ; Unit = '-'
Case (171)
  Name = 'B11 taut'    ; Unit = '-'
Case (172)
  Name = 'B12 taut'    ; Unit = '-'
Case (173)
  Name = 'B13 taut'    ; Unit = '-'
Case (174)
  Name = 'B14 taut'    ; Unit = '-'
Case (175)
  Name = 'B15 taut'    ; Unit = '-'
Case (176)
  Name = 'B16 taut'    ; Unit = '-'
Case (177)
  Name = 'B17 taut'    ; Unit = '-'
Case (178)
  Name = 'B18 taut'    ; Unit = '-'
Case (179)
  Name = 'B19 taut'    ; Unit = '-'
Case (180)
  Name = 'B20 taut'    ; Unit = '-'
Case Default
  Name='N/A'           ; Unit = '?'

```

```
End Select
Case Default
    Name='N/A'           ; Unit = '?'
End Select

Return
End
```

OKMSGIVF.for

```
Subroutine OK_MessageBox(t)
  use dfwin
  use dfwinty
  integer (kind=Int_Ptr_Kind()) hWnd

  character*(*) t

  character(len=256) :: msg, title

! Display a messagebox with an OK button
! Note that all strings must be null terminated for C's sake

  msg = Trim(t) // char(0)
  title = 'UDSM' // char(0)

  hWnd = 0

  iret = MessageBox( hWnd,
*                   msg,
*                   title,
*                   MB_OK )

  Return
End
```

MYMODEL1.for

```
Subroutine MyModel1 ( IDTask, iMod, IsUndr,
*                   iStep, iTer, iEl, Int,
*                   X, Y, Z,
*                   Time0, dTime,
*                   Props, Sig0, Swp0, StVar0,
*                   dEps, D, BulkW,
*                   Sig, Swp, StVar, ipl,
*                   nStat,
*                   NonSym, iStrsDep, iTimeDep, iTang,
*                   iAbort )
!
! Purpose: User supplied soil model, HASP-E, implicit gpm with mean effective stress
as governing parameter
! Depending on IDTask, 1 : Initialize state variables
!                       2 : calculate stresses,
!                       3 : calculate material stiffness matrix
!                       4 : return number of state variables
!                       5 : inquire matrix properties
!                           return switch for non-symmetric D-matrix
!                           stress/time dependent matrix
!                       6 : calculate elastic material stiffness matrix
! Arguments:
! I/O   Type
! IDTask  I   I   : see above
! iMod    I   I   : model number (1..10)
! IsUndr  I   I   : =1 for undrained, 0 otherwise
! iStep   I   I   : Global step number
! iter    I   I   : Global iteration number
! iel     I   I   : Global element number
! Int     I   I   : Global integration point number
! X       I   R   : X-Position of integration point
! Y       I   R   : Y-Position of integration point
! Z       I   R   : Z-Position of integration point
! Time0   I   R   : Time at start of step
! dTime   I   R   : Time increment
! Props   I   R( ) : List with model parameters
! Sig0    I   R( ) : Stresses at start of step
! Swp0    I   R   : Excess pore pressure start of step
! StVar0  I   R( ) : State variable at start of step
! dEps    I   R( ) : Strain increment
! D       I/O  R(, ) : Material stiffness matrix
! BulkW   I/O  R   : Bulkmodulus for water (undrained only)
! Sig     0   R( ) : Resulting stresses
! Swp     0   R   : Resulting excess pore pressure
! StVar   0   R( ) : Resulting values state variables
! ipl     0   I   : Plasticity indicator
! nStat   0   I   : Number of state variables
! NonSym  0   I   : Non-Symmetric D-matrix ?
! iStrsDep 0   I   : =1 for stress dependent D-matrix
! iTimeDep 0   I   : =1 for time dependent D-matrix
! iAbort  0   I   : =1 to force stopping of calculation
!
! Implicit Double Precision (A-H, O-Z)
!
! Dimension Props(*), Sig0(*), StVar0(*), dEps(*), D(6,6),
*           Sig(*), StVar(*)
!
!--- Local variables (Arrays)
!
```

```

! BT(20)          : Array which contains whether the strings are slack or taut
(IDTask=2)
! dStrain(6)      : Array of strain increments (compression is positive) (IDTask=2)
! dStraind(6)     : Array of deviatoric strain increments (compression is positive)
(IDTask=2)
! dStress(6)      : Array of stress increments (compression is positive) (IDTask=2)
! SLen(20)        : Array with length of strings (IDTask=1, IDTask=2)
! Straint(6)      : Array of strains at start of step (compression is positive)
(IDTask=2)
! Straintt(6)     : Array of strains at the end of step (compression is positive)
(IDTask=2)
! Stressdt(6)     : Array of deviatoric stresses at start of step (compression is
positive) (IDTask=2)
! Stresst(6)      : Array of stresses at start of step (compression is positive)
(IDTask=2)
! Stresstt(6)     : Array of stresses at the end of step (compression is positive)
(IDTask=2)
!
!--- Local variables (Matrices)
!
! SH(20,6)        : Matrix which contains the position of each brick (IDTask=2)
!
!--- Local variables
!
! Bulk            : Bulk modulus at start of step (IDTask=2, IDTask=4, IDTask=6)
! BulkW           : Bulk modulus of water (IDTask=4, IDTask=6)
! CStol           : Tolerance for assessing whether the critical state is reached
(IDTask=2)
! dBG            : Proportion of shear stiffness belonging to one brick (IDTask=1,
IDTask=2)
! dEpsV           : Increment of volumetric strain (IDTask=2)
! dEpsq           : Increment of deviatoric strain (IDTask=2)
! dp             : Increment of mean effective stress in zero search (IDTask=2)
! dShearStrain    : Increment of shear strain invariant (IDTask=2)
! dSwp           : Increment of excess pore water pressure (IDTask=2)
! D1,D2 and D3    : Auxilliary variables for calculation od D matrix (IDTask=4,
IDTask=6)
! EpsV           : Volumetric strain (IDTask=3, IDTask=6)
! et             : Void ratio at start of step (IDTask=2, IDTask=4, IDTask=6)
! etai           : Initial stress ratio (IDTask=1)
! etamax         : Max previous stress ratio (IDTask=1)
! etat           : Shear ratio at start of step (IDTask=2)
! ett           : Void ratio at the end of step (IDTask=2)
! e0             : Initial void ratio (IDTask=1, IDTask=2, IDTask=4, IDTask=6)
! fprim          : First derivative of yield function in GPMP algorithm (IDTask=2)
! fsecond        : Second derivative of yield function in GPMP algorithm (IDTask=2)
! ftol           : Tolerance for zero value of yield function (IDTask=2)
! f1,f2,f3,...   : Values of yield function for p1,p2,p3,... (IDTask=2)
! G             : Shear modulus at start of step (IDTask=2, IDTask=4, IDTask=6)
! Gbrick         : Shear modulus when i brick strings are taut (IDTask=1)
! Gur           : Shear modulus for URL at pref=100 kPa (IDTask=1, IDTask=2)
! G0            : Small strain shear modulus at pref=100 (IDTask=1, IDTask=2,
IDTask=4, IDTask=6)
! G1            : Shear modulus at start of step for p=100 kPa (IDTask=1, IDTask=2)
! G2            : Shear modulus at the end of step for p=100 kPa (IDTask=2)
! Gama          : specific volume for CLS at p'=1 kPa (IDTask=1, IDTask=2)
! ilam          : Auxilliary counter in GPMP algorithm (IDTask)=2
! kmax          : Max number of iterations for zero search (IDTask=2)
! k             : counter of number of iterations for zero search (IDTask=2)
! nStatV        : Number of used state variables
! OCR           : Input value of overconsolidation ratio (IDTask=1)
! omega         : Hardening coefficient (IDTask=1, IDTask=2)

```

```

! omegalim      : Limiting value for hardening coefficient in order to avoid
numerical instability (IDTask=2)
! pe           : Elastic trial value of p in GPMP algorithm (IDTask=2)
! pi           : Initial value of mean effective stress
! Pie          : Value of Pi (IDTask=1)
! pmax         : Max previous mean effective stress (IDTask=1)
! POM1, POM2, ...: Auxilliary real variables
! POP          : Input value of preoverburden pressure (IDTask=1)
! pnew         : New guess of p
! pt           : mean effective stress at start of step (IDTask=2, IDTask=4,
IDTask=6)
! ptt          : mean effective stress at the end of step (IDTask=2)
! PSINC        : State parameter for NC soil (IDTask=1, IDTask=2)
! PSIOC        : State parameter for OC soil ((IDTask=1, IDTask=2)
! p0           : Hardening parameter for yield surface (IDTask=1, IDTask=2)
! p0bs         : Hardening parameter for bounding surface (IDTask=1, IDTask=2)
! p0t          : Hardening parameter for current yield surface at start of step
(IDTask=2)
! p0tt         : Hardening parameter for current yield surface at start of step
(IDTask=2)
! p1,p2,p3,... : guesses of p in GPMP algorithm (IDTask=2)
! qi           : Initial value of shear deviator stress (IDTask=1)
! qmax         : Max previous shear deviator stress (IDTask=1)
! qt           : Shear deviator stress at start of step (IDTask=2)
! qtt          : Shear deviator stress at the end of step (IDTask=2)
! R            : Isotropic OCR ratio (IDTask=1, IDTask=2)
! Rmax         : Max value of R in order to avoid numerical instability (IDTask=1,
IDTask=2)
! Rmin         : Min value of R in order to avoid non-possible values (IDTask=1,
IDTask=2)
! strain70     : Reference shear strain (IDTask=1)
! Sxmax        : Max previous horizontal stress in x direction (IDTask=1)
! Symax        : Max previous vertical stress (IDTask=1)
! Szmax        : Max previous horizontal stress in y direction (IDTask=1)
! thetat       : Lodes angle at start of step
! xk           : kappa, slope of U/R line (IDTask=1, IDTask=2, IDTask=4, IDTask=6)
! xka          : small strain kappa (IDTask=1, IDTask=2)
! xKNC         : Ratio of horizontal and vertical stresses at NC state (IDTask=1)
! xl           : lambda, slope of ICL (IDTask=1, IDTask=2, IDTask=4, IDTask=6))
! xM           : M, slope of CSL in p-q plane at the start of step (IDTask=1,
IDTask=2)
! xMc          : slope of CSL in p-q plane in TXC (IDTask=1, IDTask=2)
! xMe          : slope of CSL in p-q plane in TXE (IDTask=1, IDTask=2)
! xNu          : Poissons ratio (IDTask=1, IDTask=2, IDTask=4, IDTask=6)
! xX,yY,zZ     : Parameters that describe the dependency of M on Lode's angle
!
!--- Auxilliaty Subroutines
!
! CalculateStressInvariants : Calculates p,q,eta and theta
! CalculateStressDeviator   : Calculates array of deviatoric stress
! CalculateVoidRatio        : Calculates current void ratio
! CalculateStrainInvariants : Calculates volumetric strain, deviatoric strain
invariant and shear strain invariant
! CalculateStrainDeviator   : Calculates array of deviatoric strain
! CalculateElasticModuli    : Calculates bulk and shear modulus at start of step
! CalculateOmega            : Calculates hardening coefficient
! CalculateGMPYield2        : Calculates value of yield function for input mean
effective stress using GPM integration with mean effective stress as governing
parameter
! CalculateGMPStress2       : Calculates stresses at the end of step for input mean
effective stress using GPM integration with mean effective stress as governing
parameter

```



```

! MZEROR           : Fills real array with zeros
! COPYRVEC        : Copies a Double array R1 with Dimension K to R2
! CalculateElasticModuliBrick: Calculates shear modulus for Brick model
!
! Dimension Stresst(6), Straint(6), dStrain(6), Straintt(6),
! *          Stressdt(6), dStraind(6), dStress(6), Stresstt(6),
! *          Stressdtt(6), dSig(6), SLen(20), SH(20,6), BT(20)
! Parameter (ftol=1.D-9, dp=1.D-6, CStol=1.D-6,
! *omegalim=15000, Rmax=100.0, Rmin=1.0,
! *Pie=3.14159 26535 89793 23846 26433 83279 50288 41971 69399d0)

! Expected contents of Props(1..?)
! 1 : lambda      slope of ICL
! 2 : kappa       slope of U/R line
! 3 : Mc          slope of CSL in p-q plane in TXC
! 4 : Me          slope of CSL in p-q plane in TXE
! 5 : nu          Poisson ratio
! 5 : evoid0      Initial void ration
! 7 : OCR         overconsolidation ratio
! 8 : POP         preoverburden pressure
! 9 : K0NC        ratio of horizontal and vertical stresses
! 10 : G0,ref    small strain shear modulus at pref=100 kPa
! 11 : gama0,7   reference strain

nStatV = 180
If (IDTask.Eq. 1) Then ! Initialize state variables
    if (StVar0(1).eq.0) then !Reset state variables
        ! Get Props
        xl=Props(1)
        xk=Props(2)
        xMc=Props(3)
        xMe=Props(4)
        if (xMe.eq.0.0) xMe=xMc
        xNu=Props(5)
        e0=Props(6)
        OCR=Props(7)
        POP=Props(8)
        xKNC=Props(9)
        G0=Props(10)
        strain70=Props(11)
        If (G0.gt.0.0.and.strain70.eq.0.0) Then
            Call OK_MessageBox(' Invalid value for "gama0.7" (param 11)')
        *)
            iAbort = 1
            Return
        End If
        ! Calculate Gama from OCR, POP
        Symax=-OCR*Sig0(2)+POP
        Sxmax=xKNC*Symax
        Szmax=xKNC*Symax
        pmax=(Symax+Sxmax+Szmax)/3
        qmax=abs(Symax-Sxmax)
        etamax=qmax/pmax
        pi=-(Sig0(1)+Sig0(2)+Sig0(3))/3.0
        qi=sqrt(3.0*(1.0/6.0*((Sig0(1)-Sig0(2))**2.0+
        *(Sig0(2)-Sig0(3))**2.0+(Sig0(3)-Sig0(1))**2.0)+Sig0(4)**2.0+
        *Sig0(5)**2.0+Sig0(6)**2.0))
        etai=qi/pi
        xM=xMc
        POM1=(pmax*(etamax**2.0+xM**2.0)/2.0/xM**2.0)**((xl-xk)/xl)
        POM2=pi**((xk/xl)
        Gama=1.0+e0+xl*log(POM1*POM2)

```

```

PSIOC=1.0+e0+xl*log(pi)-Gama
PSINC=(xl-xk)*log(2.0*xM**2.0/(xM**2.0+etai**2.0))
R=exp((PSINC-PSIOC)/(xl-xk))
If (R.gt.Rmax) then
    Gama=1.0+e0+xl*log(pi)-PSINC+(xl-xk)*log(Rmax)
    R=Rmax
end if
if (R.lt.Rmin) then
    Gama=1.0+e0+xl*log(pi)-PSINC+(xl-xk)*log(Rmin)
    R=Rmin
end if
omega=(1.0+(PSINC-PSIOC)/PSINC)*R
p0=qi**2.0/(pi*xM**2.0)+pi
p0bs=R*p0
! Check if value of Poissons ratio is ok
If (xNu.Gt.0.5d0-1d-6 .Or. xNu.Lt.-1d-6) Then
    Call OK_MessageBox(' Invalid value for "nu" (param 5)')
    iAbort = 1
    Return
End If
if (xNu.Lt.0) then
    xNu = 0.0
    Props(5) = xNu
end if
if (xNu.Gt.0.495d0) then
    xNu = 0.495
    Props(5) = xNu
end if
! Initialize Brick model
Gur=3.0*(1.0-2.0*xNu)/(2.0*(1.0+xNu))*(1.0+e0)/xk*100.0
if (G0.lt.Gur) G0=Gur
dBG=(G0-Gur)/(20*G0)
do i=1,20
    Gbrick=(1.0-(real(i)-0.5)*dBG)*G0
    SLen(i)=strain70/0.385*(sqrt(G0/Gbrick)-1.0)
end do
G=G0*pi/100.0
! Variables X, Y and Z for determination of parameter M
zZ=-0.229
xX=((xMc**(1.0/zZ)+xMe**(1.0/zZ))/2.0)**zZ
yY=(1.0-(xMc/xMe)**(1.0/zZ))/(1.0+(xMc/xMe)**(1.0/zZ))
! Fill state variable array
do i=1,nStatV
    StVar0(i)=0.0
end do
StVar0(1)=1.0
StVar0(2)=Gama
StVar0(3)=R
StVar0(4)=PSIOC
StVar0(5)=PSINC
StVar0(6)=omega
StVar0(7)=p0
StVar0(8)=p0bs
StVar0(9)=Symax
StVar0(10)=Symax/(-Sig0(2))
StVar0(11)=1.0
StVar0(12)=G0
StVar0(13)=Gur
StVar0(14)=G
StVar0(15)=dBG
StVar0(16)=-30.0
StVar0(17)=xX

```

```

    stVar0(18)=yY
    StVar0(19)=xM
    do i=1,20
        StVar0(20+i)=SLen(i)
    end do
end if
If (iel+int.eq.2) Call WriVec(1, 'Props', Props, 11)
End If ! IDTask = 1
If (IDTask .Eq. 2) Then ! Calculate stresses
    If (IsUndr.Eq.1) Then
        dEpsV = dEps(1) + dEps(2) + dEps(3)
        dSwp = BulkW * dEpsV
        Swp = Swp0 + dSwp
    Else
        Swp = Swp0
    End If
    If (iEl+Int+iter.Eq.3 .And. iStep.Lt.10) Then
        Call WriMat( -1, 'D66', D, 6, 6, 6 )
    End If
! Material properties
xl=Props(1)
xk=Props(2)
xNu=Props(5)
e0=Props(6)
G0=Props(10)
! State variables
Gama=StVar0(2)
G1=StVar0(12)
Gur=StVar0(13)
dBG=StVar0(15)
xX=StVar0(17)
yY=StVar0(18)
zZ=-0.229
do i=1,20
    SLen(i)=StVar0(20+i)
end do
do i=1,20
    do j=1,6
        SH(i,j)=StVar0(20+20*j+i)
    end do
end do
! Reversing sign for input stresses and strains
do i=1,6
    Stresst(i)=-Sig0(i)
    Straint(i)=-dEps(i+6)
    dStrain(i)=-dEps(i)
    Straintt(i)=Straint(i)+dStrain(i)
end do
!Stress and strain invariants
Call CalculateStressInvariants(Stresst,pt,qt,etat,thetat)
Call CalculateStressDeviator(Stresst,pt,Stressdt)
Call CalculateVoidRatio(Straint,e0,et)
Call CalculateVoidRatio(Straintt,e0,ett)
Call CalculateStrainInvariants(dStrain,dEpsv,dEpsq,dShearStrain)
Call CalculateStrainDeviator(dStrain,dEpsv,dStraind)
!Calculate value of parameter M at the start of step
xM=xX*(1.0+yY*sin(3.0*thetat))*zZ
if (isnan(xM)) xM=xX*(1.0-yY)*zZ
!Calculate OMEGA - hardening coefficient at the beginning of increment
Call CalculateOmega(xl,xk,xM,Gama,pt,qt,et,omegalim,Rmax,Rmin,
*omega,R,PSIOC,PSINC)
!Material modules

```

```

    if (G0.eq.0.0) then
        Call CalculateElasticModuli(et,pt,xk,xNu,Bulk,G)
    else
        Call CalculateElasticModuliBrick(BT,Straintt,SH,SLen,G1,
*G2,Gur,G0,G,et,pt,xNu,xka,dBG,Bulk)
        xk=xka
    end if
!Hardening parameter for curent yield surface
p0t=qt**2.0/pt/xM**2.0+pt
! Stress determination GPM P algorithm
!
! Trial step
pe=pt*exp((et-ett)/xk)
ipl=4
k=1
ilam=1
kmax=25
p1=pe
100 if (p1.lt.0.001) p1=0.001
    if (k.gt.kmax) goto 110
    Call CalculateGPMPYield2(xl,xk,xM,et,ett,G,pt,qt,p0t,dEpsq,
*dStraind,Stressdt,CStol,omega,p1,f1,xX,yY,zZ)
    if (abs(f1).lt.ftol) then
        goto 110
    else
        p2=p1+dp
        p3=p1-dp
        Call CalculateGPMPYield2(xl,xk,xM,et,ett,G,pt,qt,p0t,dEpsq,
*dStraind,Stressdt,CStol,omega,p2,f2,xX,yY,zZ)
        Call CalculateGPMPYield2(xl,xk,xM,et,ett,G,pt,qt,p0t,dEpsq,
*dStraind,Stressdt,CStol,omega,p3,f3,xX,yY,zZ)
        if (isnan((f2-f3)/2.0/dp)) then
            if (ilam.eq.2) goto 110
            ilam=2
            pomA=exp((et-ett)*omega/(xl-xk))
            p1=pt*(1.0/pomA)**((xk-xl)/xk/omega)
            if (p1.lt.0.001) p1=0.001
            k=1
            goto 100
        End If
        fprim=(f2-f3)/2.0/dp
        fsecond=(f2-2.0*f1+f3)/dp**2.0
        if (isnan(p1-2.0*f1*fprim/(2.0*fprim**2.0-f1*fsecond))) then
            if (ilam.eq.2) goto 110
            ilam=2
            pomA=exp((et-ett)*omega/(xl-xk))
            p1=pt*(1.0/pomA)**((xk-xl)/xk/omega)
            if (p1.lt.0.001) p1=0.001
            k=1
            goto 100
        end if
        pnew=p1-2.0*f1*fprim/(2.0*fprim**2.0-f1*fsecond)
        k=k+1
        p1=pnew
        if (p1.lt.0.001) then
            p1=0.001
            goto 110
        end if
        goto 100
    end if
110 continue
    ptt=p1

```

```

    Call CalculateGPMPStress2(xl,xk,xM,et,ett,G,pt,qt,p0t,dEpsq,
*dStraind,Stressdt,CStol,omega,ptt,dlam,p0tt,Stresstt,xX,yY,zZ)
    if (dlam.lt.0.0.and.ilam.eq.1) then
        ilam=2
        pomA=exp((et-ett)*omega/(xl-xk))
        p1=pt*(1.0/pomA)**((xk-xl)/xk/omega)
        if (p1.lt.0.001) p1=0.001
        k=1
        goto 100
    end if
    if (dlam.lt.0.0.and.ilam.eq.2) then
        pomA=exp((et-ett)*omega/(xl-xk))
        ptt=pt*(1.0/pomA)**((xk-xl)/xk/omega)
        if (ptt.lt.0.001) ptt=0.001
        Call CalculateGPMPStress2(xl,xk,xM,et,ett,G,pt,qt,p0t,dEpsq,
*dStraind,Stressdt,CStol,omega,ptt,dlam,p0tt,Stresstt,xX,yY,zZ)
    end if
    if (isnan(ptt).or.ptt.gt.huge(1.0d0)) then
        ptt=pt
        Stresstt=Stresst
        p0tt=p0t
    end if
!
!Return stresses
do i=1,6
    Sig(i)=-Stresstt(i)
end do
150 continue
!Update state variables
StVar(1)=StVar0(1)
StVar(2)=StVar0(2)
StVar(3)=R
StVar(4)=PSIOC
StVar(5)=PSINC
StVar(6)=omega
StVar(7)=p0t
StVar(8)=p0t*R
If (StVar0(9).lt.Stresstt(2)) then
    StVar(9)=Stresstt(2)
else
    StVar(9)=StVar0(9)
end if
StVar(10)=StVar(9)/Stresstt(2)
StVar(11)=1.0
StVar(16)=thetat*180.0/Pie
StVar(17)=StVar0(17)
StVar(18)=StVar0(18)
StVar(19)=xM
if (G0.eq.0.0) then
    StVar(12)=Gur
    StVar(13)=StVar0(13)
    StVar(14)=G
    StVar(15)=StVar0(15)
else
    StVar(12)=G2
    StVar(13)=StVar0(13)
    StVar(14)=G
    StVar(15)=StVar0(15)
do i=1,20
    StVar(20+i)=StVar0(20+i)
end do
do i=1,20

```

```

        do j=1,6
            StVar(20+20*j+i)=SH(i,j)
        end do
    end do
    do i=1,20
        StVar(160+i)=BT(i)
    end do
end if

End If ! IDTask = 2
If ( IDTask .Eq. 3 .Or.
*   IDTask .Eq. 6      ) Then ! Calculate D-Matrix
    xk=Props(2)
    xNu=Props(5)
    e0=Props(6)
    G0=Props(10)
    pt=-(Sig0(1)+Sig0(2)+Sig0(3))/3.0
    if (G0.eq.0.0) then
        EpsV=-(dEps(7)+dEps(8)+dEps(9))
        et=(1.0+e0)*exp(-EpsV)-1.0
        Call CalculateElasticModuli(et,pt,xk,xNu,Bulk,G)
    else
        G=G0*pt/100.0
        if (G.lt.1.0) G=1.0
        Bulk=2.0*(1.0+xNu)/(3.0*(1.0-2.0*xNu))*G
    end if
    D1=Bulk+4.0*G/3.0
    D2=Bulk-2.0*G/3.0
    D3=G
    Call MZeroR(D,36)
    D(1,1)=D1
    D(2,2)=D1
    D(3,3)=D1
    D(1,2)=D2
    D(1,3)=D2
    D(2,1)=D2
    D(2,3)=D2
    D(3,1)=D2
    D(3,2)=D2
    D(4,4)=D3
    D(5,5)=D3
    D(6,6)=D3
    BulkW = 0
    If (IsUndr.Eq.1) Then
        ! BulkW = ...
        xNu_U = 0.495d0
        Fac=(1+xNu_U)/(1-2*xNu_U) - (1+xNu)/(1-2*xNu)
        Fac=2D0*G/3D0 * Fac
        BulkW = Fac
    End If
End If ! IDTask = 3, 6

If (IDTask .Eq. 4) Then ! Number of state parameters
    nStat = nStatV
End If ! IDTask = 4

If (IDTask .Eq. 5) Then ! matrix type
    NonSym = 0 ! 1 for non-symmetric D-matrix
    iStrsDep = 1 ! 1 for stress dependent D-matrix
    iTang = 0 ! 1 for tangent D-matrix
    iTimeDep = 0 ! 1 for time dependent D-matrix
End If ! IDTask = 5

```

```
Return  
End ! MyModel1
```

MYMODEL2.for

```

Subroutine MyModel2 ( IDTask, iMod, IsUndr,
**                    iStep, iTer, iEl, Int,
**                    X, Y, Z,
**                    Time0, dTime,
**                    Props, Sig0, Swp0, StVar0,
**                    dEps, D, BulkW,
**                    Sig, Swp, StVar, ipl,
**                    nStat,
**                    NonSym, iStrsDep, iTimeDep, iTang,
**                    iAbort )
!
! Purpose: User supplied soil model, HASP-E, explicit modified Euler with automatic
! subincrementation and error control
! Depending on IDTask, 1 : Initialize state variables
!                       2 : calculate stresses,
!                       3 : calculate material stiffness matrix
!                       4 : return number of state variables
!                       5 : inquire matrix properties
!                          return switch for non-symmetric D-matrix
!                          stress/time dependent matrix
!                       6 : calculate elastic material stiffness matrix
!
! Arguments:
!           I/O  Type
! IDTask   I    I    : see above
! iMod     I    I    : model number (1..10)
! IsUndr   I    I    : =1 for undrained, 0 otherwise
! iStep    I    I    : Global step number
! iter     I    I    : Global iteration number
! iel      I    I    : Global element number
! Int      I    I    : Global integration point number
! X        I    R    : X-Position of integration point
! Y        I    R    : Y-Position of integration point
! Z        I    R    : Z-Position of integration point
! Time0    I    R    : Time at start of step
! dTime    I    R    : Time increment
! Props    I    R( ) : List with model parameters
! Sig0     I    R( ) : Stresses at start of step
! Swp0     I    R    : Excess pore pressure start of step
! StVar0   I    R( ) : State variable at start of step
! dEps     I    R( ) : Strain increment
! D        I/O  R(, ) : Material stiffness matrix
! BulkW    I/O  R    : Bulkmodulus for water (undrained only)
! Sig      0    R( ) : Resulting stresses
! Swp      0    R    : Resulting excess pore pressure
! StVar    0    R( ) : Resulting values state variables
! ipl      0    I    : Plasticity indicator
! nStat    0    I    : Number of state variables
! NonSym   0    I    : Non-Symmetric D-matrix ?
! iStrsDep 0    I    : =1 for stress dependent D-matrix
! iTimeDep 0    I    : =1 for time dependent D-matrix
! iAbort   0    I    : =1 to force stopping of calculation
!
!       Implicit Double Precision (A-H, 0-Z)
!
!       Dimension Props(*), Sig0(*), StVar0(*), dEps(*), D(6,6),
**          Sig(*), StVar(*)
!
!---- Local variables (Arrays)
!

```



```

! AUXV11(6)      : Array which contains the product of Va and De (transpose(Va) * De)
(modEuler) (IDTask=2)
! AUXV12(6)      : Array which contains the product of Va and De (transpose(Va) * De)
(modEuler) (IDTask=2)
! AUXV21(6)      : Array which contains the product of De and Vb (De * Vb) (modEuler)
(IDTask=2)
! AUXV22(6)      : Array which contains the product of De and Vb (De * Vb) (modEuler)
(IDTask=2)
! AUXV3(6)       : Array which contains the product (transpose(Va0) * De) (IDTask=2)
! AUXV4(6)       : Array which contains the product (De * Vb0) (IDTask=2)
! BT(20)         : Array which contains whether the strings are slack or taut at the
end of step
! dSig(6)        : Array of stress increment (IDTask=2)
! dStresse1(6)   : Array of elastic stress increments (modEuler) (IDTask=2)
! dStresse2(6)   : Array of elastic stress increments (modEuler) (IDTask=2)
! dStrain(6)     : Array of strain increments (compression is positive) (IDTask=2)
! dStress1(6)    : Array of stress increments (compression is positive) (modEuler)
(IDTask=2)
! dStress2(6)    : Array of stress increments (compression is positive) (modEuler)
(IDTask=2)
! dTstrain       : Array of strain increments in one subincrement (IDTask=2)
! SLen(20)       : Array with length of strings (IDTask=1, IDTask=2)
! Straint(6)     : Array of strains at start of step (compression is positive)
(IDTask=2)
! Straintt(6)    : Array of strains at the end of step (compression is positive)
(IDTask=2)
! Stresst(6)     : Array of stresses at start of step (compression is positive)
(IDTask=2)
! Stresstt(6)    : Array of stresses at the end of step (compression is positive)
(IDTask=2)
! Stress0(6)     : Array of uncorrected stresses - drift correction (IDTask=2)
! Tstrain(6)     : Array of strains at start of subincrement (IDTask=2)
! Tstress(6)     : Array of stresses at start of subincrements (IDTask=2)
! Tstress1(6)    : Array of stresses at start of subincrements (modEuler) (IDTask=2)
! TTstrain(6)    : Array of strains at the end of subincrement (IDTask=2)
! TTstress(6)    : Array of stresset at the end of subincrement (IDTask=2)
! Va1(6)         : Array of yield function derivatives (modEuler) (IDTask=2)
! Va2(6)         : Array of yield function derivatives (modEuler) (IDTask=2)
! Va0(6)         : Array of yield function derivatives for uncorrected stresses -
drift correction (IDTask=2)
! Vb1(6)         : Array of plastic potential function derivatives (modEuler)
(IDTask=2)
! Vb2(6)         : Array of plastic potential function derivatives (modEuler)
(IDTask=2)
! Vb0(6)         : Array of plastic potential function derivatives - drift correction
(IDTask=2)
!
!
!--- Local variables (Matrices)
!
! De0(6,6)       : Elastic De matrix in drift correction algorithm (IDTask=2)
! De1(6,6)       : Elastic De matrix (modEuler) (IDTask=2)
! De2(6,6)       : Elastic De matrix (modEuler) (IDTask=2)
! SH(20,6)       : Matrix which contains the position of each brick (IDTask=2)
! SHt(20,6)      : Matrix which contains the position of each brick at the end of one
subincrement (trial substep) (IDTask=2)
!
!--- Local variables
!
! AUX11          : Auxilliary real variable, transpose(Va) * dStresse (modEuler)
(IDTask=2)

```

```

! AUX12      : Auxilliary real variable, transpose(Va) * dStresse (modEuler)
(IDTask=2)
! AUX21      : Auxilliary real variable, transpose(Va) * De * Vb (modEuler)
(IDTask=2)
! AUX22      : Auxilliary real variable, transpose(Va) * De * Vb (modEuler)
(IDTask=2)
! AUX3       : Auxilliary real variable, transpose(Va0) * De * Vb0 (IDTask=2)
! AUX4       : Auxilliary real variable, transpose(Va0) * Va0
! Bulkd      : Bulk modulus in drift correction algorithm (IDTask=2)
! dBG        : Proportion of shear stiffness belonging to one brick (IDTask=1,
IDTask=2)
! Bulk       : Bulk modulus at start of step (IDTask=2, IDTask=4, IDTask=6)
! BulkW      : Bulk modulus of water (IDTask=4, IDTask=6)
! dEpsV      : Increment of volumetric strain (IDTask=2)
! dlam       : Plastic multiplier (IDTask=2)
! dlam1      : Plastic multiplier (modEuler) (IDTask=2)
! dlam2      : Plastic multiplier (modEuler) (IDTask=2)
! dp01       : Increment of hardening parameter (modEuler) (IDTask=2)
! dp02       : Increment of hardening parameter (modEuler) (IDTask=2)
! dShearStrain : Increment of shear strain invariant (IDTask=2)
! dsubtime   : size of subincrement (IDTask=2)
! dSwp       : Increment of excess pore water pressure (IDTask=2)
! D1,D2 and D3 : Auxilliary variables for calculation od D matrix (IDTask=4,
IDTask=6)
! EM         : Machine Error tolerance (IDTask=2)
! EpsV       : Volumetric strain (IDTask=3, IDTask=6)
! Error      : Error estimation in automatic subincrementation (IDTask=2)
! Error1     : Error estimation in stresses determination in automatic
subincrementation (IDTask=2)
! Error2     : Error estimation in hardening parameter determination in automatic
subincrementation (IDTask=2)
! et         : Void ratio at start of step (IDTask=2, IDTask=4, IDTask=6)
! eta        : Stress ratio (IDTask=2)
! etai       : Initial stress ratio (IDTask=1)
! etamax     : Max previous stress ratio (IDTask=1)
! etat       : Shear ratio at start of step (IDTask=2)
! e0         : Initial void ratio (IDTask=1, IDTask=2, IDTask=4, IDTask=6)
! ftol       : Tolerance for zero value of yield function (IDTask=2)
! f0         : Value of yield function that must be corrected - drift correction
(IDTask=2)
! G          : Shear modulus at start of step (IDTask=2, IDTask=4, IDTask=6)
! Gbrick     : Shear modulus when i brick strings are taut (IDTask=1)
! Gur        : Shear modulus for URL at pref=100 kPa (IDTask=1, IDTask=2)
! G0         : Small strain shear modulus at pref=100 (IDTask=1, IDTask=2,
IDTask=4, IDTask=6)
! G1s       : Shear modulus at start of subincrement for p=100 kPa (IDTask=2)
! G2         : Shear modulus at the end of step for p=100 kPa (IDTask=2)
! Gd         : Shear modulus in drift correction algorithm (IDTask=2)
! Gama       : Specific volume for CLS at p'=1 kPa (IDTask=1, IDTask=2)
! iDrift     : Counter in drift correction loop (IDTask=2)
! isuc       : =0 if previous subincrement is successful, =1 if not (IDTask=2)
! iss        : Counts the number of successful subincrements (IDTask=2)
! its        : Counts the number of total subincrements (IDTask=2)
! Nsubs      : Number of subincrements (IDTask=2)
! nStatV     : Number of used state variables
! OCR        : Input value of overconsolidation ratio (IDTask=1)
! omega      : Hardening coefficient (IDTask=1, IDTask=2)
! omegalim   : Limiting value for hardening coefficient in order to avoid
numerical instability (IDTask=2)
! omega0     : Hardening coefficient in drift correction algorithm (IDTask=2)
! p          : mean effective stress (IDTask=2)
! pi         : Initial value of mean effective stress

```

```

! Pie          : Value of Pi (IDTask=1)
! pmax         : Max previous mean effective stress (IDTask=1)
! POM1, POM2, ...: Auxilliary real variables
! POP         : Input value of preoverburden pressure (IDTask=1)
! pt          : Mean effective stress at start of step (IDTask=2, IDTask=4,
IDTask=6)
! ptt         : mean effective stress at the end of step (IDTask=2)
! PSINC       : State parameter for NC soil (IDTask=1, IDTask=2)
! PSINC0      : State parameter for NC soil in drift correction algorithm
(IDTask=2)
! PSIOC       : State parameter for OC soil ((IDTask=1, IDTask=2)
! PSIOC0      : State parameter for OC soil in drift correction algorithm
(IDTask=2)
! p0          : Hardening parameter for yield surface (IDTask=1, IDTask=2)
! p0bs        : Hardening parameter for bounding surface (IDTask=1, IDTask=2)
! p0t         : Hardening parameter for current yield surface at start of step
(IDTask=2)
! p0tt        : Hardening parameter for current yield surface at start of step
(IDTask=2)
! q           : Shear deviator stress (IDTask=2)
! qi          : Initial value of shear deviator stress (IDTask=1)
! qmax        : Max previous shear deviator stress (IDTask=1)
! qt          : Shear deviator stress at start of step (IDTask=2)
! qtt         : Shear deviator stress at the end of step (IDTask=2)
! R           : Isotropic OCR ratio (IDTask=1, IDTask=2)
! Rmax        : Max value of R in order to avoid numerical instability (IDTask=1,
IDTask=2)
! Rmin        : Min value of R in order to avoid non-possible values (IDTask=1,
IDTask=2)
! R0          : Isotropic OCR ratio in drift correction algorithm (IDTask=2)
! strain70    : Reference shear strain (IDTask=1)
! stepsizem  : Relative size of next subincrement size to the previous
subincrement size (IDTask=2)
! STOL        : Error tolerance for evaluation of subincrement size (IDTask=2)
! subtime1    : Time at the begining of subincrement
! subtime2    : Time at the end of subincrement
! Sxmax       : Max previous horizontal stress in x direction (IDTask=1)
! Symax       : Max previous vertical stress (IDTask=1)
! Szmax       : Max previous horizontal stress in y direction (IDTask=1)
! TBulk1      : Bulk modulus at the start of subincrement (modEuler) (IDTask=2)
! TBulk2      : Bulk modulus at the start of subincrement (modEuler) (IDTask=2)
! Te          : Void ratio at start of subincrement
! Teta        : Stress ratio at start of increment (IDTask=2)
! theta       : Lodes angle (IDTask=2)
! thetat      : Lodes angle at start of step (IDTask=2)
! TG1         : Shear modulus at start of subincrement (modEuler) (IDTask=2)
! TG2         : Shear modulus at start of subincrement (modEuler) (IDTask=2)
! Tomega1     : hardening coefficient at start of step (modEuler) (IDTask=2)
! Tomega2     : hardening coefficient at start of step (modEuler) (IDTask=2)
! Tp          : Mean effective stress at start of subincrement (IDTask=2)
! TPSINC1     : State parameter for NC soil at start of subincrement (modEuler)
(IDTask=2)
! TPSINC2     : State parameter for NC soil at start of subincrement (modEuler)
(IDTask=2)
! TPSIOC1     : State parameter for OC soil at start of subincrement (modEuler)
(IDTask=2)
! TPSIOC2     : State parameter for OC soil at start of subincrement (modEuler)
(IDTask=2)
! Tp0         : Hardening parameter at start of subincrement (IDTask=2)
! Tp01        : Hardening parameter at start of subincrement (modEuler) (IDTask=2)
! Tp1         : Mean effective stress (modEuler) (IDTask=2)
! Tq          : Deviator stress at start of subincrement (IDTask=2)

```

```

! Tq1          : Deviator stress at start of subincrement (modEuler) (IDTask=2)
! TR1          : Isotropic OCR ratio at start of subincrement (modEuler) (IDTask=2)
! TR2          : Isotropic OCR ratio at start of subincrement (modEuler) (IDTask=2)
! TTe         : Void ratio at the end of subincrement (IDTask=2)
! Teta        : Stress ratio at the end of subincrement (IDTask=2)
! Teta1       : Stress ratio at the end of subincrement (modEuler) (IDTask=2)
! TTeta       : Stress ratio at the end of subincrement (IDTask=2)
! TTf         : Yield function at the end of subincrement (IDTask=2)
! Ttheta      : Lodes angle at start of subincrement (IDTask=2)
! Ttheta1     : Lodes angle at start of subincrement (modEuler) (IDTask=2)
! TTp        : Mean effective stress at start of subincrement (IDTask=2)
! TTp0       : Hardening parameter at the end of subincrement (IDTask=2)
! TTq        : Deviator stress at the end of subincrement (IDTask=2)
! TTtheta     : Lodes angle at the end of subincrement (IDTask=2)
! VarA0      : Variable A in stress integration procedure (drift correction)
(IDTask=2)
! VarA1      : Variable A in stress integration procedure (modEuler) (IDTask=2)
! VarA2      : Variable A in stress integration procedure (modEuler) (IDTask=2)
! VarB0      : Variable B in stress integration procedure (drift correction)
(IDTask=2)
! VarB1      : Variable B in stress integration procedure (modEuler) (IDTask=2)
! VarB2      : Variable B in stress integration procedure (modEuler) (IDTask=2)
! xk         : kappa, slope of U/R line (IDTask=1, IDTask=2, IDTask=4, IDTask=6)
! xka        : small strain kappa (IDTask=1, IDTask=2)
! xkd        : kappa for drift correction (IDTask=2)
! xk1, xk2   : kappa in (modEuler) (IDTask=1, IDTask=2)
! xKNC       : Ratio of horizontal and vertical stresses at NC state (IDTask=1)
! xl         : Lambda, slope of ICL (IDTask=1, IDTask=2, IDTask=4, IDTask=6))
! xM         : M, slope of CSL in p-q plane (IDTask=1, IDTask=2)
! xMc        : slope of CSL in p-q plane in TXC (IDTask=1, IDTask=2)
! xMe        : slope of CSL in p-q plane in TXE (IDTask=1, IDTask=2)
! xMs        : slope of CSL in p-q plane at the start of subincrement (IDTask=2)
! xMs1       : slope of CSL in p-q plane for dStress2 calculation (modEuler)
(IDTask=2)
! xNu        : Poissons ratio (IDTask=1, IDTask=2, IDTask=4, IDTask=6)
! xX,yY,zZ   : Parameters that describe the dependency of M on Lode's angle
(IDTask=1, IDTask=2)
!
!--- Auxilliary Subroutines
!
! CalculateStressInvariants : Calculates p,q,eta and theta
! CalculateVoidRatio       : Calculates current void ratio
! CalculateElasticModuli   : Calculates bulk and shear modulus at start of step
! CalculateOmega           : Calculates hardening coefficient
! CalculateVectorA         : Calculates vector A which contains derivatives of
yield function (not dependent on Lode's angle)
! CalculateVectorA2        : Calculates vector A which contains derivatives of
yield function (dependent on Lode's angle)
! CalculateDeMatrix        : Calculates elastic D matrix
! MZEROR                   : Fills real array with zeros
! MatVec                   : Multiplies matrix with a vector. Result is vector
! DInProd                  : Inner product of two vectors. Result is scalar
! AddVec                   : Calculates sum of two vectors. result is vector
! COPYRVEC                 : Copies a Double array R1 with Dimension K to R2
! CalculateElasticModuliBrick: Calculates shear modulus for Brick model
!
! Dimension Stresst(6), Straint(6), dStrain(6), Straintt(6),
* dStress1(6), Stresstt(6), dSig(6), Va1(6), De1(6,6),
* dStresse1(6), AUXV11(6), AUXV21(6), Tstress(6),
* Tstrain(6), dTstrain(6), TTstress(6), TTstrain(6),
* Stress0(6), Va0(6), AUXV3(6), AUXV4(6), De0(6,6),
* Tstress1(6), Va2(6), dStresse2(6), De2(6,6),

```

```

*      AUXV12(6), AUXV22(6), dStress2(6), dStress(6), SLen(20),
*      SH(20,6), SHs(20,6), BT(20), BTs(20), SHt(20,6), Vb0(6),
*      Vb1(6), Vb2(6)
Parameter (ftol=1.D-9, Htol=1.D-6, EM=1.D-16,
*omegalim=15000, Rmax=100.0, Rmin=1.0,
*Pie=3.14159 26535 89793 23846 26433 83279 50288 41971 69399d0)

```

```

! Expected contents of Props(1..?)
! 1 : lambda   slope of ICL
! 2 : kappa    slope of U/R line
! 3 : Mc       slope of CSL in p-q plane in TXC
! 4 : Me       slope of CSL in p-q plane in TXE
! 5 : nu       Poisson ratio
! 6 : evoid0   Initial void ration
! 7 : OCR      overconsolidation ratio
! 8 : POP      preoverburden pressure
! 9 : K0NC     ratio of horizontal and vertical stresses
! 10 : STOL    error tolerance in subincrementation algorithm
! 11 : Go,ref  small strain shear modulus at pref=100 kPa
! 12 : gama0,7 reference strain

```

```
nStatV = 180
```

```

If (IDTask .Eq. 1) Then ! Initialize state variables
  if (StVar0(1).eq.0) then !Reset state variables
    ! Get Props
    xl=Props(1)
    xk=Props(2)
    xMc=Props(3)
    xMe=Props(4)
    if (xMe.eq.0.0) xMe=xMc
    xNu=Props(5)
    e0=Props(6)
    OCR=Props(7)
    POP=Props(8)
    xKNC=Props(9)
    G0=Props(11)
    strain70=Props(12)
    If (G0.gt.0.0.and.strain70.eq.0.0) Then
      Call OK_MessageBox(' Invalid value for "gama0.7" (param 12)')
* )
    iAbort = 1
    Return
  End If
  ! Calculate Gama from OCR, POP
  Symax=-OCR*Sig0(2)+POP
  Sxmax=xKNC*Symax
  Szmax=xKNC*Symax
  pmax=(Symax+Sxmax+Szmax)/3
  qmax=abs(Symax-Sxmax)
  etamax=qmax/pmax
  pi=-((Sig0(1)+Sig0(2)+Sig0(3))/3.0
  qi=sqrt(3.0*(1.0/6.0*((Sig0(1)-Sig0(2))**2.0+
*(Sig0(2)-Sig0(3))**2.0+(Sig0(3)-Sig0(1))**2.0)+Sig0(4)**2.0+
*Sig0(5)**2.0+Sig0(6)**2.0))
  etai=qi/pi
  xM=xMc
  POM1=(pmax*(etamax**2.0+xM**2.0)/2.0/xM**2.0)**((xl-xk)/xl)
  POM2=pi**((xk/xl)
  Gama=1.0+e0+xl*log(POM1*POM2)
  PSIOC=1.0+e0+xl*log(pi)-Gama
  PSINC=(xl-xk)*log(2.0*xM**2.0/(xM**2.0+etai**2.0))
  R=exp((PSINC-PSIOC)/(xl-xk))

```

```

If (R.gt.Rmax) then
    Gama=1.0+e0+xl*log(pi)-PSINC+(xl-xk)*log(Rmax)
    R=Rmax
end if
if (R.lt.Rmin) then
    Gama=1.0+e0+xl*log(pi)-PSINC+(xl-xk)*log(Rmin)
    R=Rmin
end if
omega=(1.0+(PSINC-PSIOC)/PSINC)*R
p0=qi**2.0/(pi*xM**2.0)+pi
p0bs=R*p0
! Check if value of Poissons ratio is ok
If (xNu.Gt.0.5d0-1d-6 .Or. xNu.Lt.-1d-6) Then
    Call OK_MessageBox(' Invalid value for "nu" (param 5)')
    iAbort = 1
    Return
End If
if (xNu.Lt.0) then
    xNu = 0.0
    Props(5) = xNu
end if
if (xNu.Gt.0.495d0) then
    xNu = 0.495
    Props(5) = xNu
end if
! Initialize Brick model
Gur=3.0*(1.0-2.0*xNu)/(2.0*(1.0+xNu))*(1.0+e0)/xk*100.0
if (G0.lt.Gur) G0=Gur
dBG=(G0-Gur)/(20*G0)
do i=1,20
    Gbrick=(1.0-(real(i)-0.5)*dBG)*G0
    SLen(i)=strain70/0.385*(sqrt(G0/Gbrick)-1.0)
end do
G=G0*pi/100.0
! Variables X, Y and Z for determination of parameter M
zZ=-0.229
xX=((xMc**(1.0/zZ)+xMe**(1.0/zZ))/2.0)**zZ
yY=(1.0-(xMc/xMe)**(1.0/zZ))/(1.0+(xMc/xMe)**(1.0/zZ))
! Fill state variable array
do i=1,nStatV
    StVar0(i)=0.0
end do
StVar0(1)=1.0
StVar0(2)=Gama
StVar0(3)=R
StVar0(4)=PSIOC
StVar0(5)=PSINC
StVar0(6)=omega
StVar0(7)=p0
StVar0(8)=p0bs
StVar0(9)=Symax
StVar0(10)=Symax/(-Sig0(2))
StVar0(11)=2.0
StVar0(12)=G0
StVar0(13)=Gur
StVar0(14)=G
StVar0(15)=dBG
StVar0(16)=-30.0
StVar0(17)=xX
StVar0(18)=yY
StVar0(19)=xM
do i=1,20

```



```

        StVar0(20+i)=SLen(i)
    end do
end if
If (iel+int.eq.2) Call WriVec(1, 'Props', Props, 12)
End If ! IDTask = 1
If (IDTask .Eq. 2) Then ! Calculate stresses
    If (IsUndr.Eq.1) Then
        dEpsV = dEps(1) + dEps(2) + dEps(3)
        dSwp = BulkW * dEpsV
        Swp = Swp0 + dSwp
    Else
        Swp = Swp0
    End If
    If (iel+Int+iter.Eq.3 .And. iStep.Lt.10) Then
        Call WriMat( -1, 'D66', D, 6, 6, 6 )
    End If
! Material properties
xl=Props(1)
xk=Props(2)
xNu=Props(5)
e0=Props(6)
G0=Props(11)
! State variables
Gama=StVar0(2)
G1s=StVar0(12)
Gur=StVar0(13)
dBG=StVar0(15)
xX=StVar0(17)
yY=StVar0(18)
zZ=-0.229
do i=1,20
    SLen(i)=StVar0(20+i)
end do
do i=1,20
    do j=1,6
        SH(i,j)=StVar0(20+20*j+i)
    end do
end do
! Reversing sign for input stresses and strains
do i=1,6
    Stresst(i)=-Sig0(i)
    Straint(i)=-dEps(i+6)
    dStrain(i)=-dEps(i)
    Straintt(i)=Straint(i)+dStrain(i)
end do
!Stress and strain invariants
Call CalculateStressInvariants(Stresst,pt,qt,etat,thetat)
Call CalculateVoidRatio(Straint,e0,et)
!Calculate value of parameter M at the start of step
xM=xX*(1.0+yY*sin(3.0*thetat))**zZ
if (isnan(xM)) xM=xX*(1.0-yY)**zZ
!Material modules
if (G0.eq.0.0) Call CalculateElasticModuli(et,pt,xk,xNu,Bulk,G)
!Hardening parameter for curent yield surface
p0t=qt**2.0/pt/xM**2.0+pt
!Calculate OMEGA - hardening coefficient at the beginning of increment
Call CalculateOmega(xl,xk,xM,Gama,pt,qt,et,omegalim,Rmax,Rmin,
*omega,R,PSIOC,PSINC)
! Stress determination explicit subincrementation modified euler with automatic
determination of increment size
!
! Initialization

```

```

STOL=Props(10)
if (STOL.lt.1.D-9) STOL=1.D-9
ipl=4
isuc=0
its=0
iss=0
subtime1=0.0
dsubtime=1.0
Call MZEROR(Tstress,6)
Call MZEROR(Tstrain,6)
Call MZEROR(dTstrain,6)
Call MZEROR(TTstress,6)
Call MZEROR(TTstrain,6)
Call COPYRVEC(Stresst,Tstress,6)
Call COPYRVEC(Straint,Tstrain,6)
do i=1,6
    dTstrain(i)=dStrain(i)*dsubtime
end do
Call AddVec(Tstrain,dTstrain,1.0,1.0,6,TTstrain)
! Stress calculation
110 continue
its=its+1
    ! Calculation of dStress1 and dp01
    Call CalculateStressInvariants(Tstress, Tp, Tq, Teta, Ttheta)
    xMs=xX*(1.0+yY*sin(3.0*Ttheta))*zZ
    if (isnan(xMs)) xMs=xX*(1.0-yY)*zZ
    xMs1=xMs
    Call CalculateVoidRatio(Tstrain, e0, Te)
    Call CalculateVoidRatio(TTstrain, e0, TTe)
    if (G0.eq.0.0) then
        Call CalculateElasticModuli(Te, Tp, xk, xNu, TBulk1, TG1)
        xk1=xk
    else
        do i=1,20
            do j=1,6
                SHt(i,j)=SH(i,j)
            end do
        end do
        Call CalculateElasticModuliBrick(BT, TTstrain, SHt, SLen,
*G1s, G2s, Gur, G0, TG1, Te, Tp, xNu, xk1, dBG, TBulk1)
    end if
    Tp0=Tq**2.0/Tp/xMs**2.0+Tp
    Call CalculateOmega(xl, xk, xMs1, Gama, Tp, Tq, Te, omegalim, Rmax,
*Rmin, Tomega1, TR1, TPSIOC1, TPSINC1)
    Call MZEROR(Va1,6)
    Call CalculateVectorA2(Tp, Tp0, Tq, Ttheta, xMs1, Tstress, xX, yY,
*zZ, Va1)
    Call MZEROR(Vb1,6)
    Call CalculateVectorA(Tp, Tp0, xMs1, Tstress, Vb1)
    Call MZEROR(De1,36)
    Call CalculateDeMatrix(TBulk1, TG1, De1)
    VarA1=(1.0+Te)*Tp0*Tomega1*Tp*(2.0*Tp-Tp0)/(xl-xk1)
    VarB1=(1.0+Te)*Tp0*Tomega1*(2.0*Tp-Tp0)/(xl-xk1)
    Call MZEROR(dStresse1,6)
    Call MatVec(De1,6,dTstrain,6,dStresse1)
    AUX11=0.0
    AUX11=DInProd(Va1,dStresse1,6)
    Call MZEROR(AUXV11,6)
    Call MatVec(De1,6,Va1,6,AUXV11)
    AUX21=0.0
    AUX21=DInProd(AUXV11,Vb1,6)
    dlam1=AUX11/(VarA1+AUX21)

```



```

if (isnan(dlam1).or.dlam1.lt.0.0) dlam1=0.0
Call MZEROR(AUXV21,6)
Call MatVec(De1,6,Vb1,6,AUXV21)
Call MZEROR(dStress1,6)
Call AddVec(dStresse1,AUXV21,1.0,-dlam1,6,dStress1)
Call MZEROR(Tstress1,6)
Call AddVec(TStress,dStress1,1.0,1.0,6,Tstress1)
dp01=dlam1*VarB1
Tp01=Tp0+dp01
! Calculation of dStress2 and dp02
Call CalculateStressInvariants(Tstress1,Tp1,Tq1,Teta1,
*Ttheta1)
xMs1=xX*(1.0+yY*sin(3.0*Ttheta1))*zZ
if (isnan(xMs1)) xMs1=xX*(1.0-yY)*zZ
if (G0.eq.0.0) then
    Call CalculateElasticModuli(TTe,Tp1,xk,xNu,TBulk2,TG2)
    xk2=xk
else
    TG2=TG1*Tp1/Tp
    TBulk2=2.0*TG2*(1.0+xNu)/(3.0*(1.0-2.0*xNu))
    xk2=(1+TTe)*Tp1*1.0/TG2*3.0*(1.0-2.0*xNu)/(2.0*(1.0+xNu)
*)
end if
Call CalculateOmega(xl,xk,xMs1,Gama,Tp1,Tq1,TTe,omegalim,
*Rmax,Rmin,Tomega2,TR2,TPSIOC2,TPSINC2)
Call MZEROR(Va2,6)
Call CalculateVectorA2(Tp1,Tp01,Tq1,Ttheta1,xMs1,Tstress1,
*xX,yY,zZ,Va2)
Call MZEROR(Vb2,6)
Call CalculateVectorA(Tp1,Tp01,xMs1,Tstress1,Vb2)
Call MZEROR(De2,36)
Call CalculateDeMatrix(TBulk2,TG2,De2)
VarA2=(1.0+TTe)*Tp01*Tomega2*Tp1*(2.0*Tp1-Tp01)/(xl-xk2)
VarB2=(1.0+TTe)*Tp01*Tomega2*(2.0*Tp1-Tp01)/(xl-xk2)
Call MZEROR(dStresse2,6)
Call MatVec(De2,6,dTstrain,6,dStresse2)
AUX12=0.0
AUX12=DInProd(Va2,dStresse2,6)
Call MZEROR(AUXV12,6)
Call MatVec(De2,6,Va2,6,AUXV12)
AUX22=0.0
AUX22=DInProd(AUXV12,Vb2,6)
dlam2=AUX12/(VarA2+AUX22)
if (isnan(dlam2).or.dlam2.lt.0.0) dlam2=0.0
Call MZEROR(AUXV22,6)
Call MatVec(De2,6,Vb2,6,AUXV22)
Call MZEROR(dStress2,6)
Call AddVec(dStresse2,AUXV22,1.0,-dlam2,6,dStress2)
dp02=dlam2*VarB2
! Stresses and hardening parameter for modified Euler
Call MZEROR(dStress,6)
Call AddVec(dStress1,dStress2,0.5,0.5,6,dStress)
Call MZEROR(TTstress,6)
Call AddVec(Tstress,dStress,1.0,1.0,6,TTstress)
TTp0=Tp0+(dp01+dp02)/2.0
! Error estimation
Error1=sqrt((dStress2(1)-dStress1(1))*2.0+(dStress2(2)-
*dStress1(2))*2.0+(dStress2(3)-dStress1(3))*2.0+(dStress2(4)-
*dStress1(4))*2.0+(dStress2(5)-dStress1(5))*2.0+(dStress2(6)-
*dStress1(6))*2.0)/(2.0*sqrt(TTstress(1)**2.0+TTstress(2)**2.0+
*TTstress(2)**2.0+TTstress(3)**2.0+TTstress(4)**2.0+TTstress(6)**
*2.0))

```

```

Error2=abs(dp02-dp01)/(2.0*TTp0)
Error=Error1
if (Error2.gt.Error) Error=Error2
if (EM.gt.Error) Error=EM
if (Error.gt.STOL.and.dsubtime.gt.0.000001) then
  stepsize=0.9*sqrt(STOL/Error)
  if (stepsize.lt.0.1) stepsize=0.1
  dsubtime=stepsize*dsubtime
  if (dsubtime.lt.0.000001) dsubtime=0.000001
  Call MZEROR(dTstrain,6)
  do i=1,6
    dTstrain(i)=dStrain(i)*dsubtime
  end do
  Call MZEROR(TTstrain,6)
  Call AddVec(Tstrain,dTstrain,1.0,1.0,6,TTstrain)
  isuc=1
  goto 110
end if
Call CalculateStressInvariants(TTstress,TTp,TTq,TTeta,
*TTtheta)
TTf=TTq**2.0/xMs**2.0+TTp*(TTp-TTp0)
if (dlam1.ne.0.0.and.dlam2.ne.0.0.and.abs(TTf).gt.ftol) then
  do iDrift=1,10
    Call MZEROR(Stress0,6)
    Call COPYRVEC(TTstress,Stress0,6)
    Call CalculateStressInvariants(Stress0,p,q,eta,
*theta)
    p0=TTp0
    f0=TTf
    if (G0.eq.0.0) then
      Call CalculateElasticModuli(TTe,p,xk,xNu,Bulkd,
*Gd)
      xkd=xk
    else
      Gd=TG1*p/Tp
      Bulkd=2.0*Gd*(1.0+xNu)/(3.0*(1.0-2.0*xNu))
      xkd=(1+TTe)*p*1.0/Gd*3.0*(1.0-2.0*xNu)/(2.0*(1.0
*xNu))
    end if
    Call CalculateDeMatrix(Bulkd,Gd,De0)
    Call CalculateOmega(xl,xk,xMs,Gama,p,q,TTe,omegalim,
*Rmax,Rmin,omega0,R0,PSIOC0,PSINC0)
    Call MZEROR(Va0,6)
    Call CalculateVectorA2(p,p0,q,theta,xMs,Stress0,xX,
*yY,zZ,Va0)
    Call MZEROR(Vb0,6)
    Call CalculateVectorA(p,p0,xMs,Stress0,Vb0)
    VarA0=(1.0+TTe)*p0*omega0*p*(2.0*p-p0)/(xl-xkd)
    VarB0=(1.0+TTe)*p0*omega0*(2.0*p-p0)/(xl-xkd)
    Call MZEROR(AUXV3,6)
    Call MatVec(De0,6,Va0,6,AUXV3)
    AUX3=DInProd(AUXV3,Vb0,6)
    dlam=f0/(VarA0+AUX3)
    Call MZEROR(AUXV4,6)
    Call MatVec(De0,6,Vb0,6,AUXV4)
    Call MZEROR(TTstress,6)
    Call AddVec(Stress0,AUXV4,1.0,-dlam,6,TTstress)
    TTp0=p0+dlam*VarB0
    Call CalculateStressInvariants(TTstress,TTp,TTq,
*TTeta,TTtheta)
    TTf=TTq**2.0/xMs**2.0+TTp*(TTp-TTp0)
    If (abs(TTf).gt.abs(f0)) then

```

```

        AUX4=DInProd(Va0,Va0,6)
        dlam=f0/AUX4
        Call MZEROR(TTstress,6)
        Call AddVec(Stress0,Va0,1.0,-dlam,6,TTstress)
        TTp0=p0
        Call CalculateStressInvariants(TTstress,TTp,TTq,
*TTetatt,TTtheta)
        TTf=TTq**2.0/xMs**2.0+TTp*(TTp-TTp0)
        end if
        if (abs(TTf).lt.ftol) exit
    end do
end if
iss=iss+1
subtime2=subtime1+dsubtime
stepsize=0.9*sqrt(STOL/Error)
if (stepsize.gt.1.1) stepsize=1.1
if (isuc.eq.1) then
    if (stepsize.gt.1.0) stepsize=1.0
end if
dsubtime=stepsize*dsubtime
if (dsubtime.lt.0.000001) dsubtime=0.000001
if ((subtime2+dsubtime).gt.1.0) dsubtime=1.0-subtime2
isuc=0
Call MZEROR(Tstress,6)
Call COPYRVEC(TTstress,Tstress,6)
Call MZEROR(Tstrain,6)
Call COPYRVEC(TTstrain,Tstrain,6)
Call MZEROR(dTstrain,6)
do i=1,6
    dTstrain(i)=dStrain(i)*dsubtime
end do
Call MZEROR(TTstrain,6)
Call AddVec(Tstrain,dTstrain,1.0,1.0,6,TTStrain)
subtime1=subtime2
if (G0.ne.0.0) then
    G1s=G2s
    do i=1,20
        do j=1,6
            SH(i,j)=SHt(i,j)
        end do
    end do
end if
if (subtime2.lt.1.0) goto 110

! Final stresses and hardening parameter
Call MZEROR(Stresstt,6)
Call COPYRVEC(TTstress,Stresstt,6)
p0tt=TTp0
!
!Return stresses
do i=1,6
    Sig(i)=-Stresstt(i)
end do
!Update state variables
StVar(1)=StVar0(1)
StVar(2)=StVar0(2)
StVar(3)=R
StVar(4)=PSIOC
StVar(5)=PSINC
StVar(6)=omega
StVar(7)=p0t
StVar(8)=p0t*R

```

```

If (StVar0(9).lt.Stresstt(2)) then
    StVar(9)=Stresstt(2)
else
    StVar(9)=StVar0(9)
end if
StVar(10)=StVar(9)/Stresstt(2)
StVar(11)=2.0
StVar(16)=thetat*180.0/Pie
StVar(17)=StVar0(17)
StVar(18)=StVar0(18)
StVar(19)=xM
StVar(20)=Error
if (G0.eq.0.0) then
    StVar(12)=Gur
    StVar(13)=StVar0(13)
    StVar(14)=G
    StVar(15)=StVar0(15)
else
    StVar(12)=G2s
    StVar(13)=StVar0(13)
    StVar(14)=TG1
    StVar(15)=StVar0(15)
    do i=1,20
        StVar(20+i)=StVar0(20+i)
    end do
    do i=1,20
        do j=1,6
            StVar(20+20*j+i)=SH(i,j)
        end do
    end do
    do i=1,20
        StVar(160+i)=BT(i)
    end do
end if

End If ! IDTask = 2

If ( IDTask .Eq. 3 .Or.
* IDTask .Eq. 6 ) Then ! Calculate D-Matrix
    xk=Props(2)
    xNu=Props(5)
    e0=Props(6)
    G0=Props(11)
    pt=-(Sig0(1)+Sig0(2)+Sig0(3))/3.0
    if (G0.eq.0.0) then
        EpsV=-(dEps(7)+dEps(8)+dEps(9))
        et=(1.0+e0)*exp(-EpsV)-1.0
        Call CalculateElasticModuli(et,pt,xk,xNu,Bulk,G)
    else
        G=G0*pt/100.0
        if (G.lt.1.0) G=1.0
        Bulk=2.0*(1.0+xNu)/(3.0*(1.0-2.0*xNu))*G
    end if
    D1=Bulk+4.0*G/3.0
    D2=Bulk-2.0*G/3.0
    D3=G
    Call MZeroR(D,36)
    D(1,1)=D1
    D(2,2)=D1
    D(3,3)=D1
    D(1,2)=D2

```

```

D(1,3)=D2
D(2,1)=D2
D(2,3)=D2
D(3,1)=D2
D(3,2)=D2
D(4,4)=D3
D(5,5)=D3
D(6,6)=D3
BulkW = 0
If (IsUndr.Eq.1) Then
  ! BulkW = ...
  xNu_U = 0.495d0
  Fac=(1+xNu_U)/(1-2*xNu_U) - (1+xNu)/(1-2*xNu)
  Fac=2D0*G/3D0 * Fac
  BulkW = Fac
End If
End If ! IDTask = 3, 6

If (IDTask .Eq. 4) Then ! Number of state parameters
  nStat = nStatV
End If ! IDTask = 4

If (IDTask .Eq. 5) Then ! matrix type
  NonSym = 0 ! 1 for non-symmetric D-matrix
  iStrsDep = 1 ! 1 for stress dependent D-matrix
  iTang = 0 ! 1 for tangent D-matrix
  iTimeDep = 0 ! 1 for time dependent D-matrix
End If ! IDTask = 5

Return
End ! MyModel2

```



```

! AUXV1X(6)      : Array which contains the product of Va and De (transpose(Va) * De)
(RKDP) (X=1,2,3,4,5,6) (IDTask=2)
! AUXV2X(6)      : Array which contains the product of De and Vb (De * Vb) (RKDP)
(X=1,2,3,4,5,6) (IDTask=2)
! AUXV3(6)       : Array which contains the product (transpose(Va0) * De) (IDTask=2)
! AUXV4(6)       : Array which contains the product (De * Vb0) (IDTask=2)
! BT(20)         : Array which contains whether the strings are slack or taut at the
end of step
! dStresseX(6)   : Array of elastic stress increments (RKDP) (X=1,2,3,4,5,6)
(IDTask=2)
! dStrain(6)     : Array of strain increments (compression is positive) (IDTask=2)
! dStressX(6)    : Array of stress increments (compression is positive)
(X=1,2,3,4,5,6) (modEuler) (IDTask=2)
! dTstrain      : Array of strain increments in one subincrement (IDTask=2)
! SLen(20)      : Array with length of strings (IDTask=1, IDTask=2)
! Straint(6)    : Array of strains at start of step (compression is positive)
(IDTask=2)
! Straintt(6)   : Array of strains at the end of step (compression is positive)
(IDTask=2)
! Stresst(6)    : Array of stresses at start of step (compression is positive)
(IDTask=2)
! Stresstt(6)   : Array of stresses at the end of step (compression is positive)
(IDTask=2)
! Stress0(6)    : Array of uncorrected stresses - drift correction (IDTask=2)
! Tstrain(6)    : Array of strains at start of subincrement (IDTask=2)
! Tstress(6)    : Array of stresses at start of subincrements (IDTask=2)
! StressX(6)    : Array of stresses (X=1,2,3,4,5,6) (RKDP) (IDTask=2)
! TTstrain(6)   : Array of strains at the end of subincrement (IDTask=2)
! TTstress(6)   : Array of stresset at the end of subincrement (IDTask=2)
! VaX(6)        : Array of yield function derivatives (RKDP) (X=1,2,3,4,5,6)
(IDTask=2)
! Va0(6)        : Array of yield function derivatives for uncorrected stresses -
drift correction (IDTask=2)
! VbX(6)        : Array of plastic potential function derivatives (RKDP)
(X=1,2,3,4,5,6) (IDTask=2)
! Vb0(6)        : Array of plastic potential function derivatives - drift correction
(IDTask=2)
!
!
!--- Local variables (Matrices)
!
! De0(6,6)      : Elastic De matrix in drift correction algorithm (IDTask=2)
! DeX(6,6)      : Elastic De matrix (RKDP) (X=1,2,3,4,5,6) (IDTask=2)
! SH(20,6)      : Matrix which contains the position of each brick (IDTask=2)
! SHt(20,6)     : Matrix which contains the position of each brick at the end of one
subincrement (trial substep) (IDTask=2)
!
!--- Local variables
!
! AUX1X         : Auxilliary real variable, transpose(Va) * dStresse (RKDP)
(X=1,2,3,4,5,6) (IDTask=2)
! AUX2X         : Auxilliary real variable, transpose(Va) * De * Vb (RKDP)
(X=1,2,3,4,5,6) (IDTask=2)
! AUX3         : Auxilliary real variable, transpose(Va0) * De * Vb0 (IDTask=2)
! AUX4         : Auxilliary real variable, transpose(Va0) * Va0
! Bulk         : Bulk modulus at start of step (IDTask=2, IDTask=4, IDTask=6)
! Bulkd        : Bulk modulus in drift correction algorithm (IDTask=2)
! BulkX        : Bulk modulus in RKDP algorithm (IDTask=2)
! BulkW        : Bulk modulus of water (IDTask=4, IDTask=6)
! dBG         : Proportion of shear stiffness belonging to one brick (IDTask=1,
IDTask=2)
! dEv          : Increment of volumetric strain during one subincrement

```

```

! dEpsV      : Increment of volumetric strain (IDTask=2)
! dlam       : Plastic multiplier (IDTask=2)
! dlamX      : Plastic multiplier (RKDP) (X=1,2,3,4,5,6) (IDTask=2)
! dp0X       : Increment of hardening parameter (RKDP) (X=1,2,3,4,5,6) (IDTask=2)
! dShearStrain : Increment of shear strain invariant (IDTask=2)
! dsubtime   : size of subincrement (IDTask=2)
! dSwp       : Increment of excess pore water pressure (IDTask=2)
! D1,D2 and D3 : Auxilliary variables for calculation od D matrix (IDTask=4,
IDTask=6)
! EM         : Machine Error tolerance (IDTask=2)
! EpsV       : Volumetric strain (IDTask=3, IDTask=6)
! Error      : Error estimation in automatic subincrementation (IDTask=2)
! Error1     : Error estimation in stresses determination in automatic
subincrementation (IDTask=2)
! Error2     : Error estimation in hardening parameter determination in automatic
subincrementation (IDTask=2)
! et         : Void ratio at start of step (IDTask=2, IDTask=4, IDTask=6)
! eta       : Stress ratio (IDTask=2)
! etai      : Initial stress ratio (IDTask=1)
! etamax    : Max previous stress ratio (IDTask=1)
! etat      : Shear ratio at start of step (IDTask=2)
! etaX      : Shear ratio in RKDP algorithm (IDTask=2)
! e0        : Initial void ratio (IDTask=1, IDTask=2, IDTask=4, IDTask=6)
! eX        : Void ratio in RKDP algorithm (X=2,3,4,5,6) (IDTask=2)
! ftol      : Tolerance for zero value of yield function (IDTask=2)
! f0        : Value of yield function that must be corrected - drift correction
(IDTask=2)
! G         : Shear modulus at start of step (IDTask=2, IDTask=4, IDTask=6)
! GX        : Shear modulus in RKDM algorithm (X=1,2,3,4,5,6) (IDTask=2)
! Gd        : Shear modulus in drift correction algorithm (IDTask=2)
! Gbrick    : Shear modulus when i brick strings are taut (IDTask=1)
! Gur       : Shear modulus for URL at pref=100 kPa (IDTask=1, IDTask=2)
! G0        : Small strain shear modulus at pref=100 (IDTask=1, IDTask=2,
IDTask=4, IDTask=6)
! G1s      : Shear modulus at start of subincrement for p=100 kPa (IDTask=2)
! G2s      : Shear modulus at the end of step for p=100 kPa (IDTask=2)
! Gama      : Specific volume for CLS at p'=1 kPa (IDTask=1, IDTask=2)
! iDrift    : Counter in drift correction loop (IDTask=2)
! isuc      : =0 if previous subincrement is successful, =1 if not (IDTask=2)
! iss       : Counts the number of successful subincrements (IDTask=2)
! its       : Counts the number of total subincrements (IDTask=2)
! Nsubs     : Number of subincrements (IDTask=2)
! nStatV    : Number of used state variables
! OCR       : Input value of overconsolidation ratio (IDTask=1)
! omega     : Hardening coefficient (IDTask=1, IDTask=2)
! omegaX    : Hardening coefficient in RKDP (IDTask=2)
! omegalim  : Limiting value for hardening coefficient in order to avoid
numerical instability (IDTask=2)
! omega0    : Hardening coefficient in drift correction algorithm (IDTask=2)
! p         : mean effective stress (IDTask=2)
! pX        : mean effective stress in RKDP algorithm (X=1,2,3,4,5,6) (IDTask=2)
! pi        : Initial value of mean effective stress
! Pie       : Value of Pi (IDTask=1)
! pmax      : Max previous mean effective stress (IDTask=1)
! POM1, POM2, ... : Auxilliary real variables
! POP       : Input value of preoverburden pressure (IDTask=1)
! pt        : Mean effective stress at start of step (IDTask=2, IDTask=4,
IDTask=6)
! ptt       : mean effective stress at the end of step (IDTask=2)
! PSINC     : State parameter for NC soil (IDTask=1, IDTask=2)
! PSINCX    : State parameter for NC soil in RKDP algorithm (X=1,2,3,4,5,6)
(IDTask=2)

```



```

! PSINC0      : State parameter for NC soil in drift correction algorithm
(IDTask=2)
! PSIOC      : State parameter for OC soil ((IDTask=1, IDTask=2)
! PSIOCX     : State parameter for OC soil in RKDP algorithm (X=1,2,3,4,5,6)
(IDTask=2)
! PSIOC0     : State parameter for OC soil in drift correction algorithm
(IDTask=2)
! p0         : Hardening parameter for yield surface (IDTask=1, IDTask=2)
! p0bs       : Hardening parameter for bounding surface (IDTask=1, IDTask=2)
! p0t        : Hardening parameter for current yield surface at start of step
(IDTask=2)
! p0tt       : Hardening parameter for current yield surface at start of step
(IDTask=2)
! p0X        : Hardening parameter for current yield surface in RKDP algorithm
(X=1,2,3,4,5,6) (IDTask=2)
! q          : Shear deviator stress (IDTask=2)
! qX         : Shear deviator stress in RKDP algorithm (X=1,2,3,4,5,6) (IDTask=2)
! qi         : Initial value of shear deviator stress (IDTask=1)
! qmax       : Max previous shear deviator stress (IDTask=1)
! qt         : Shear deviator stress at start of step (IDTask=2)
! qtt        : Shear deviator stress at the end of step (IDTask=2)
! R          : Isotropic OCR ratio (IDTask=1, IDTask=2)
! Rmax       : Max value of R in order to avoid numerical instability (IDTask=1,
IDTask=2)
! Rmin       : Min value of R in order to avoid non-possible values (IDTask=1,
IDTask=2)
! RX         : Isotropic OCR ratio in RKDP algorithm (X=1,2,3,4,5,6) (IDTask=2)
! R0         : Isotropic OCR ratio in drift correction algorithm (IDTask=2)
! stepsize   : Relative size of next subincrement size to the previous
subincrement size (IDTask=2)
! subtime1   : Time at the begining of subincrement
! subtime2   : Time at the end of subincrement
! STOL       : Error tolerance for evaluation of subincrement size (IDTask=2)
! Sxmax      : Max previous horizontal stress in x direction (IDTask=1)
! Symax      : Max previous vertical stress (IDTask=1)
! Szmax      : Max previous horizontal stress in y direction (IDTask=1)
! Te         : Void ratio at start of subincrement
! theta      : Lodes angle (IDTask=2)
! thetat     : Lodes angle at start of step (IDTask=2)
! thetaX     : Lodes angle in RKDP algorithm (X=1,2,3,4,5,6) (IDTask=2)
! TTe        : Void ratio at the end of subincrement (IDTask=2)
! TTf        : Yield function at the end of subincrement (IDTask=2)
! TTp        : Mean effective stress at start of subincrement (IDTask=2)
! TTp0       : Hardening parameter at the end of subincrement (IDTask=2)
! TTq        : Deviator stress at the end of subincrement (IDTask=2)
! TTtheta    : Lodes angle at the end of subincrement (IDTask=2)
! VarA0      : Variable A in stress integration procedure (drift correction)
(IDTask=2)
! VarAX      : Variable A in stress integration procedure (RKDP) (X=1,2,3,4,5,6)
(IDTask=2)
! VarB0      : Variable B in stress integration procedure (drift correction)
(IDTask=2)
! VarBX      : Variable B in stress integration procedure (RKDP) (X=1,2,3,4,5,6)
(IDTask=2)
! xk         : kappa, slope of U/R line (IDTask=1, IDTask=2, IDTask=4, IDTask=6)
! xka        : small strain kappa (IDTask=1, IDTask=2)
! xkd        : kappa for drift correction (IDTask=2)
! xkX        : kappa in (RKDP) (IDTask=2)
! xKNC       : Ratio of horizontal and vertical stresses at NC state (IDTask=1)
! xL         : Lambda, slope of ICL (IDTask=1, IDTask=2, IDTask=4, IDTask=6))
! xM         : M, slope of CSL in p-q plane (IDTask=1, IDTask=2)
! xMc        : slope of CSL in p-q plane in TXC (IDTask=1, IDTask=2)

```

```

! xMe          : slope of CSL in p-q plane in TXE (IDTask=1, IDTask=2)
! xMs          : slope of CSL in p-q plane at the start of subincrement (IDTask=2)
! xMsX         : slope of CSL in P-q plane (RKDP) (X=1,2,3,4,5,6) (IDTask=2)
! xNu          : Poissons ratio (IDTask=1, IDTask=2, IDTask=4, IDTask=6)
! xX,yY,zZ     : Parameters that describe the dependency of M on Lode's angle
(IDTask=1, IDTask=2)
!
!--- Auxilliary Subroutines
!
! CalculateStressInvariants : Calculates p,q,eta and theta
! CalculateVoidRatio       : Calculates current void ratio
! CalculateElasticModuli   : Calculates bulk and shear modulus at start of step
! CalculateOmega           : Calculates hardening coefficient
! CalculateVectorA         : Calculates vector A which contains derivatives of
yield function (not dependent on Lode's angle)
! CalculateVectorA2        : Calculates vector A which contains derivatives of
yield function (dependent on Lode's angle)
! CalculateDeMatrix        : Calculates elastic D matrix
! MZEROR                  : Fills real array with zeros
! MatVec                   : Multiplies matrix with a vector. Result is vector
! DInProd                  : Inner product of two vectors. Result is scalar
! AddVec                   : Calculates sum of two vectors. result is vector
! CalculateElasticModuliBrick: Calculates shear modulus for Brick model
! COPYRVEC                 : Copies a Double array R1 with Dimension K to R2
!
! Dimension Stresst(6), Straint(6), dStrain(6), Straintt(6),
* dStress1(6), Stresstt(6), dSig(6), Va1(6), De1(6,6),
* dStresse1(6), AUXV11(6), AUXV21(6), Tstress(6),
* Tstrain(6), dTstrain(6), TTstress(6), TTstrain(6),
* Stress0(6), Va0(6), AUXV3(6), AUXV4(6), De0(6,6),
* Tstress1(6), Va2(6), dStresse2(6), De2(6,6),
* AUXV12(6), AUXV22(6), dStress2(6), dStress(6),
* Stress1(6), Stress2(6), Stress3(6), Va3(6), De3(6,6),
* dStresse3(6), AUXV13(6), AUXV23(6), dStress3(6),
* Stress4(6), Va4(6), De4(6,6), dStresse4(6), AUXV14(6),
* AUXV24(6), dStress4(6), Stress5(6), Va5(6), De5(6,6),
* dStresse5(6), AUXV15(6), AUXV25(6), dStress5(6),
* Stress6(6), Va6(6), De6(6,6), dStresse6(6), AUXV16(6),
* AUXV26(6), dStress6(6), StressError(6), SLen(20),
* SH(20,6), SHt(20,6), BT(20), Vb0(6), Vb1(6), Vb2(6),
* Vb3(6), Vb4(6), Vb5(6), Vb6(6)
! Parameter (ftol=1.D-9, Htol=1.D-6, EM=1.D-16,
*omegalim=15000, Rmax=100.0, Rmin=1.0,
*Pie=3.14159 26535 89793 23846 26433 83279 50288 41971 69399d0)

! Expected contents of Props(1..?)
! 1 : lambda    slope of ICL
! 2 : kappa     slope of U/R line
! 3 : Mc        slope of CSL in p-q plane in TXC
! 4 : Me        slope of CSL in p-q plane in TXE
! 5 : nu        Poisson ratio
! 6 : evoid0    Initial void ration
! 7 : OCR       overconsolidation ratio
! 8 : POP       preoverburden pressure
! 9 : K0NC      ratio of horizontal and vertical stresses
! 10 : STOL     Error tolerance for substepping
! 11 : Go,ref   small strain shear modulus at pref=100 kPa
! 12 : gama0,7  reference strain

nStatV = 180
If (IDTask .Eq. 1) Then ! Initialize state variables
    if (StVar0(1).eq.0) then !Reset state variables

```

```

! Get Props
xl=Props(1)
xk=Props(2)
xMc=Props(3)
xMe=Props(4)
if (xMe.eq.0.0) xMe=xMc
xNu=Props(5)
e0=Props(6)
OCR=Props(7)
POP=Props(8)
xKNC=Props(9)
G0=Props(11)
strain70=Props(12)
If (G0.gt.0.0.and.strain70.eq.0.0) Then
  Call OK_MessageBox(' Invalid value for "gama0.7" (param 12)')
*)
  iAbort = 1
  Return
End If
! Calculate Gama from OCR, POP
Symax=-OCR*Sig0(2)+POP
Sxmax=xKNC*Symax
Szmax=xKNC*Symax
pmax=(Symax+Sxmax+Szmax)/3
qmax=abs(Symax-Sxmax)
etamax=qmax/pmax
pi=-(Sig0(1)+Sig0(2)+Sig0(3))/3.0
qi=sqrt(3.0*(1.0/6.0*((Sig0(1)-Sig0(2))**2.0+
*(Sig0(2)-Sig0(3))**2.0+(Sig0(3)-Sig0(1))**2.0)+Sig0(4)**2.0+
*Sig0(5)**2.0+Sig0(6)**2.0))
etai=qi/pi
xM=xMc
POM1=(pmax*(etamax**2.0+xM**2.0)/2.0/xM**2.0)**((xl-xk)/xl)
POM2=pi**((xk/xl)
Gama=1.0+e0*xl*log(POM1*POM2)
PSIOC=1.0+e0*xl*log(pi)-Gama
PSINC=(xl-xk)*log(2.0*xM**2.0/(xM**2.0+etai**2.0))
R=exp((PSINC-PSIOC)/(xl-xk))
If (R.gt.Rmax) then
  Gama=1.0+e0*xl*log(pi)-PSINC+(xl-xk)*log(Rmax)
  R=Rmax
end if
if (R.lt.Rmin) then
  Gama=1.0+e0*xl*log(pi)-PSINC+(xl-xk)*log(Rmin)
  R=Rmin
end if
omega=(1.0+(PSINC-PSIOC)/PSINC)*R
p0=qi**2.0/(pi*xM**2.0)+pi
p0bs=R*p0
! Check if value of Poissons ratio is ok
If (xNu.Gt.0.5d0-1d-6 .Or. xNu.Lt.-1d-6) Then
  Call OK_MessageBox(' Invalid value for "nu" (param 5)')
  iAbort = 1
  Return
End If
if (xNu.Lt.0) then
  xNu = 0.0
  Props(5) = xNu
end if
if (xNu.Gt.0.495d0) then
  xNu = 0.495
  Props(5) = xNu

```

```

end if
! Initialize Brick model
Gur=3.0*(1.0-2.0*xNu)/(2.0*(1.0+xNu))*(1.0+e0)/xk*100.0
if (G0.lt.Gur) G0=Gur
dBG=(G0-Gur)/(20*G0)
do i=1,20
    Gbrick=(1.0-(real(i)-0.5)*dBG)*G0
    SLen(i)=strain70/0.385*(sqrt(G0/Gbrick)-1.0)
end do
G=G0*pi/100.0
! Fill state variable array
! Variables X, Y and Z for determination of parameter M
zZ=-0.229
xX=((xMc**(1.0/zZ)+xMe**(1.0/zZ))/2.0)**zZ
yY=(1.0-(xMc/xMe)**(1.0/zZ))/(1.0+(xMc/xMe)**(1.0/zZ))
do i=1,nStatV
    StVar0(i)=0.0
end do
StVar0(1)=1.0
StVar0(2)=Gama
StVar0(3)=R
StVar0(4)=PSIOC
StVar0(5)=PSINC
StVar0(6)=omega
StVar0(7)=p0
StVar0(8)=p0bs
StVar0(9)=Symax
StVar0(10)=Symax/(-Sig0(2))
StVar0(11)=3.0
StVar0(12)=G0
StVar0(13)=Gur
StVar0(14)=G
StVar0(15)=dBG
StVar0(16)=-30.0
StVar0(17)=xX
StVar0(18)=yY
StVar0(19)=xM
do i=1,20
    StVar0(20+i)=SLen(i)
end do
end if
If (iel+int.eq.2) Call WriVec(1, 'Props', Props, 12)
End If ! IDTask = 1
If (IDTask .Eq. 2) Then ! Calculate stresses
    If (IsUndr.Eq.1) Then
        dEpsV = dEps(1) + dEps(2) + dEps(3)
        dSwp = BulkW * dEpsV
        Swp = Swp0 + dSwp
    Else
        Swp = Swp0
    End If
    If (iEl+Int+iter.Eq.3 .And. iStep.Lt.10) Then
        Call WriMat( -1, 'D66', D, 6, 6, 6 )
    End If
! Material properties
xl=Props(1)
xk=Props(2)
xNu=Props(5)
e0=Props(6)
G0=Props(11)
! State variables
Gama=StVar0(2)

```

```

G1s=StVar0(12)
Gur=StVar0(13)
dBG=StVar0(15)
xX=StVar0(17)
yY=StVar0(18)
zZ=-0.229
do i=1,20
  SLen(i)=StVar0(20+i)
end do
do i=1,20
  do j=1,6
    SH(i,j)=StVar0(20+20*j+i)
  end do
end do
! Reversing sign for input stresses and strains
do i=1,6
  Stresst(i)=-Sig0(i)
  Straint(i)=-dEps(i+6)
  dStrain(i)=-dEps(i)
  Straintt(i)=Straint(i)+dStrain(i)
end do
!Stress and strain invariants
Call CalculateStressInvariants(Stresst,pt,qt,etat,thetat)
Call CalculateVoidRatio(Straint,e0,et)
!Calculate value of parameter M at the start of step
xM=xX*(1.0+yY*sin(3.0*thetat))*zZ
if (isnan(xM)) xM=xX*(1.0-yY)*zZ
!Material modules
if (G0.eq.0.0) Call CalculateElasticModuli(et,pt,xk,xNu,Bulk,G)
!Hardening parameter for curent yield surface
p0t=qt**2.0/pt/xM**2.0+pt
!Calculate OMEGA - hardening coefficient at the beginning of increment
Call CalculateOmega(xl,xk,xM,Gama,pt,qt,et,omegalim,Rmax,Rmin,
*omega,R,PSIOC,PSINC)
! Stress determination explicit fifth-order accurate (Runge-Kutta-Dormand-Price)
algorithm
!
! Initialization
STOL=Props(10)
if (STOL.lt.1.D-9) STOL=1.D-9
ipl=4
isuc=0
its=0
iss=0
subtime1=0.0
dsubtime=1.0
Call MZEROR(Tstress,6)
Call MZEROR(Tstrain,6)
Call MZEROR(dTstrain,6)
Call MZEROR(TTstress,6)
Call MZEROR(TTstrain,6)
Call COPYRVEC(Stresst,Tstress,6)
Call COPYRVEC(Straint,Tstrain,6)
do i=1,6
  dTstrain(i)=dStrain(i)*dsubtime
end do
Call AddVec(Tstrain,dTstrain,1.0,1.0,6,TTstrain)
dEv=dTstrain(1)+dTstrain(2)+dTstrain(3)
! Stress calculation for one subincrement
110 continue
its=its+1
! Calculation of dStress1 and dp01

```

```

Call MZEROR(Stress1,6)
Call COPYRVEC(Tstress,Stress1,6)
Call CalculateStressInvariants(Stress1,p1,q1,eta1,theta1)
xMs=xX*(1.0+yY*sin(3.0*theta1))*zZ
if (isnan(xMs)) xMs=xX*(1.0-yY)*zZ
Call CalculateVoidRatio(Tstrain,e0,Te)
Call CalculateVoidRatio(TTstrain,e0,TTe)
if (G0.eq.0.0) then
  Call CalculateElasticModuli(Te,p1,xk,xNu,Bulk1,G1)
  xk1=xk
else
  do i=1,20
    do j=1,6
      SHt(i,j)=SH(i,j)
    end do
  end do
  Call CalculateElasticModuliBrick(BT,TTstrain,SHt,SLen,
*G1s,G2s,Gur,G0,G1,Te,p1,xNu,xk1,dBG,Bulk1)
end if
p01=q1**2.0/p1/xMs**2.0+p1
Call CalculateOmega(xl,xk,xMs,Gama,p1,q1,Te,omegalim,Rmax,
*Rmin,omega1,R1,PSIOC1,PSINC1)
Call MZEROR(Va1,6)
Call CalculateVectorA2(p1,p01,q1,theta1,xMs,Stress1,xX,yY,
*zZ,Va1)
Call MZEROR(Vb1,6)
Call CalculateVectorA(p1,p01,xMs,Stress1,Vb1)
Call MZEROR(De1,36)
Call CalculateDeMatrix(Bulk1,G1,De1)
VarA1=(1.0+Te)*p01*omega1*p1*(2.0*p1-p01)/(xl-xk1)
VarB1=(1.0+Te)*p01*omega1*(2.0*p1-p01)/(xl-xk1)
Call MZEROR(dStresse1,6)
Call MatVec(De1,6,dTstrain,6,dStresse1)
AUX11=0.0
AUX11=DInProd(Va1,dStresse1,6)
Call MZEROR(AUXV11,6)
Call MatVec(De1,6,Va1,6,AUXV11)
AUX21=0.0
AUX21=DInProd(AUXV11,Vb1,6)
dlam1=AUX11/(VarA1+AUX21)
if (isnan(dlam1).or.dlam1.lt.0.0) dlam1=0.0
Call MZEROR(AUXV21,6)
Call MatVec(De1,6,Vb1,6,AUXV21)
Call MZEROR(dStress1,6)
Call AddVec(dStresse1,AUXV21,1.0,-dlam1,6,dStress1)
dp01=dlam1*VarB1
Call MZEROR(Stress2,6)
Call AddVec(Stress1,dStress1,1.0,0.2,6,Stress2)
p02=p01+0.2*dp01
e2=(1.0+Te)*exp(-0.2*dEv)-1.0
if (e2.lt.0.0) e2=0.0
if (e2.gt.1000.0) e2=1000.0
! Calculation of dStress2 and dp02
Call CalculateStressInvariants(Stress2,p2,q2,eta2,
*theta2)
xMs2=xX*(1.0+yY*sin(3.0*theta2))*zZ
if (isnan(xMs2)) xMs2=xX*(1.0-yY)*zZ
if (G0.eq.0.0) then
  Call CalculateElasticModuli(e2,p2,xk,xNu,Bulk2,G2)
  xk2=xk
else
  G2=G1*p2/p1

```

```

Bulk2=2.0*G2*(1.0+xNu)/(3.0*(1.0-2.0*xNu))
xk2=(1+e2)*p2*1.0/G2*3.0*(1.0-2.0*xNu)/(2.0*(1.0+xNu)
*)
end if
Call CalculateOmega(xl, xk, xMs2, Gama, p2, q2, e2, omegalim, Rmax,
*Rmin, omega2, R2, PSIOC2, PSINC2)
Call MZEROR(Va2, 6)
Call CalculateVectorA2(p2, p02, q2, theta2, xMs2, Stress2, xX, yY,
*zZ, Va2)
Call MZEROR(Vb2, 6)
Call CalculateVectorA(p2, p02, xMs2, Stress2, Vb2)
Call MZEROR(De2, 36)
Call CalculateDeMatrix(Bulk2, G2, De2)
VarA2=(1.0+e2)*p02*omega2*p2*(2.0*p2-p02)/(xl-xk2)
VarB2=(1.0+e2)*p02*omega2*(2.0*p2-p02)/(xl-xk2)
Call MZEROR(dStresse2, 6)
Call MatVec(De2, 6, dTstrain, 6, dStresse2)
AUX12=0.0
AUX12=DInProd(Va2, dStresse2, 6)
Call MZEROR(AUXV12, 6)
Call MatVec(De2, 6, Va2, 6, AUXV12)
AUX22=0.0
AUX22=DInProd(AUXV12, Vb2, 6)
dlam2=AUX12/(VarA2+AUX22)
if (isnan(dlam2).or.dlam2.lt.0.0) dlam2=0.0
Call MZEROR(AUXV22, 6)
Call MatVec(De2, 6, Vb2, 6, AUXV22)
Call MZEROR(dStress2, 6)
Call AddVec(dStresse2, AUXV22, 1.0, -dlam2, 6, dStress2)
dp02=dlam2*VarB2
Call MZEROR(Stress3, 6)
do j=1, 6
Stress3(j)=Stress1(j)+3.0/40.0*dStress1(j)+9.0/40.0*
*dStress2(j)
end do
p03=p01+3.0/40.0*dp01+9.0/40.0*dp02
e3=(1.0+Te)*exp(-(3.0/40+9.0/40)*dEv)-1.0
if (e3.lt.0.0) e3=0.0
if (e3.gt.1000.0) e3=1000.0
! Calculation of dStress3 and dp03
Call CalculateStressInvariants(Stress3, p3, q3, eta3,
*theta3)
xMs3=xX*(1.0+yY*sin(3.0*theta3))*zZ
if (isnan(xMs3)) xMs3=xX*(1.0-yY)*zZ
if (G0.eq.0.0) then
Call CalculateElasticModuli(e3, p3, xk, xNu, Bulk3, G3)
xk3=xk
else
G3=G1*p3/p1
Bulk3=2.0*G3*(1.0+xNu)/(3.0*(1.0-2.0*xNu))
xk3=(1+e3)*p3*1.0/G3*3.0*(1.0-2.0*xNu)/(2.0*(1.0+xNu)
*)
end if
Call CalculateOmega(xl, xk, xMs3, Gama, p3, q3, e3, omegalim, Rmax,
*Rmin, omega3, R3, PSIOC3, PSINC3)
Call MZEROR(Va3, 6)
Call CalculateVectorA2(p3, p03, q3, theta3, xMs3, Stress3, xX, yY,
*zZ, Va3)
Call MZEROR(Vb3, 6)
Call CalculateVectorA(p3, p03, xMs3, Stress3, Vb3)
Call MZEROR(De3, 36)
Call CalculateDeMatrix(Bulk3, G3, De3)

```



```

VarA3=(1.0+e3)*p03*omega3*p3*(2.0*p3-p03)/(xl-xk3)
VarB3=(1.0+e3)*p03*omega3*(2.0*p3-p03)/(xl-xk3)
Call MZEROR(dStresse3,6)
Call MatVec(De3,6,dTstrain,6,dStresse3)
AUX13=0.0
AUX13=DInProd(Va3,dStresse3,6)
Call MZEROR(AUXV13,6)
Call MatVec(De3,6,Va3,6,AUXV13)
AUX23=0.0
AUX23=DInProd(AUXV13,Vb3,6)
dlam3=AUX13/(VarA3+AUX23)
if (isnan(dlam3).or.dlam3.lt.0.0) dlam3=0.0
Call MZEROR(AUXV23,6)
Call MatVec(De3,6,Vb3,6,AUXV23)
Call MZEROR(dStress3,6)
Call AddVec(dStresse3,AUXV23,1.0,-dlam3,6,dStress3)
dp03=dlam3*VarB3
Call MZEROR(Stress4,6)
do j=1,6
    Stress4(j)=Stress1(j)+3.0/10.0*dStress1(j)-9.0/10.0*
*dStress2(j)+6.0/5.0*dStress3(j)
end do
p04=p01+3.0/10.0*dp01-9.0/10.0*dp02+6.0/5.0*dp03
e4=(1.0+Te)*exp(-(3.0/10-9.0/10+6.0/5.0)*dEv)-1.0
if (e4.lt.0.0) e4=0.0
if (e4.gt.1000.0) e4=1000.0
! Calculation of dStress4 and dp04
Call CalculateStressInvariants(Stress4,p4,q4,eta4,
*theta4)
xMs4=xX*(1.0+yY*sin(3.0*theta4))*zZ
if (isnan(xMs4)) xMs4=xX*(1.0-yY)*zZ
if (G0.eq.0.0) then
    Call CalculateElasticModuli(e4,p4,xk,xNu,Bulk4,G4)
    xk4=xk
else
    G4=G1*p4/p1
    Bulk4=2.0*G4*(1.0+xNu)/(3.0*(1.0-2.0*xNu))
    xk4=(1+e4)*p4*1.0/G4*3.0*(1.0-2.0*xNu)/(2.0*(1.0+xNu)
*)
end if
Call CalculateOmega(xl,xk,xMs4,Gama,p4,q4,e4,omegalim,Rmax,
*Rmin,omega4,R4,PSIOC4,PSINC4)
Call MZEROR(Va4,6)
Call CalculateVectorA2(p4,p04,q4,theta4,xMs4,Stress4,xX,yY,
*zZ,Va4)
Call MZEROR(Vb4,6)
Call CalculateVectorA(p4,p04,xMs4,Stress4,Vb4)
Call MZEROR(De4,36)
Call CalculateDeMatrix(Bulk4,G4,De4)
VarA4=(1.0+e4)*p04*omega4*p4*(2.0*p4-p04)/(xl-xk4)
VarB4=(1.0+e4)*p04*omega4*(2.0*p4-p04)/(xl-xk4)
Call MZEROR(dStresse4,6)
Call MatVec(De4,6,dTstrain,6,dStresse4)
AUX14=0.0
AUX14=DInProd(Va4,dStresse4,6)
Call MZEROR(AUXV14,6)
Call MatVec(De4,6,Va4,6,AUXV14)
AUX24=0.0
AUX24=DInProd(AUXV14,Vb4,6)
dlam4=AUX14/(VarA4+AUX24)
if (isnan(dlam4).or.dlam4.lt.0.0) dlam4=0.0
Call MZEROR(AUXV24,6)

```



```

Call MatVec(De4, 6, Vb4, 6, AUXV24)
Call MZEROR(dStress4, 6)
Call AddVec(dStresse4, AUXV24, 1.0, -dlam4, 6, dStress4)
dp04=dlam4*VarB4
Call MZEROR(Stress5, 6)
do j=1, 6
    Stress5(j)=Stress1(j)+226.0/729.0*dStress1(j)-
*25.0/27.0*dStress2(j)+880.0/729.0*dStress3(j)+55.0/729.0*
*dStress4(j)
end do
p05=p01+226.0/729.0*dp01-25.0/27.0*dp02+880.0/729.0*dp03+
*55.0/729.0*dp04
e5=(1.0+Te)*exp(-(226.0/729.0-25.0/27.0+880.0/729.0+55.0/
*729.0)*dEv)-1.0
if (e5.lt.0.0) e5=0.0
if (e5.gt.1000.0) e5=1000.0
! Calculation of dStress5 and dp05
Call CalculateStressInvariants(Stress5, p5, q5, eta5,
*theta5)
xMs5=xX*(1.0+yY*sin(3.0*theta5))*zZ
if (isnan(xMs5)) xMs5=xX*(1.0-yY)*zZ
if (G0.eq.0.0) then
    Call CalculateElasticModuli(e5, p5, xk, xNu, Bulk5, G5)
    xk5=xk
else
    G5=G1*p5/p1
    Bulk5=2.0*G5*(1.0+xNu)/(3.0*(1.0-2.0*xNu))
    xk5=(1+e5)*p5*1.0/G5*3.0*(1.0-2.0*xNu)/(2.0*(1.0+xNu)
*)
end if
Call CalculateOmega(xl, xk, xMs5, Gama, p5, q5, e5, omegalim, Rmax,
*Rmin, omega5, R5, PSIOC5, PSINC5)
Call MZEROR(Va5, 6)
Call CalculateVectorA2(p5, p05, q5, theta5, xMs5, Stress5, xX, yY,
*zZ, Va5)
Call MZEROR(Vb5, 6)
Call CalculateVectorA(p5, p05, xMs5, Stress5, Vb5)
Call MZEROR(De5, 36)
Call CalculateDeMatrix(Bulk5, G5, De5)
VarA5=(1.0+e5)*p05*omega5*p5*(2.0*p5-p05)/(xl-xk5)
VarB5=(1.0+e5)*p05*omega5*(2.0*p5-p05)/(xl-xk5)
Call MZEROR(dStresse5, 6)
Call MatVec(De5, 6, dTstrain, 6, dStresse5)
AUX15=0.0
AUX15=DInProd(Va5, dStresse5, 6)
Call MZEROR(AUXV15, 6)
Call MatVec(De5, 6, Va5, 6, AUXV15)
AUX25=0.0
AUX25=DInProd(AUXV15, Vb5, 6)
dlam5=AUX15/(VarA5+AUX25)
if (isnan(dlam5).or.dlam5.lt.0.0) dlam5=0.0
Call MZEROR(AUXV25, 6)
Call MatVec(De5, 6, Vb5, 6, AUXV25)
Call MZEROR(dStress5, 6)
Call AddVec(dStresse5, AUXV25, 1.0, -dlam5, 6, dStress5)
dp05=dlam5*VarB5
Call MZEROR(Stress6, 6)
do j=1, 6
    Stress6(j)=Stress1(j)-181.0/270.0*dStress1(j)+
*5.0/2.0*dStress2(j)-266.0/297.0*dStress3(j)-91.0/27.0*
*dStress4(j)+189.0/55.0*dStress5(j)
end do

```

```

    p06=p01-181.0/270.0*dp01+5.0/2.0*dp02-266.0/297.0*dp03-
*91.0/27.0*dp04+189.0/55.0*dp05
    e6=(1.0+Te)*exp(-(-181.0/270.0+5.0/2.0-266.0/297.0-91.0/27.0
*+189.0/55.0)*dEv)-1.0
    if (e6.lt.0.0) e6=0.0
    if (e6.gt.1000.0) e6=1000.0
    !Calculation of dStress6 and dp06
    Call CalculateStressInvariants(Stress6,p6,q6,eta6,
*theta6)
    xMs6=xX*(1.0+yY*sin(3.0*theta6))*zZ
    if (isnan(xMs6)) xMs6=xX*(1.0-yY)*zZ
    if (G0.eq.0.0) then
        Call CalculateElasticModuli(e6,p6,xk,xNu,Bulk6,G6)
        xk6=xk
    else
        G6=G1*p6/p1
        Bulk6=2.0*G6*(1.0+xNu)/(3.0*(1.0-2.0*xNu))
        xk6=(1+e6)*p6*1.0/G6*3.0*(1.0-2.0*xNu)/(2.0*(1.0+xNu)
*)
    end if
    Call CalculateOmega(xl,xk,xMs6,Gama,p6,q6,e6,omegalim,Rmax,
*Rmin,omega6,R6,PSIOC6,PSINC6)
    Call MZEROR(Va6,6)
    Call CalculateVectorA2(p6,p06,q6,theta6,xMs6,Stress6,xX,yY,
*zZ,Va6)
    Call MZEROR(Vb6,6)
    Call CalculateVectorA(p6,p06,xMs6,Stress6,Vb6)
    Call MZEROR(De6,36)
    Call CalculateDeMatrix(Bulk6,G6,De6)
    VarA6=(1.0+e6)*p06*omega6*p6*(2.0*p6-p06)/(xl-xk6)
    VarB6=(1.0+e6)*p06*omega6*(2.0*p6-p06)/(xl-xk6)
    Call MZEROR(dStresse6,6)
    Call MatVec(De6,6,dTstrain,6,dStresse6)
    AUX16=0.0
    AUX16=DInProd(Va6,dStresse6,6)
    Call MZEROR(AUXV16,6)
    Call MatVec(De6,6,Va6,6,AUXV16)
    AUX26=0.0
    AUX26=DInProd(AUXV16,Vb6,6)
    dlam6=AUX16/(VarA6+AUX26)
    if (isnan(dlam6).or.dlam6.lt.0.0) dlam6=0.0
    Call MZEROR(AUXV26,6)
    Call MatVec(De6,6,Vb6,6,AUXV26)
    Call MZEROR(dStresse6,6)
    Call AddVec(dStresse6,AUXV26,1.0,-dlam6,6,dStresse6)
    dp06=dlam6*VarB6
    ! Stresses and hardening parameter for 5th order
    Call MZEROR(TTstress,6)
    do j=1,6
        TTstress(j)=Stress1(j)+19.0/216.0*dStress1(j)+
*1000.0/2079.0*dStress3(j)-125.0/216.0*dStress4(j)+
*81.0/88.0*dStress5(j)+5.0/56.0*dStress6(j)
    end do
    TTp0=p01+19.0/216.0*dp01+1000.0/2079.0*dp03-
*125.0/216.0*dp04+81.0/88.0*dp05+5.0/56.0*dp06
! Error estimation
    Call MZEROR(StressError,6)
    do j=1,6
        StressError(j)=11.0/360.0*dStress1(j)-10.0/63.0*
*dStress3(j)+55.0/72.0*dStress4(j)-27.0/40.0*dStress5(j)+
*11.0/280.0*dStress6(j)
    end do

```

```

    p0Error=11.0/360.0*dp01-11.0/63.0*dp03+55.0/72.0*dp04-
*27.0/40.0*dp05+11.0/280.0*dp06
    Error1=sqrt(StressError(1)**2.0+StressError(2)**2.0+
*StressError(3)**2.0+StressError(4)**2.0+StressError(5)**2.0+
*StressError(6)**2.0)/sqrt(TTstress(1)**2.0+TTstress(2)**2.0+
*TTstress(3)**2.0+TTstress(4)**2.0+TTstress(5)**2.0+
*TTstress(6)**2.0)
    Error2=abs(p0Error)/TTp0
    Error=Error1
    if (Error2.gt.Error) Error=Error2
    if (EM.gt.Error) Error=EM
    if (Error.gt.STOL.and.dsubtime.gt.0.000001) then
        stepsize=0.9*((STOL/Error)**(1.0/5.0))
        if (stepsize.lt.0.1) stepsize=0.1
        dsubtime=stepsize*dsubtime
        if (dsubtime.lt.0.000001) dsubtime=0.000001
        Call MZEROR(dTstrain,6)
        do i=1,6
            dTstrain(i)=dStrain(i)*dsubtime
        end do
        Call MZEROR(TTstrain,6)
        Call AddVec(Tstrain,dTstrain,1.0,1.0,6,TTstrain)
        dEv=dTstrain(1)+dTstrain(2)+dTstrain(3)
        isuc=1
        goto 110
    end if
    Call CalculateStressInvariants(TTstress,TTp,TTq,TTeta,
*TTtheta)
    TTf=TTq**2.0/xMs**2.0+TTp*(TTp-TTp0)
    if (dlam1.ne.0.0.and.dlam2.ne.0.0.and.dlam3.ne.0.0.and.
*dlam4.ne.0.0.and.dlam5.ne.0.0.and.dlam6.ne.0.0.and.
*abs(TTf).gt.ftol) then
        do iDrift=1,10
            Call MZEROR(Stress0,6)
            Call COPYRVEC(TTstress,Stress0,6)
            Call CalculateStressInvariants(Stress0,p,q,eta,
*theta)

            p0=TTp0
            f0=TTf
            if (G0.eq.0.0) then
                Call CalculateElasticModuli(TTe,p,xk,xNu,Bulkd,
*Gd)

                xkd=xk
            else
                Gd=G1*p/p1
                Bulkd=2.0*Gd*(1.0+xNu)/(3.0*(1.0-2.0*xNu))
                xkd=(1+TTe)*p*1.0/Gd*3.0*(1.0-2.0*xNu)/(2.0*(1.0
*+xNu))

            end if
            Call CalculateDeMatrix(Bulkd,Gd,De0)
            Call CalculateOmega(xl,xk,xMs,Gama,p,q,TTe,omegalim,
*Rmax,Rmin,omega0,R0,PSIOC0,PSINC0)
            Call MZEROR(Va0,6)
            Call CalculateVectorA2(p,p0,q,theta,xMs,Stress0,xX,
*yY,zZ,Va0)

            Call MZEROR(Vb0,6)
            Call CalculateVectorA(p,p0,xMs,Stress0,Vb0)
            VarA0=(1.0+TTe)*p0*omega0*p*(2.0*p-p0)/(xl-xkd)
            VarB0=(1.0+TTe)*p0*omega0*(2.0*p-p0)/(xl-xkd)
            Call MZEROR(AUXV3,6)
            Call MatVec(De0,6,Va0,6,AUXV3)
            AUX3=DInProd(AUXV3,Vb0,6)

```

```

        dlam=f0/(VarA0+AUX3)
        Call MZEROR(AUXV4,6)
        Call MatVec(De0,6,Vb0,6,AUXV4)
        Call MZEROR(TTstress,6)
        Call AddVec(Stress0,AUXV4,1.0,-dlam,6,TTstress)
        TTp0=p0+dlam*VarB0
        Call CalculateStressInvariants(TTstress,TTp,TTq,
*TTeta,TTtheta)
        TTf=TTq**2.0/xMs**2.0+TTp*(TTp-TTp0)
        If (abs(TTf).gt.abs(f0)) then
            AUX4=DInProd(Va0,Va0,6)
            dlam=f0/AUX4
            Call MZEROR(TTstress,6)
            Call AddVec(Stress0,Va0,1.0,-dlam,6,TTstress)
            TTp0=p0
            Call CalculateStressInvariants(TTstress,TTp,TTq,
*TTetatt,TTtheta)
            TTf=TTq**2.0/xMs**2.0+TTp*(TTp-TTp0)
            end if
            if (abs(TTf).lt.ftol) exit
        end do
        end if
        iss=iss+1
        subtime2=subtime1+dsubtime
        stepsize=0.9*((STOL/Error)**(1.0/5.0))
        if (stepsize.gt.1.1) stepsize=1.1
        if (isuc.eq.1) then
            if (stepsize.gt.1.0) stepsize=1.0
        end if
        dsubtime=stepsize*dsubtime
        if (dsubtime.lt.0.000001) dsubtime=0.000001
        if ((subtime2+dsubtime).gt.1.0) dsubtime=1.0-subtime2
        isuc=0
        Call MZEROR(Tstress,6)
        Call COPYRVEC(TTstress,Tstress,6)
        Call MZEROR(Tstrain,6)
        Call COPYRVEC(TTstrain,Tstrain,6)
        Call MZEROR(dTstrain,6)
        do i=1,6
            dTstrain(i)=dStrain(i)*dsubtime
        end do
        Call MZEROR(TTstrain,6)
        Call AddVec(Tstrain,dTstrain,1.0,1.0,6,TTStrain)
        dEv=dTstrain(1)+dTstrain(2)+dTstrain(3)
        subtime1=subtime2
        if (G0.ne.0.0) then
            G1s=G2s
            do i=1,20
                do j=1,6
                    SH(i,j)=SHt(i,j)
                end do
            end do
        end if
        if (subtime2.lt.1.0) goto 110

! Final stresses and hardening parameter
Call MZEROR(Stresstt,6)
Call COPYRVEC(TTstress,Stresstt,6)
p0tt=TTp0
!
!Return stresses
do i=1,6

```

```

        Sig(i)=-Stresstt(i)
    end do
!Update state variables
    StVar(1)=StVar0(1)
    StVar(2)=StVar0(2)
    StVar(3)=R
    StVar(4)=PSIOC
    StVar(5)=PSINC
    StVar(6)=omega
    StVar(7)=p0t
    StVar(8)=p0t*R
    If (StVar0(9).lt.Stresstt(2)) then
        StVar(9)=Stresstt(2)
    else
        StVar(9)=StVar0(9)
    end if
    StVar(10)=StVar(9)/Stresstt(2)
    StVar(11)=3.0
    StVar(16)=thetat*180.0/Pie
    StVar(17)=StVar0(17)
    StVar(18)=StVar0(18)
    StVar(19)=xM
    StVar(20)=Error
    if (G0.eq.0.0) then
        StVar(12)=Gur
        StVar(13)=StVar0(13)
        StVar(14)=G
        StVar(15)=StVar0(15)
    else
        StVar(12)=G2s
        StVar(13)=StVar0(13)
        StVar(14)=G1
        StVar(15)=StVar0(15)
        do i=1,20
            StVar(20+i)=StVar0(20+i)
        end do
        do i=1,20
            do j=1,6
                StVar(20+20*j+i)=SH(i,j)
            end do
        end do
        do i=1,20
            StVar(160+i)=BT(i)
        end do
    end if
End If ! IDTask = 2
If ( IDTask .Eq. 3 .Or.
* IDTask .Eq. 6 ) Then ! Calculate D-Matrix
    xk=Props(2)
    xNu=Props(5)
    e0=Props(6)
    G0=Props(11)
    pt=-((Sig0(1)+Sig0(2)+Sig0(3))/3.0
    if (G0.eq.0.0) then
        EpsV=-((dEps(7)+dEps(8)+dEps(9))
        et=(1.0+e0)*exp(-EpsV)-1.0
        Call CalculateElasticModuli(et,pt,xk,xNu,Bulk,G)
    else
        G=G0*pt/100.0
        if (G.lt.1.0) G=1.0
        Bulk=2.0*(1.0+xNu)/(3.0*(1.0-2.0*xNu))*G
    end if

```

```

D1=Bulk+4.0*G/3.0
D2=Bulk-2.0*G/3.0
D3=G
Call MZeroR(D,36)
D(1,1)=D1
D(2,2)=D1
D(3,3)=D1
D(1,2)=D2
D(1,3)=D2
D(2,1)=D2
D(2,3)=D2
D(3,1)=D2
D(3,2)=D2
D(4,4)=D3
D(5,5)=D3
D(6,6)=D3
BulkW = 0
If (IsUndr.Eq.1) Then
  ! BulkW = ...
  xNu_U = 0.495d0
  Fac=(1+xNu_U)/(1-2*xNu_U) - (1+xNu)/(1-2*xNu)
  Fac=2D0*G/3D0 * Fac
  BulkW = Fac
End If
End If ! IDTask = 3, 6

If (IDTask .Eq. 4) Then ! Number of state parameters
  nStat = nStatV
End If ! IDTask = 4

If (IDTask .Eq. 5) Then ! matrix type
  NonSym = 0 ! 1 for non-symmetric D-matrix
  iStrsDep = 1 ! 1 for stress dependent D-matrix
  iTang = 0 ! 1 for tangent D-matrix
  iTimeDep = 0 ! 1 for time dependent D-matrix
End If ! IDTask = 5

Return
End ! MyModel3

```

Биографија

Никола Обрадовић рођен је 1. децембра 1991. године у Ваљеву. Основну школу завршио је у Лајковцу, а гимназију (природно-математички смер) завршио је у Лазаревцу. Носилац је Вукове дипломе у основној школи и гимназији. Основне академске студије на Грађевинском факултету Универзитета у Београду уписао је 2010. године на студијском програму Грађевинарство. На Модулу за конструкције дипломирао је 6. октобра 2014. године са просечном оценом 9.75 и оценом 10 на синтезном пројекту са темом “Пројекат армиранобетонске конструкције вишеспратно стамбено пословне зграде према Еврокоду”. Мастер академске студије на Грађевинском факултету Универзитета у Београду уписао је 2014. године, и завршио 29. маја 2015. године на Модулу за конструкције са просечном оценом 9.86 и оценом 10 на дипломском-мастер раду са темом “Пројекат темељне конструкције ветрогенератора према задатим подацима”. Докторске академске студије на Грађевинском факултету Универзитета у Београду уписао је у октобру 2015. година на студијском програму Грађевинарство.

На Грађевинском факултету Универзитета у Београду запослен је од фебруара 2016. године, када је изабран у звање асистента-студента докторских студија за ужу научну област Фундирање, на Катедри за Грађевинску геотехнику. Током свог рада, држао је вежбе из предмета Основе фундирања, Фундирање, Специјални проблеми фундирања, Потпорне конструкције и Основе геотехничких конструкција. Учествовао је у изради преко тридесет завршних радова из области Фундирања.

Аутор је и коаутор више научних радова: два рада у међународним часописима индексираним на SCI листи, преко петнаест радова публикованих у зборницима домаћих и међународних конференција, и једног техничког решења. Уже подручје научно истраживачког рада је Механика тла, Фундирање и Геоеколошко инжењерство. Био је ангажован у оквиру пројекта ТР 36046 (2018-2020), који је финансирало Министарство просвете, науке и технолошког развоја Републике Србије.

Током рада на Грађевинском факултету Универзитета у Београду, у оквиру рада Института за саобраћајнице и геотехнику, учествовао је у изради неколико пројеката темељних конструкција, пројеката санације темељних конструкција, у изради стручних мишљења и техничких контрола. Члан је Српског и Међународног друштва (ISSMGE) за механику тла и геотехничко инжењерство, комисије за стандарде KS U182 - Геотехника Института за стандардизацију Србије и Инжењерске коморе Србије.

Изјава о ауторству

Име и презиме аутора: Никола Д. Обрадовић

Број индекса: 901/15

Изјављујем

да је докторска дисертација под насловом

Развој “HASP” конститутивног модела за преконсолидоване глине са применом у нумеричкој анализи методом коначних елемената

наслов на енглеском језику:

Development of “HASP” constitutive model for overconsolidated clays with application in finite element numerical analysis

- резултат сопственог истраживачког рада;
- да дисертација у целини ни у деловима није била предложена за стицање друге дипломе према студијским програмима других високошколских установа;
- да су резултати коректно наведени и
- да нисам кршио/ла ауторска права и користио/ла интелектуалну својину других лица.

Потпис аутора

У Београду, март 2024.



Изјава о истоветности штампане и електронске верзије докторског рада

Име и презиме аутора: Никола Д. Обрадовић
Број индекса: 901/15
Студијски програм: Грађевинарство
Наслов рада: **Развој “HASP” конститутивног модела за
преконсолидоване глине са применом у
нумеричкој анализи методом коначних елемената**
Ментор: др Сања Јоцковић, доцент, Универзитет у Београду,
Грађевински факултет

Изјављујем да је штампана верзија мог докторског рада истоветна електронској верзији коју сам предао/ла ради похрањена у **Дигиталном репозиторијуму Универзитета у Београду**.

Дозвољавам да се објаве моји лични подаци везани за добијање академског назива доктора наука, као што су име и презиме, година и место рођења и датум одбране рада.

Ови лични подаци могу се објавити на мрежним страницама дигиталне библиотеке, у електронском каталогу и у публикацијама Универзитета у Београду.

Потпис аутора

У Београду, март 2024.



Изјава о коришћењу

Овлашћујем Универзитетску библиотеку „Светозар Марковић“ да у Дигитални репозиторијум Универзитета у Београду унесе моју докторску дисертацију под насловом:

Развој “HASP” конститутивног модела за преконсолидоване глине са применом у нумеричкој анализи методом коначних елемената

(Development of “HASP” constitutive model for overconsolidated clays with application in finite element numerical analysis)

која је моје ауторско дело.

Дисертацију са свим прилозима предао/ла сам у електронском формату погодном за трајно архивирање.

Моју докторску дисертацију похрањену у Дигиталном репозиторијуму Универзитета у Београду и доступну у отвореном приступу могу да користе сви који поштују одредбе садржане у одабраном типу лиценце Креативне заједнице (Creative Commons) за коју сам се одлучио/ла.

1. Ауторство (CC BY)
2. Ауторство – некомерцијално (CC BY-NC)
3. Ауторство – некомерцијално – без прерада (CC BY-NC-ND)
4. Ауторство – некомерцијално – делити под истим условима (CC BY-NC-SA)
5. Ауторство – без прерада (CC BY-ND)
6. Ауторство – делити под истим условима (CC BY-SA)

(Молимо да заокружите само једну од шест понуђених лиценци. Кратак опис лиценци је саставни део ове изјаве).

Потпис аутора

У Београду, март 2024.



1. **Ауторство.** Дозвољаваате умножавање, дистрибуцију и јавно саопштавање дела, и прераде, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце, чак и у комерцијалне сврхе. Ово је најслободнија од свих лиценци.

2. **Ауторство – некомерцијално.** Дозвољаваате умножавање, дистрибуцију и јавно саопштавање дела, и прераде, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце. Ова лиценца не дозвољава комерцијалну употребу дела.

3. **Ауторство – некомерцијално – без прерада.** Дозвољаваате умножавање, дистрибуцију и јавно саопштавање дела, без промена, преобликовања или употребе дела у свом делу, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце. Ова лиценца не дозвољава комерцијалну употребу дела. У односу на све остале лиценце, овом лиценцом се ограничава највећи обим права коришћења дела.

4. **Ауторство – некомерцијално – делити под истим условима.** Дозвољаваате умножавање, дистрибуцију и јавно саопштавање дела, и прераде, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце и ако се прерада дистрибуира под истом или сличном лиценцом. Ова лиценца не дозвољава комерцијалну употребу дела и прерада.

5. **Ауторство – без прерада.** Дозвољаваате умножавање, дистрибуцију и јавно саопштавање дела, без промена, преобликовања или употребе дела у свом делу, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце. Ова лиценца дозвољава комерцијалну употребу дела.

6. **Ауторство – делити под истим условима.** Дозвољаваате умножавање, дистрибуцију и јавно саопштавање дела, и прераде, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце и ако се прерада дистрибуира под истом или сличном лиценцом. Ова лиценца дозвољава комерцијалну употребу дела и прерада. Слична је софтверским лиценцама, односно лиценцама отвореног кода.