Razvoj algoritma za proračun slijeganja jednoliko opterećene pravokutne površine

Strelec, Filip

Master's thesis / Diplomski rad

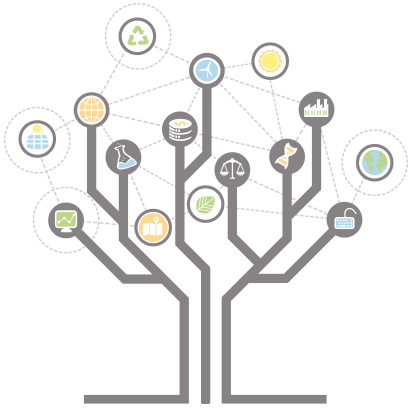
2020

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Geotechnical Engineering / Sveučilište u Zagrebu, Geotehnički fakultet**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:130:518273

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-03-29**



Repository / Repozitorij:

Repository of Faculty of Geotechnical Engineering - Theses and Dissertations

zir.nsk.hr

dabar
DIGITALNI AKADEMSKI ARHIVI I REPOZITORIJI

SVEUČILIŠTE U ZAGREBU
GEOTEHNIČKI FAKULTET

FILIP STRELEC

RAZVOJ ALGORITMA ZA PRORAČUN SLIJEGANJA JEDNOLIKO OPTEREĆENE PRAVOKUTNE POVRŠINE

DIPLOMSKI RAD

VARAŽDIN, 2020.

Sazivam članove ispitnog povjerenstva

za _____ u _____ sa

Obranu ovog rada kandidat će vršiti i pred

ispitnim povjerenstvom u Varaždinu

Varaždin, _____

Predsjednik
ispitnog povjerenstva:
Izv. prof. dr. sc. Igor Petrović

Članovi povjerenstva

1) Izv. prof. dr. sc. Krešo Ivanušić
2) Izv. prof. dr. sc. Igor Petrović
3) Doc. dr. sc. Mario Gazdek

SVEUČILIŠTE U ZAGREBU
GEOTEHNIČKI FAKULTET

DIPLOMSKI RAD

RAZVOJ ALGORITMA ZA PRORAČUN SLIJEGANJA JEDNOLIKO OPTEREĆENE PRAVOKUTNE POVRŠINE

Kandidat:
FILIP STRELEC, univ.bacc.ing.amb.



Mentor:
Izv. prof. dr.sc. KREŠO IVANDIĆ

VARAŽDIN, 2020.

Sveučilište u Zagrebu
Geotehnički fakultet

ZADATAK ZA DIPLOMSKI RAD

Pristupnik: FILIP STRELEC

Matični broj: 250 - 2018./2019.

Smjer: GEOINŽENJERSTVO OKOLIŠA

NASLOV DIPLOMSKOG RADA:

RAZVOJ ALGORITMA ZA PRORAČUN SLIJEGANJA JEDNOLIKO OPTEREĆENE PRAVOKUTNE POVRŠINE

Rad treba sadržati:
1. Uvod
2. Općenito o slijeganju
3. Načini proračuna slijeganja
4. Proračuni slijeganja ispod plitkog temelja
5. Izračun dodatnog naprezanja Steinbrennerovom metodom
6. Izvedba i vizualizacija proračuna slijeganja
7. Korištenje programa
8. Prostor za unaprjeđenje rada
9. Doprinos pri izradi rada
10. Zaključak
11. Literatura
12. Popis slika

Pristupnik je dužan predati mentoru jedan uvezen primjerak diplomskog rada sa sažetkom. Vrijeme izrade diplomskog rada je od 45 do 90 dana.

Zadatak zadan: 11.05.2020. Rok predaje: 07.09.2020.

Mentor:
Izv.prof.dr.sc. Krešo Ivandić

REPUBLIKA HRVATSKA - GEOTEHNIČKI FAKULTET - SVEUČILIŠTE U ZAGREBU - VARAŽDIN

Predsjednik Odbora za nastavu:
Izv.prof.dr.sc. Igor Petrović

IZJAVA O AKADEMSKOJ ČESTITOSTI

Izjavljujem i svojim potpisom potvrđujem da je diplomski rad pod naslovom:

RAZVOJ ALGORITMA ZA PRORAČUN SLIJEGANJA JEDNOLIKO OPTEREĆENE PRAVOKUTNE POVRŠINE

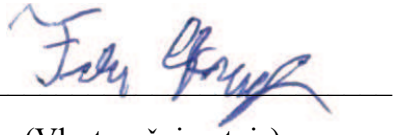
rezultat mog vlastitog rada koji se temelji na istraživanjima te objavljenoj i citiranoj literaturi te je izrađen pod mentorstvom izv. prof. dr.sc. Kreše Ivandića.

Izjavljujem da nijedan dio rada nije napisan na nedozvoljen način, odnosno da je prepisan iz necitiranog rada te da nijedan dio rada ne krši bilo čija autorska prava. Izjavljujem također, da nijedan dio rada nije iskorišten za bilo koji drugi rad u bilo kojoj drugoj visokoškolskoj, znanstvenoj ili obrazovnoj ustanovi.

U Varaždinu, 02.09.2020.

Filip Strelec, univ.bacc.ing.amb

(Ime i prezime)



(Vlastoručni potpis)

IZJAVA MENTORA O POSTOTKU SLIČNOSTI ZAVRŠNOG RADA S VEĆ OBJAVLJENIM RADOVIMA

Izjavljujem i svojim potpisom potvrđujem da je diplomski rad pod naslovom:

RAZVOJ ALGORITMA ZA PRORAČUN SLIJEGANJA JEDNOLIKO OPTEREĆENE PRAVOKUTNE POVRŠINE

pregledan anti-plagijat programskim paketom PlagScan te da postotak sličnosti cjelovitog završnog rada, s već objavljenim radovima, ne prelazi 20%, kao i da pojedinačni postotak sličnosti završnog rada sa svakom literaturnom referencom pojedinačno ne prelazi 5%.

U Varaždinu, 01.09.2020.

Izv.prof.dr.sc.Krešo Ivandić
(Mentor)

(Vlastoručni potpis)

SAŽETAK

Tema diplomskog rada je razvoj algoritma za proračun slijeganja jednoliko opterećene pravokutne površine. Izvedba i vizualizacija proračuna implementirana je putem Unity okruženja. U diplomskom radu objašnjeni su načini proračuna slijeganja i Steinbrennerova metoda čija je implementacija putem C# programskog jezika korištena za izvedbu i vizualizaciju proračuna slijeganja u Unity okruženju. Program podržava izvođenje na računalima i mobilnim uređajima s Android ili iOS operativnim sustavom.

Ključne riječi: dodatna naprezanja, plitki temelj, jednoliko opterećenje, Steinbrenner, slijeganje, Unity, C#

ABSTRACT

The topic of the thesis is the development of an algorithm for the calculation of soil settlement for a uniformly loaded rectangular surface. The implementation of the algorithm and the visualization of the solution were done in the Unity environment. The thesis explains the methods of settlement calculation and Steinbrenner's method, the implementation of which was done by using C # programming language in order to perform and visualize settlement calculations in the Unity environment. The program was supported by computer operative systems and mobile devices with an Android or an iOS operating system.

Key words: surcharge load, shallow foundation, uniform load, Steinbrenner, soil settlement, Unity, C#

SADRŽAJ

1. UVOD 1
2. OPĆENITO O SLIJEGANJU 2
2.1. Dodatna naprezanja u tlu 2
2.2. Slijeganje tla 3
3. NAČINI PRORAČUNA SLIJEGANJA 6
3. 1. Koncentrirana sila na površini linearno-elastičnog poluprostora 6
3. 2. Slijeganje uslijed kontinuiranog opterećenja na kružnoj temeljnoj plohi izotropnog elastičnog poluprorstora 8
3. 4. Početno i konsolidacijsko slijeganje temelja na površini potpuno saturiranog poroznog elastičnog poluprostora 10
4. PRORAČUNI SLIJEGANJA ISPOD PLITKOG TEMELJA 11
4.1. Slijeganje plitkog temelja (Mayne, Poulos) 11
4.2. Tradicionalni proračun slijeganja plitkih temelja 16
5. IZRAČUN DODATNOG NAPREZANJA STEINBRENNEROVOM METODOM 18
6. IZVEDBA I VIZUALIZACIJA PRORAČUNA SLIJEGANJA 21
6.1. Programski alati korišteni za izvedbu i vizualizaciju proračuna 21
6.1.1. Unity – okruženje za razvoj i vizualizaciju problema 21
6.1.2. Programski jezik C# 23
6.2. Provedba i vizualizacija izračuna 23
6.2.1. Unos parametara mjerenja 23
6.2.2. Definiranje parametara slojeva 25
6.2.3. Odabir koordinate izračuna 27
6.2.4. Izračun Steinbrenner-ovog utjecajnog faktora *I* za zadanu koordinatu 27
6.2.4.1. Implementacija Steinbrennerove metode u C# programski jezik 27
6.2.4.2. Izračun utjecajnog faktora *I* za točku odabranih koordinata principom *superpozicije* 29
6.2.5. Provedba i vizualizacija izračuna dodatnog naprezanja 38
6.2.6. Provedba i vizualizacija izračuna relativnih deformacija 39
6.2.7. Provedba i vizualizacija izračuna slijeganja tla 40
7. KORIŠTENJE PROGRAMA 42
7.1. Upute za korištenje programa 43

7.1.1. Unos parametara mjerenja 43
7.1.2. Unos raspona dubine i modula stišljivosti slojeva 44
7.1.3. Unos koordinate izračuna 45
7.2. Korisničko sučelje i dodatne mogućnosti programa 46
7.2.1. Tipkovničke kratice i navigacija kamere 46
7.2.2. Korisničko sučelje 47
7.2.3. Dodatne mogućnosti programa 48
8. PROSTOR ZA UNAPRIJEĐENJE RADA 49
8.1. Unaprjeđenje iskoristivosti programa 50
8.2. Unaprjeđenje mogućnosti izračuna programa 50
9. DOPRINOS PRI IZRADI RADA 52
10. ZAKLJUČAK 52
10. LITERATURA 53
POPIS SLIKA 54
POPIS TABLICA 55

1. UVOD

Proračun slijeganja plitkih temelja oduvijek je bio interesantan problem upravo zbog njegove paradoksalnosti. U praktičkoj izvedbi relativno ga je jednostavno riješiti, no zbog otežane mogućnosti točnog utvrđivanja geometrijskih karakteristika i profila tla, pouzdanost rezultata slijeganja temelja često zna biti upitna. Otežanost u odabiranju reprezentativnih parametara mehaničkih svojstava tla na određenoj lokaciji terenskih zahvata pridodaje smanjenju točnosti krajnjeg izračuna slijeganja tla. Cilj rada bio je razviti kvalitetan program koji se može koristiti i na lokaciji terenskih zahvata putem mobilnog uređaja, a da točno i brzo pruža uvid u slijeganja definirane temeljne plohe, i da pritom bude lako razumljiv za korisnika.

Prvi dio rada objašnjava što je slijeganje i načine proračuna slijeganja (poglavlja 2.-5.) Drugi dio rada posvećen je implementaciji Steinbrennerove metode u C# i vizualizaciji rješenja u Unity okruženju.

Programski kod dostupan je na GitHub-u na adresi:
https://github.com/filip4428/soil-settlement-under-load_Unity

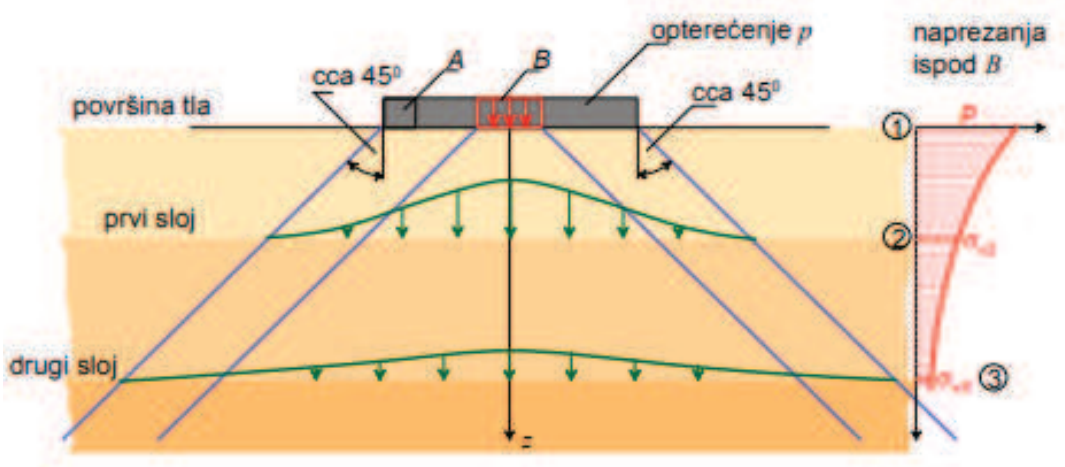
Program se može isprobati u web pregledniku, na adresi
https://filip4428.github.io/Soil-settlement-WebGL/

2. OPĆENITO O SLIJEGANJU

2.1. Dodatna naprezanja u tlu

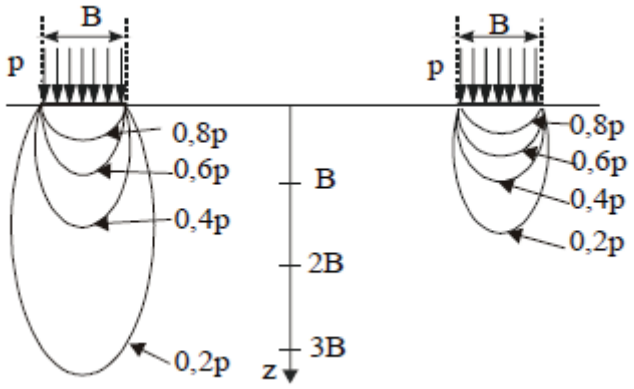
Opterećenjem ili rasterećenjem temeljnog tla uglavnom dolazi do promjene dodatnih naprezanja u tlu što najčešće uzrokuje deformaciju tla. U tlu uvijek postoje početna naprezanja, kojima se potom pribrojavaju dodatna naprezanja kako bi se dobilo konačno stanje naprezanja. Dodatna naprezanja nastaju uslijed dodatnog opterećenja npr. nanesenog građevinom pri čemu dolazi do slijeganja tla. Najčešće se procjenjuju samo vertikalna naprezanja budući da su velik udio svih dodatnih naprezanja upravo vertikalna naprezanja, uz idealizaciju tla linearno-elastičnim modelom, zajedno s parametrima stišljivosti tla. Radi pojednostavljenja izračuna slijeganja, horizontalne deformacije često se zanemaruju. To vrijedi u slučaju samo kada su opterećenja dovoljno manja, dakle nekoliko puta manja od opterećenja pri kojem bi došlo do sloma tla pod temeljem, tj. od nosivosti tla pod temeljem.

Raspodjela naprezanja u tlu prikazana je na slici 1.



Slika 1. *Širenje dodatnih naprezanja u tlu (Kvasnička, Domitrović, 2007)*

Po horizontalnom presjeku tla, dodatna vertikalna naprezanja (σ_v) opadaju, a raspodjela naprezanja se širi sa povećanjem dubine presjeka. Dodatna naprezanja uzrokovana kontinuiranim opterećenjem temeljne plohe manje opadaju sa dubinom u odnosu na naprezanja uzrokovana linijskim opterećenjem. Razlika raspodjele naprezanja vidljiva je na slici 2.



Slika 2. *Raspodjela dodatnih naprezanja tijekom (a) linijskog opterećenja, (b) kvadratne plohe (Roje-Bonacci, 2007)*

2.2. Slijeganje tla

Slijeganje je deformacija temeljnog tla uslijed promjene (porasta) naprezanja u vertikalnom smjeru u tlu. Najčešće je slijeganje tla uzrokovano dodatnim opterećenjem, no pojavljuje se i uslijed različitih dinamičkih efekata poput puzanja, smanjenja razine podzemne vode i sl. Mehanizam slijeganja zasićenog tla i važnost istiskivanja vode prvi je uočio i razradio Terzaghi još 1925. godine. Promjenom opterećenja mijenja se i unutrašnja slika naprezanja i sam volumen tla. Uzrok promijene volumena tla su elastične i plastične deformacije. Do elastičnih deformacija dolazi uslijed promjene volumena čestica tla. Plastične deformacije uglavnom se javljaju promjenom volumena u porama tla.

U nekoherentnim tlima, gdje je velika vodopropusnost, slijeganje nastaje odmah po opterećenju. Porastom opterećenja raste i brzina istjecanja vode iz pora pa u takvim tlima smanjenje pornog tlaka vremenski ne zaostaje za porastom opterećenja. Kod koherentnih tla, gdje su koeficijenti

vodopropusnosti mali, slijeganje postaje ovisno o vremenu i naziva se proces konsolidacije (proces slijeganja u vremenu).

Ukupno slijeganje (s_t) se može podijeliti na tri vremenski zavisne komponente:

$$s_t = s_i + s_c + s_s \quad (1)$$

*Trenutno slijeganje (**s_i**)* je brzo i nastaje odmah nanošenjem opterećenja Svo opterećenje preuzima voda i ne dolazi promjene volumena čestica tla (nedrenirano stanje). Ovo slijeganje kod zasićenih tla nije odraz istiskivanja vode iz pora i nije vezano za povećanje efektivnih naprezanja dok je kod nekoherentnih tla to najizraženija komponenta.

*Primarno konsolidacijsko slijeganje (**s_c**)* ima najveći udio u ukupnom slijeganju i brzo se odvija. Tijekom primarne konsolidacije porni tlak pada na nulu zbog istjecanja vode iz tla (drenirano stanje). Primarna konsolidacija završava kad efektivna naprezanja u potpunosti preuzmu opterećenje. Kod zasićenih tla poput gline, taj proces je izrazito spor.

Sekundarno konsolidacijsko slijeganje (**s_s**) predstavlja veoma sporu promjenu volumena tokom dužeg perioda bez povećanja efektivnih naprezanja u tlu. Ova pojava izazvana je deformacijom pri konstantnom opterećenju, a naziva se puzanje tla. Puzanje tla karakteristično je za koherentna tla je i smatra se da je posljedica deformacije samih čestica.

Zasićena, vezana tla mogu se slijegati samo otjecanjem porne vode jer su čestice tla i voda praktično nestlačivi. Stlačivost odnosno stišljivost temeljnog tla opisuje se modulom stišljivosti *Ms* koji ovisi uglavnom o naponu u tlu tj. za svaku vrijednost modula stišljivosti mora se definirati pripadno područje naprezanja. Vrijednosti modula stišljivosti navedene u tablicama 1. i 2. mogu se uzeti samo kao okvirne veličine za pojedine vrste tla. (Gotić, Gotić 2000)

Tablica 1. *Vrijednosti modula stišljivosti Ms u nevezanim tlima*

Vrsta tla	Gustoća	Oblik zrna	Modul stišljivosti *Ms* [MN/m²]
Pijesak	rahli	okrugli	20 - 50
	rahli	šiljasti	40 - 80
	srednje gusti	okrugli	50 - 100
	srednje gusti	šiljasti	50 - 150
	gusti	šiljasti	150 - 250
Šljunak	srednje gusti		100 - 200

Tablica 2. *Vrijednosti modula stišljivosti Ms u vezanim tlima*

Vrsta tla	Stanje konzistencije	Modul stišljivosti *Ms* [MN/m²]
Pijesak	polučvrsto	5,0 – 10,0
	tvrdo	2,5 – 5,0
	mekano	1,0 – 2,5
Ilovača	polučvrsto	5,0 – 20,0
	mekano	4,0 – 8,0
Prah		3,0 – 10,0

Slijeganje je neizbježno, ali se ono u stišljivom tlu može dopustiti ako ne ugrožava funkcionalnost i stabilnost građevine (granično stanje uporabljivosti – Eurokod 7). Ako ti zahtjevi nisu ispunjeni, svojstva temeljnog tla se moraju poboljšati ili se opterećenje mora odvoditi u dublje, nosive slojeve tla. U svjetskoj graditeljskoj praksi za poboljšanje tla sve se više primjenjuje plitko temeljenje jer se pokazalo kao jeftinije rješenje.

3. NAČINI PRORAČUNA SLIJEGANJA

Kod proračuna dodatnih naprezanja u tlu (opterećenje od temelja) usvajaju se pretpostavke o tlu kao izotropnom, homogenom, linearno elastično mediju, zbog potrebe proračuna prognoze slijeganja. Tlo se najčešće idealizira teorijom elastičnosti.

Teorija elastičnosti nam omogućava rješavanje elastičnih diferencijalnih jednadžbi za definirane rubne uvjete. U slučaju jednostavnijih rubnih problema, do rezultata se dolazi putem analitičkih rješenja. Za kompliciranije slučajeve primjenjuje se princip *superpozicije*, ili se do rješenja dolazi aproksimativnim numeričkim metodama.

Do mnogih rješenja složenijih problema moguće je doći iz poznatih, temeljnih rješenja. Posebice je značajno Boussinesqovo rješenje za određivanje dodatnih naprezanja u tlu opterećeno različitim oblicima temeljne plohe. Najpoznatija takva rješenja za jednoliko opterećenje na površini elastičnog homogenog poluprostora su:

- Određivanje opterećenja ispod kružne plohe (Newmark)
- Određivanje opterećenja ispod pravokutnika (Steinbrenner)

Elastični poluprostor nazivamo dio prostora ispod kojeg je linearno elastični materijal a omeđen je horizontalnom ravninom.

3. 1. Koncentrirana sila na površini linearno-elastičnog poluprostora

Gornjom idealizacijom tla elastičnim poluprostorom omogućuje definiranje tla s Youngovim *E* i Poissonovim koeficijentom *v*. U tom slučaju, sloj tla je relativno velike debljine u odnosu na veličinu opterećene površine, a parametri stišljivosti se ne mijenjaju po dubini. (Kvasnička, Domitrović, 2007) Slijeganje se tada može izračunati izravno na temelju Boussinesqovog rješenja (1885) za djelovanje koncentrirane sile na površini homogenog elastičnog poluprostora (slika 2) prema izrazu:

$$s_p = \frac{(1-v^2)\cdot P}{\pi \cdot E \cdot r} \tag{2}$$

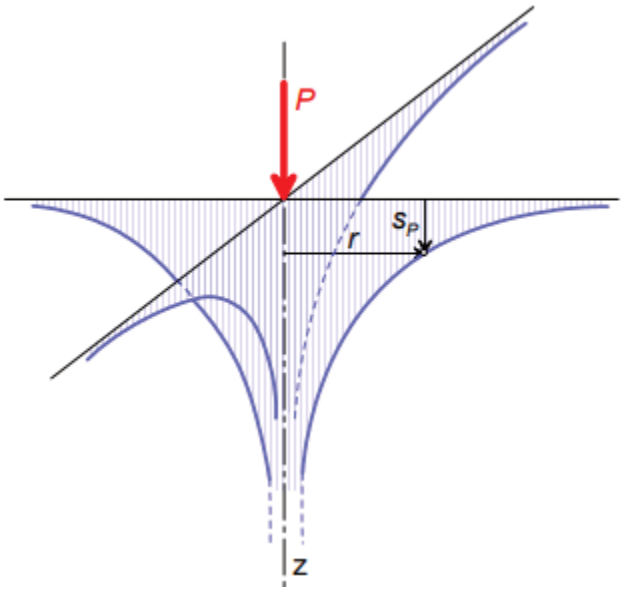
Gdje je:

P: veličina koncentrirane sile

r: horizontalna udaljenost od sile do točke u kojoj se izračunava slijeganje

E: Youngov modul elastičnosti

v: Poissonov koeficijent tla



***Slika 3**. Aksonometrijski prikaz slijeganja ispod koncentrirane sile na elastičnom poluprostoru (Kvasnička, Domitrović, 2007)*

Izraz (2) daje točne vrijednosti slijeganja, ali prema njemu slijeganja za $r = 0$ su beskonačno velika tj. rješenje ima singularnu točku na mjestu djelovanja koncentrirane sile gdje naprezanja i pomaci teže u beskonačnost (Szavits-Nossan, 2013).

3. 2. Slijeganje uslijed kontinuiranog opterećenja na kružnoj temeljnoj plohi izotropnog elastičnog poluprostora

Temeljem Boussinesqova rješenja, Newmark (1935) je izveo rješenje za dodatno naprezanje ispod kružno opterećene temeljne ploče na površini izotropnog elastičnog poluprostora.

Za kružnu površinu, veličine F, opterećenu kontinuiranim opterećenjem q, radijusa R, slijeganje se izračunava kao integral jednadžbe (Kvasnička, Domitrović, 2007):

$$s_{qR} = \iint_{(F)} \frac{(1-\nu^2)\cdot q \cdot dF}{\pi \cdot E \cdot r} \tag{3}$$

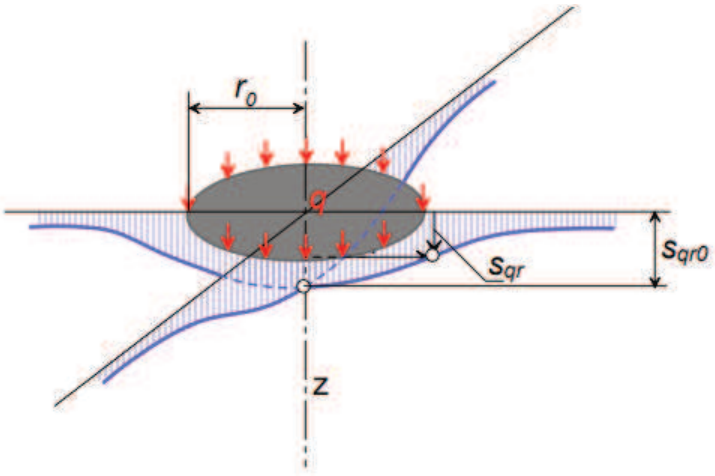
gdje je dF element kružno opterećene površine jednolikim opterećenjem q (slika 3).

Na temelju rješenja iz gornje jednadžbe može se odrediti slijeganje za bilo koju točku na horizontalnoj udaljenosti r od središta kružne plohe prema izrazu:

$$s_{qR} = \frac{q \cdot R \cdot I_r}{E} \tag{4}$$

Na temelju (4), središte plohe računa se prema izrazu:

$$s_{qRo} = \frac{2\cdot(1-\nu^2)\cdot q \cdot R}{E} \tag{5}$$



Slika 4. *Aksonometrijski prikaz slijeganja ispod kružno opterećene površine (Kvasnička, Domitrović, 2007)*

3. 3. Slijeganje ispod ruba pravokutno opterećene temeljne plohe

Temeljem Boussinesqova rješenja, Steinbrenner (1934) je izveo rješenje za dodatno opterećenje ispod ugla pravokutno opterećene plohe. Za bilo kakvu pravokutnu plohu vrijedi pravilo *superpozicije*, kojim se osnovna ploha razdjeljuje na manje plohe od kojih se računa utjecaj u točki od interesa. Za svaku odabranu dubinu (z) odrede se koeficijenti $I\sigma$ koji se potom očitavaju iz dijagrama u odnosima l/b i z/b te se proračun ponavlja za potrebne dubine.

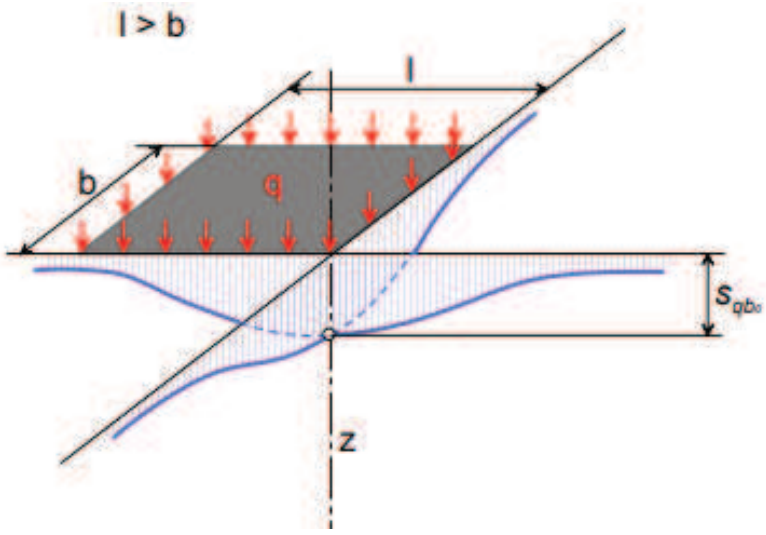
Slijeganje ispod ugla pravokutne plohe (slika 4), dimenzija $l \times b$ ($l > b$) izračunava se prema:

$$s_{qbo} = \frac{q \cdot b \cdot (1-\nu^2) \cdot I_b}{E} \tag{6}$$

I_b, je koeficijent koji je odredi Steinbrenner, a njegova vrijednost proizlazi iz izraza:

$$I_b = \left\{ \frac{l}{b} \ln \frac{1+\sqrt{\left(\frac{l}{b}\right)^2+1}}{\left(\frac{l}{b}\right)} + \ln\left[\left(\frac{l}{b}\right) + \sqrt{\left(\frac{l}{b}\right)^2+1}\right] \right\} \tag{7}$$

Ovaj je izraz primjenljiv za razne tipove nekoherentnih materijala kao što su pijesak i šljunak, no u proračunima za kruti temelj treba uzeti 7% manji koeficijent.



Slika 5. *Aksonometrijski prikaz slijeganja ispod ugla pravokutno opterećene površine (Kvasnička, Domitrović, 2007)*

3. 4. Početno i konsolidacijsko slijeganje temelja na površini potpuno saturiranog poroznog elastičnog poluprostora

Parametri tla za nedrenirano stanje su E_t i v i razlikuju se od parametara tla u dreniranom stanju E' i v'. Ponašanje tla s vodom, kad nije omogućeno dreniranje, određuje nestišljiva voda u porama. Tlo je tada potpuno zasićeno, promjena vertikalne deformacije jednaka je sumi bočnih deformacija.

Između dva para parametara (E_t, $v = 0.5$) i (E', v') se može uspostaviti veza na temelju činjenice da voda ne može preuzeti posmik tj. da je modul posmika (G) svojstvo samo čvrstih čestica i u nedreniranom i dreniranom stanju, tj, da je G = G'. Prema teoriji elastičnosti je (Kvasnička, Domitrović, 2007):

$$G = \frac{E}{2(1+v)'} \tag{8}$$

G = G' te slijedi:

$$\frac{E_t}{2(1+v)} = \frac{E'}{2(1+v')} \tag{9}$$

a uz ν = 0.5 slijedi:

$$E_t = \frac{3 \cdot E'}{2(1+v')} \tag{10}$$

Na temelju gornjeg izraza može se dobiti odnos početnog i konsolidacijskog slijeganja za, primjerice, središte kružne plohe iz izraza (5):

$$s_{qRo} = \frac{2 \cdot (1-v^2) \cdot q \cdot r_0}{E} \tag{11}$$

U slučaju kada nema promjene volumena ($v = 0.5$), početno slijeganje je onda:

$$s_{inst} = \frac{2(1-0.5^2) \cdot q \cdot r_0}{E_t} = \frac{1.5 \cdot q \cdot r_0}{E_t} = \frac{2}{3} q \cdot r_0 \frac{2 \cdot (1-v')}{3 \cdot E'} \tag{12}$$

Konsolidacijsko slijeganje je povezano s efektivnim parametrima:

$$s_{const} = \frac{2\cdot(1-v^2)\cdot q\cdot R}{E\prime} \tag{13}$$

Pa onda odnos dvaju slijeganja ovisi samo o vrijednosti Poissonovog koeficijenta:

$$\frac{s_{inst}}{s_{const}} = \frac{1+v}{2(1-v^2)} = \frac{1}{2(1-v)}\ ; za\ v = 0.2 \rightarrow \frac{1}{1.6}; za\ v = 0.3 \rightarrow \frac{1}{1.4} \tag{14}$$

Prema teoriji elastičnosti, za uobičajene vrijednosti Poissonovog koeficijenta v (0.2 – 0.3) konsolidacijsko slijeganje je 40% - 60% veće od početnog.

4. PRORAČUNI SLIJEGANJA ISPOD PLITKOG TEMELJA

4.1. Slijeganje plitkog temelja prema teoriji elastičnosti

Prema Eurokodu 7, slijeganje plitkog temelja opterećenog vertikalnom silom koja je manja od nosivosti tla određuje se prema izrazu:

$$s = \frac{pBI}{E_m} = \frac{p}{k} \tag{15}$$

gdje je:

p - prosječni kontaktni pritisak između temelja i tla

B - širina temelja

I - utjecajni koeficijent slijeganja ovisan o obliku temelja, debljini ukopavanja i debljini sloja tla

E_m - odgovarajući (ekvivalentni) modul elastičnosti

Veličina *k* predstavlja krutosti temeljnog tla i računa se prema izrazu:

$$k = \frac{p}{s} = \frac{E_m}{BI} \tag{16}$$

Davis i Poulos (David, Poulus, 1968) razmatrali su primjenu izraza teorije elastičnosti za proračun slijeganja plitkih temelja, uz laboratorijsko određivanje ekvivalentnog modula elastičnosti za raspon naprezanja u tlu. Laboratorijsko određivanje modula elastičnosti posebno vrijedi za nekoherentna tla. Uz pretpostavke teorije elastičnosti izvedeni su brojni izrazi za pomake elastičnih tijela podvrgnutih opterećenju na njihovoj površini ili unutrašnjosti. Najjednostavniji takav primjer je onaj jednoliko raspodijeljenog normalnog opterećenja na kružnoj plohi na površini izotropnog i homogenog elastičnog poluprostora. Za potpuno savitljiv kružni temelj promjera *d* na takvom poluprostoru s modulom elastičnosti *E* i Poissonovim omjerom *v*, krutost temeljnog tla *k* za slijeganje u središtu kružnog površinskog opterećenja dana je izrazom (Lukić Kristić, Szavits-Nossan, Mišćević, 2017):

$$k = \frac{E}{(1-\nu^2)d} \tag{17}$$

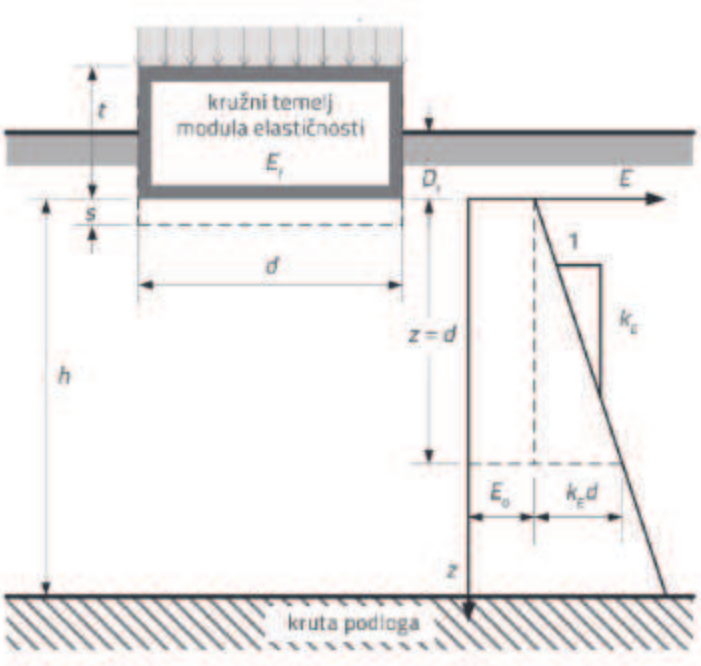
Kod apsolutno krutog kružnog temelja, izraz za krutost *k* je sljedeći:

$$k = \frac{4}{\pi}\frac{E}{(1-\nu^2)d} \tag{18}$$

Mayne i Poulos (Mayne, Poulos, 1999) dali su izraz za puno općenitiji slučaj nehomogeno, izotropnog, linearno elastičnog tla, gdje modul elastičnosti linearno raste s dubinom tla *z*. Na razini temeljne plohe je E_0, a ispod razine temeljne plohe raste s koeficijentom k_E. Tako dobivamo izraz za modul elastičnosti:

$$E = E_0 + k_E z \tag{19}$$

Ovakva tla zovu se tla Gibsonove vrste. Slijeganje takvog tipa tla prikazano je na slici 6.



Slika 6. *Slijeganje kružnog temelja ukopanog u sloj tla Gibsonove vrste (Lukić Kristić, Szavits-Nossan, Mišćević, 2017)*

Prema tome, slijeganje ispod središta savitljivog kružnog temelja na tlu Gibsonove vrste dano je izrazom:

$$s = \frac{pdI_G I_F I_E (1-\nu^2)}{E_0} = \frac{p}{k} \tag{20}$$

Ovaj se izraz može, uz malu pogrešku, koristiti i za pravokutni temelj površine A, kojemu stranice nisu previše različite. Za pravokutni temelj površine A proračunava se ekvivalentni kružni temelj promjera d prema izrazu:

$$d = \sqrt{\frac{4A}{\pi}} \tag{21}$$

Faktor I_G je faktor nehomogenosti i ovisi o parametru β i omjeru h/d, gdje je:

$$\beta = \frac{E_0}{k_E d} \tag{22}$$

te je približan izraz za faktor I_G:

$$I_G \approx \frac{1.6\frac{h}{d}}{\left(1+\frac{0.6}{\beta^{0.8}}\right)\left(1+1.6\frac{h}{d}\right)} \tag{23}$$

I_F je faktor krutosti elastičnog temelja, te ovisi o njegovoj debljini t i njegovom modulu elastičnosti E_f i dan je izrazom:

$$I_F = \frac{\pi}{4} + \frac{1}{\frac{1}{1-\frac{\pi}{4}}+10\left(\frac{E_f}{E_0+\frac{1}{2}k_E d}\right)\left(\frac{2t}{d}\right)^3} \tag{24}$$

I_E je faktor ukopanosti temelja čija se približna vrijednost računa slijedećim izrazom:

$$I_E \approx 1 - \frac{1}{3.5\exp(1.22\nu-0.4)\left(\frac{d}{D_f}+1.6\right)} \tag{25}$$

Mayne i Poulosova (Mayne, Poulos, 1999) metoda pretpostavlja da je podjela dodatnih naprezanja u nehomogenom tlu gotovo identična podjeli dodatnih naprezanja u homogenom poluprostoru. Međutim, ta pretpostavka ne vrijedi za relativne deformacije. U tom se slučaju nehomogeno tlo dijeli na n homogenih slojeva. Za vertikalno slijeganje *i-tog* od ukupnog broja n slojeva zadanih u profilu vrijedi izraz (Szavits-Nossan, 2013):

$$\Delta s_{yi} = s_{ygi} - s_{ydi} \tag{26}$$

gdje je:

s_{ygi}: vertikalni pomak točke na gornjoj granici

s_{ydi}: vertikalni pomak točke na donjoj granici *i-tog* sloja

Proračun ovih veličina, za vertikalna jednolika opterećenja u dreniranom ili nedreniranom stanju provodi se za točke ispod središta kružne temeljne plohe promjera D slijedećim, izrazom:

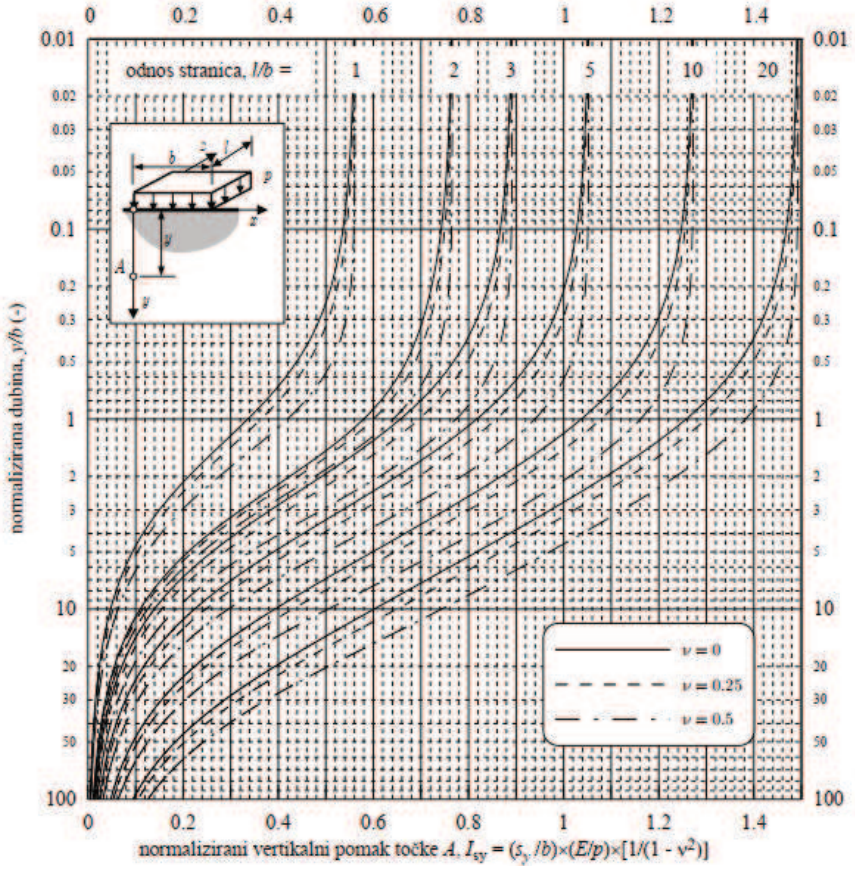
$$\Delta s_{yi} = pD\frac{1-\nu_i^2}{E_i}\Delta I_{syi} \tag{27}$$

gdje je:

$$\Delta I_{syi} = I_{sygi} - I_{sydi} \tag{28}$$

I_{sygi} je utjecajni koeficijent za slijeganje gornje a I_{sydi} donje granice *i-tog sloja*. Kod izračuna slijeganja pravokutne plohe opterećene vertikalnim jednolikim opterećenjem, promjer D potrebno je zamijeniti stranicom pravokutnika b.

Za temeljnu plohu pravokutnog oblika, koja je opterećena jednolikim vertikalnim opterećenjem, veličinu D potrebno je zamijeniti veličinom *b* koja odgovara stranici pravokutnika. Raspodjela veličina I_{sy} po dubini prikazana je na slijedećoj slici.



Slika 7. *Utjecajni koeficijenti I_{sy} za vertikalnih deformacija rubne točke pravokutnog temelja prema Newmarku i Steinbrenneru (Szavits-Nossan, 2013)*

Ukupno slijeganje na površini dobivamo zbrajanjem slijeganja svih *n* slojeva u tlu:

$$s_y(y=0) = \sum_{i=1}^{n} \Delta s_{yi} = \sum_{i=1}^{n} pD \frac{1-\nu_i^2}{E_i} \Delta I_{syi} \tag{29}$$

4.2. Tradicionalni proračun slijeganja plitkih temelja

Izračun deformacija proizlazi iz raspodjele dodatnih naprezanja u tlu, odnosa deformacija i naprezanja i odgovarajućih parametara krutosti svakog sloja u tlu. Nužno je pretpostaviti da su raspodijele dodatnih naprezanja jednake u homogenom i nehomogenom tlu kako bi sljedeći izračuni bili prihvatljivi.

Izraz za raspodjelu slijeganja po dubini ispod koordinate (x_0, y_0) dobije se prema:

$$\varepsilon_y = \frac{ds_y}{dy} \tag{30}$$

gdje je:

$\boldsymbol{\varepsilon}_y$ vertikalni pomak promatrane točke

Prema teoriji elastičnosti je:

$$\epsilon_y(y) = \frac{1}{E'(y)}[\Delta\sigma'_y(y) - \nu'(\Delta\sigma'_x(y) + \Delta\sigma'_z(y))] \tag{31}$$

Pri slučaju kad su dodatna efektivna naprezanja jednaka ukupnim dodatnim naprezanjima tada vrijedi izraz:

$$s_y(y=0) = \int_0^\infty \epsilon_y(y) = \int_0^\infty \frac{1}{E'(y)}[\Delta\sigma'_y(y) - \nu'(\Delta\sigma'_x(y) + \Delta\sigma'_z(y))]\,dy \tag{32}$$

U našem slučaju ovaj se integral integrira numerički do dubine koja se zove „utjecajna dubina“. Utjecajnom dubinom smatramo dubinu pri kojoj je vertikalno dodatno naprezanje manje od 10% efektivnog vertikalnog naprezanja. Za „utjecajnu dubinu“ *d* vrijedi:

$$\Delta\sigma_y/\sigma'_{y0} \leq 0.1 \tag{33}$$

Tada slijeganje površine terena postaje približno:

$$s_y(y=0) = \int_0^d \frac{1}{E'(y)}[\Delta\sigma'_y(y) - \nu'(\Delta\sigma'_x(y) + \Delta\sigma'_z(y))]\,dy \tag{34}$$

Ovaj se izraz može dodatno pojednostaviti pod pretpostavkom da je utjecaj dodatnih bočnih naprezanja zanemariv. U tom slučaju Youngov modul E' jednak je edometarskom modulu E'_{oed} i dobije se:

$$s_y(y=0) \approx \int_0^d \frac{\Delta\sigma'_y(y)}{E'_{oed}(y)} dy \tag{35}$$

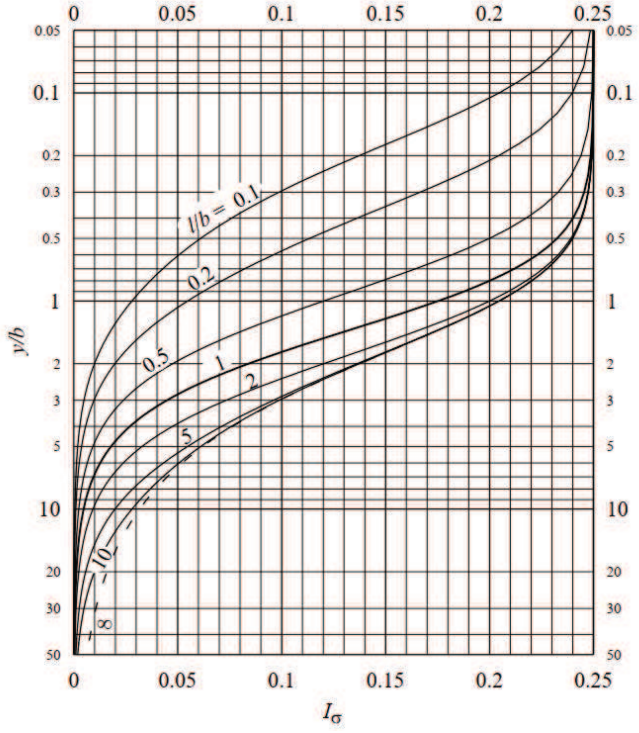
Zbog jednostavnosti ovog izraza on se najčešće upotrebljava u geotehničkoj praksi, međutim zanemaruje bočne deformacije i nije u mogućnosti je provesti točan proračun trenutačnog slijeganja u nedreniranim uvjetima.

5. IZRAČUN DODATNOG NAPREZANJA STEINBRENNEROVOM METODOM

Na osnovi Bousinesqovog rješenja, Steinbrenner (1934) dolazi do rješenja za izračun dodatnog opterećenja u dubini z' ispod ruba pravokutne temeljne plohe dimenzija $B \times L$, s time da je $B < L$ prema:

$$\Delta\sigma_y = p \cdot I_\sigma \tag{36}$$

Utjecajni koeficijent I_σ određuje se prema dijagramu za odgovarajuću dubinu z'/B i odnos L/B prema slici 8:



Slika 8. *Utjecajni koeficijent I_σ za dodatno vertikalno naprezanje ispod vrha pravokutne plohe na površini linearno elastičnog, izotropnog i homogenog poluprostora opterećenog jednolikim vertikalnim opterećenjem (Szavits-Nossan, 2013)*

Rješenje Newmarka **i** Steinbrennera dobije se prema:

$$\sigma_{yy} = \frac{p}{2\pi}\left[\frac{mn(1+m^2+2n^2)}{(1+n^2)(m^2+n^2)\sqrt{1+m^2+n^2}} + arctan\frac{m}{n\sqrt{1+m^2+n^2}}\right] \tag{38}$$

$$\sigma_{xx} = \frac{p}{2\pi}\left\{\begin{matrix}\frac{\pi}{2} - \frac{mn}{(m^2+n^2)\sqrt{1+m^2+n^2}} - arctan\frac{m}{n\sqrt{1+m^2+n^2}} + \\ +(1-2\nu)\left[arctan\frac{1}{m} - arctan\frac{n\sqrt{1+m^2+n^2}}{mn}\right]\end{matrix}\right\} \tag{39}$$

$$\sigma_{zz} = \frac{p}{2\pi}\left\{\begin{matrix}\frac{\pi}{2} - \frac{mn}{(1+n^2)\sqrt{1+m^2+n^2}} - arctan\frac{n\sqrt{1+m^2+n^2}}{m} + \\ +(1-2\nu)\left[\arctan(m) - arctan\frac{m\sqrt{1+m^2+n^2}}{n}\right]\end{matrix}\right\} \tag{40}$$

$$\sigma_{yz} = \frac{p}{2\pi}n^2 m\left[\frac{1}{n^2\sqrt{m^2+n^2}} - \frac{1}{(1+n^2)\sqrt{1+m^2+n^2}}\right] \tag{41}$$

$$\sigma_{xy} = \frac{p}{2\pi}n^2\left[\frac{1}{n^2\sqrt{1+n^2}} - \frac{1}{(m^2+n^2)\sqrt{1+m^2+n^2}}\right] \tag{42}$$

$$\sigma_{xz} = \frac{p}{2\pi}\left\{\begin{matrix}1 - \frac{n}{\sqrt{1+n^2}} - \frac{n}{\sqrt{m^2+n^2}} + \frac{n}{\sqrt{1+m^2+n^2}} + \\ +(1-2\nu)\left[\ln\frac{2n}{n+\sqrt{1+n^2}} + ln\frac{n+\sqrt{1+m^2+n^2}}{n+\sqrt{m^2+n^2}}\right]\end{matrix}\right\} \tag{43}$$

$$s_y = \frac{pb}{E}(1-2\nu)(A - \frac{1-2\nu}{1-\nu}B) \tag{44}$$

$$A = \frac{1}{2\pi}ln\frac{(1+\sqrt{1+m^2+n^2})^{2m}(m+\sqrt{1+m^2+n^2})^2}{(1+n^2)(m^2+n^2)^m}) \tag{45}$$

$$B = \frac{1}{2\pi}arctan\frac{m}{n\sqrt{1+m^2+n^2}} \tag{46}$$

gdje je:

$$m = \frac{L}{B}\ ;\ n = \frac{y}{B}$$

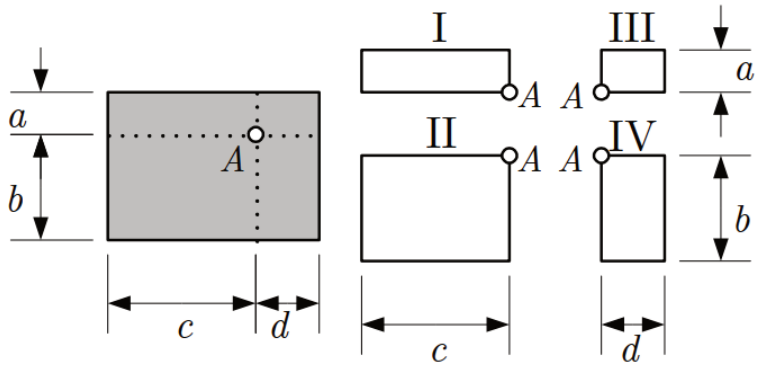
Steinbrennerova formula omogućuje nam izračun dodatnih naprezanja ispod ruba pravokutne temeljne plohe, da bi izračunali dodatna naprezanja ispod bilo koje točke temeljne plohe potrebno je iskoristiti princip superpozicije (Szavits-Nossan, 2013). Razlikujemo dvije situacije:

(a) za točku A, unutar površine pravokutnika

(b) za točku A, izvan pravokutnika.

U situaciji (*a)* prikazanoj na slici 9., pravokutna površina se podijeli na četiri manja pravokutnika i za svaki od njih se odredi utjecajni koeficijent I_σ:

$$f_A(y) = f_A^I(y) + f_A^{II}(y) + f_A^{III}(y) + f_A^{IV}(y) \tag{37}$$

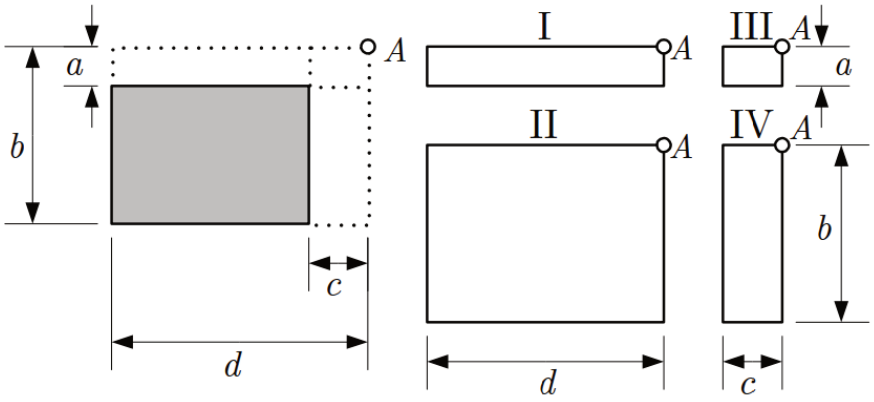


***Slika 9**. Proračun dodatnih naprezanja u točki „ispod“ temelja (Szavits-Nossan, 2013)*

Rješenje može poslužiti za proračun dodatnih naprezanja ispod bilo koje druge točke na površini temeljnoga tla. Zbrajanjem koeficijenata sva četiri temelja superponiramo njihova djelovanja principom superpozicije. Princip superpozicije se može koristiti samo ako je poluprostor linearan i elastičan, što i vrijedi za Boussinesqovo i Steinbrennerovo rješenje.

U situaciji (*b)* se osnovni pravokutnik produlji do točke A (slika 10). Nakon toga postupak je isti kao i kod situacije (a), s time da pravokutnike koji su izvan područja osnovnog pravokutnika oduzimamo (I i IIV):

$$f_A(y) = -f_A^I(y) + f_A^{II}(y) + f_A^{III}(y) - f_A^{IV}(y).$$



Slika 10. *Proračun dodatnih naprezanja u točki „izvan“ temelja (Szavits-Nossan, 2013)*

6. IZVEDBA I VIZUALIZACIJA PRORAČUNA SLIJEGANJA

U samim početcima izrade diplomskog rada stavljena je velika važnost na što veću dostupnost i kompatibilnost razvijenog programa sa što većim brojem elektroničkih uređaja. Upravo je iz tog razloga korišten Unity – okruženje za razvoj softvera (slika 11).

Trenutno poglavlje opisuje alate korištene za implementaciju proračuna slijeganja kao i samu provedbu proračuna slijeganja unutar tih alata.

6.1. Programski alati korišteni za izvedbu i vizualizaciju proračuna

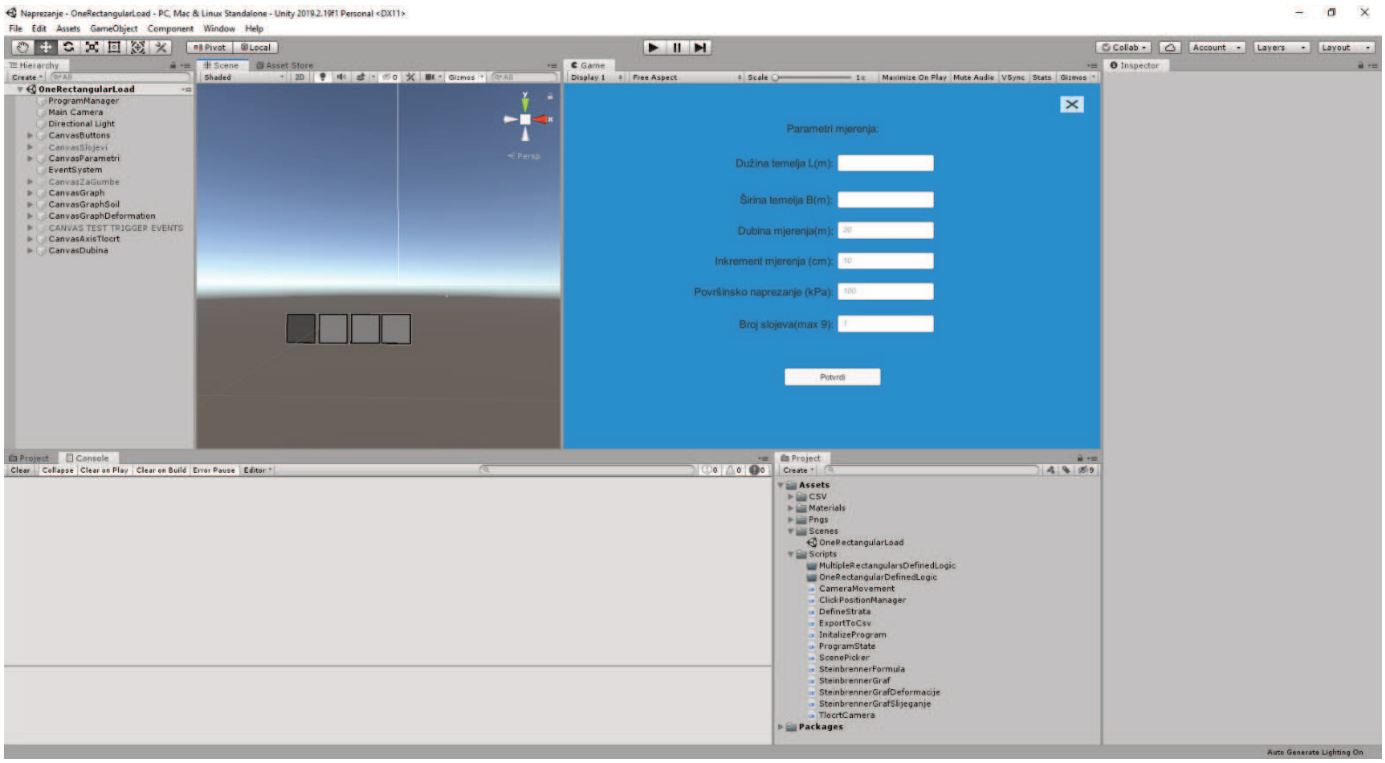
6.1.1. Unity – okruženje za razvoj i vizualizaciju problema

Unity je okruženje za razvoj softvera koji se prvenstveno koristi za izradu 2D i 3D Video igara, no njegova široka dostupnost i kompatibilnost s programskim jezikom C# omogućuje brzu implementaciju i vizualizaciju svakojakih simulacija ili izračuna. Prva inačica Unity programa objavljena je 2005. godine, prvenstveno za Mac OS X, no danas, programe razvijene u Unity okruženju moguće je pokretati na više od 20 platforma. Programe razvijene u Unity okruženju moguće je pokretati na:

- Računalnim operativnim sustavima poput Windows-a, Mac OS-a i Linux-a
- Mobilnim operativnim sustavima poput Android-a i iOS-a
- Unutar svih modernih internetskih preglednika
- Raznim igraćim konzolama
- Raznim drugim platformama

Primjeri korištenja Unity okruženja za vizualizaciju raznih znanstvenih istraživanja:

- Sustav vizualizacije za analizu biomedicinskih i genskih skupova podataka pomoću Unity okruženja (Khalifa, Nguyen, Simoff i Catchpoole, 2015)
- Web-sustav za vizualizaciju simulacije poplava temeljen na Unity3D okruženju za učinkovito upravljanje prirodnim katastrofama (Gang, Ryu, Choi i Choung, 2017)
- Izračun solarno-energetskog potencijala građevina i vizualizacija podataka unutar Unity 3D okruženja (Buyuksalih, Bayburt, Baskaraca, Karim i Rahman, 2017)



Slika 11. *Unity – okruženje za razvoj i vizualizaciju programa*

6.1.2. Programski jezik C#

C# (C Sharp) je objektno-orijentiran programski jezik opće namjene (Ecma standard, 2017). Razvijen je 2000. godine od strane Microsofta kao dio .NET inicijative te kasnije odobren kao internacionalni standard od strane Ecma-e 2002. godine i ISO-a 2003. godine

Ecma standard navodi ove ciljeve dizajna za C# programski jezik (Ecma standard, 2017):

- C# je namijenjen da bude jednostavan, moderan, objektno-orijentiran programski jezik opće namjene
- Jezik, i njegove implementacije bi trebale pružiti podršku za glavne programsko inženjerske principe poput *„strong type checking“*, *„array bounds checking“*, detekcija pokušaja korištenja varijable koja nije inicijalizirana, automatsko upravljanje memorijom.
- Jezik je namijenjen za korištenje u CBD *(„components-based development“)* načinu razvoja softverski komponenata pogodnih za primjenu u distribuiranim okruženjima.
- Podrška za internacionalizaciju i lokalizaciju je vrlo važna
- C# je namijenjen da bude prikladan za korištenje u velim sustavima koji koriste sofisticirane operativne sustave ali ga je moguće i koristiti na malim *„embeded system“* komponentama

6.2. Provedba i vizualizacija izračuna

6.2.1. Unos parametara mjerenja

Početni ekran učitanog programa daje korisniku mogućnost definiranja sljedećih parametara mjerenja: Dužina temelja L (m), Širina temelja B (m), Dubina mjerenja(m), Inkrement mjerenja (cm), Površinsko naprezanje (kPa) i broj slojeva.

Unosom sljedećih parametara i pritiskom na gumb potvrdi program provjerava jesu li svi parametri brojčanog tipa te da li je korisnik definirao parametre koji nisu pred definirani poput dužine temelja i širine temelja.

```
... //dobivanje unosa iz polja za unos podataka

InputField sirinaInputB =
GameObject.Find("InputFieldSirinaB").GetComponent<InputField>();
InputField duzinaInputL =
GameObject.Find("InputFieldDuzinaL").GetComponent<InputField>();
InputField dubinaInputParametar =
GameObject.Find("InputFieldDubinaZ").GetComponent<InputField>();
InputField brojMjerenjaInputField =
GameObject.Find("InputFieldBrojMjerenja").GetComponent<InputField>();
InputField povrsinskoNaprezanje =
GameObject.Find("InputFieldPovrsinskoNaprezanje").GetComponent<InputField>();
 InputField brojSlojevaInput =
GameObject.Find("InputFieldBrojSlojeva").GetComponent<InputField>()

    //definiranje paramentara u programState-u

programState.inkrementMjerenjaZ = 0.1f;
programState.dubinaZ = 20f;
programState.povrsinskoNaprezanje = povrNapr;
programState.brojSlojeva = brojSlojeva;
 bool canConvertSirina = double.TryParse(sirinaInputB.text, out sirinaTemelja);
programState.sirinaB = sirinaTemelja;
bool canConvertDuzina = double.TryParse(duzinaInputL.text, out duzinaTemelja);
programState.duzinaL = duzinaTemelja;
bool canConvertDubina = double.TryParse(dubinaInputParametar.text, out
dubinaMjerenjaParametar);
if (canConvertDubina){
      programState.dubinaZ = dubinaMjerenjaParametar ;
         }
bool canConvertBrojMjerenja = double.TryParse(brojMjerenjaInputField.text, out
brojMjerenja);
```

```
if (canConvertBrojMjerenja){
      programState.inkrementMjerenjaZ =  brojMjerenja*0.01;
          }
bool canConvertPovNapr = double.TryParse(povrsinskoNaprezanje.text, out povrNapr);
if (canConvertPovNapr){
      programState.povrsinskoNaprezanje =  povrNapr;
         }
bool canConvertBrojSlojeva = Int32.TryParse(brojSlojevaInput.text, out brojSlojeva);
if (canConvertBrojSlojeva){
      programState.brojSlojeva =  brojSlojeva;
          }
```

6.2.2. Definiranje parametara slojeva

Unosom broja slojeva i uspješnim unosom drugih parametara mjerenja pojavljuje se ekran za definiranje parametara slojeva. Svakom sloju potrebno je definirati raspon dubine (m) i modul stišljivosti *Ms* (MPa). Program automatski definira gornju granicu sloja na temelju prošlog sloja. Gornja granica za prvi sloj automatski je postavljena na 0 metara. Uspješnim unosom parametara slojeva i pritiskom na gumb potvrdi započinje inicijalizacija funkcije *ConfirmStrata*() čija je glavna zadaća spremanje raspona svakog pojedinog sloja u varijablu *slojeviArray* tipa „*Array of doubles*“ i spremanje modula stišljivosti pojedinog sloja u varijablu *modulStisljivostiArray* tipa „*Array of doubles*“ koji se nalaze u klasi *ProgramState*.

```
public void ConfirmStrata()
   {

//referenca na ProgramState klasu

programManager = GameObject.Find("ProgramManager");
programState = programManager.GetComponent<ProgramState>();
// definiranje potrebnih varijabli
 bool everythingFine = true;
 programState.youngDefined = everythingFine;
```

```
 brojSlojeva  = programState.brojSlojeva;

//Inicijalizacija varijabla slojeviArray i modulStisljivostiArray

programState.slojeviArray = new double[programState.brojSlojeva];
programState.modulStisljivostiArray = new double [programState.brojSlojeva];

//For petlja za upis definiranih parametara u zadane varijable

for (int i = 0; i<brojSlojeva; i++){

  bool canCnoverSlojevi;
  if (i!=brojSlojeva-1){
        canCnoverSlojevi =
        double.TryParse(GameObject.FindGameObjectWithTag("textStrataInput"+(i+1)).GetCo
        mponent<InputField>().text, out double slojevi);
        programState.slojeviArray[i] =slojevi;
  }
  else{
        canCnoverSlojevi = true;
        programState.slojeviArray[i] =programState.dubinaZ;
  }
  bool canCnovertModulStisljivost = double.TryParse(
  GameObject.FindGameObjectWithTag("textStrataYoungInput"+(i+1)).GetComponent<InputF
  ield>().text, out double young);
  if (everythingFine){
        everythingFine = canCnoverSlojevi;
        programState.youngDefined = everythingFine;
  }
  if (everythingFine){
        everythingFine = canCnovertModulStisljivost;
        programState.youngDefined = everythingFine;
  }
```

*programState.modulStisljivostiArray[i]=young*1000;*

}

}

6.2.3. Odabir koordinate izračuna

Nakon uspješnog definiranja svih potrebnih parametara mjerenja, program iscrtava koordinatni sustav u čijem je ishodištu temelj definiran prijašnjim parametrima. Korisnik ima 3 mogućnosti definiranja koordinate izračuna:

- Upisati koordinatu točke
- Pritisnuti lijevom tipkom miša koordinatu na koordinatnom sustavu
- Odabrati karakterističnu točku temelja

Pritiskom na gumb Izračunaj započinje inicijalizacija funkcije *InitializeSteinbrennerGraph*() koja koristi Steinbrennerovu formulu i princip superpozicije kako bi izračunala Steinbrennerove utjecajne faktore *I* za sve mjerene dubine odabrane koordinate, spremila te podatke u varijablu *steinbrennerRezultatOdabraneTocke* koja se nalazi u klasi *ProgramState*. Završno, funkcija množi utjecajni faktor *I* s površinskim naprezanjem (kPa) i iscrtava graf dodatnih naprezanja po dubini.

6.2.4. Izračun Steinbrennerovog utjecajnog faktora *I* za zadanu koordinatu

6.2.4.1. Implementacija Steinbrennerove metode u C# programski jezik

Kako bi izračunali slijeganje tla ispod ili oko temelja, Steinbrennerovu metodu potrebno je implementirati u C# programski jezik.

Steinbrennerovom metodom dobivamo utjecajni koeficijent *I* za svaku dubinu mjerenja. Klasa za izračun Steinbrennerove formule zove se *SteinbrennerFormula.*

public class SteinbrennerFormula : MonoBehaviour

{

//Funkcija za izračun Steinbrennerove formule s širinom i dužinom temelja kao argumentima

```
   public (List<double> list1, List<double> list2) CalculateSteinbrenner(double widthB, double lengthL)
   {
```

//Referenca na klasu ProgramState

```
      GameObject programManager = GameObject.Find("ProgramManager");
      ProgramState programState = programManager.GetComponent<ProgramState>();
```

//Deklaracija varijabli u koje spremamo dubine mjerenja i pripadajuću vrijednost utjecajnog faktora I

```
      List<double> depthList = new List<double>();
      List<double> valueList = new List<double>();
```

//For petlja za izračun i spremanje vrijednosti utjecajnog faktura I za pripadajuće dubine

```
      for (double i = 0.0; i < programState.dubinaZ; i += programState.inkrementMjerenjaZ)
      {
         double trenutnaDubinaZ = i;
         double razlikaDubine = programState.dubinaZ - trenutnaDubinaZ;
```

//Steinbrennerova formula

```
         double firstFraction = (double)Math.Atan((lengthL * widthB) / (trenutnaDubinaZ * Math.Sqrt(Math.Pow(lengthL, 2) + Math.Pow(widthB, 2) + Math.Pow(trenutnaDubinaZ, 2))));
         double secondFraction = (double)(lengthL * widthB * trenutnaDubinaZ) / (Math.Sqrt(Math.Pow(lengthL, 2) + Math.Pow(widthB, 2) + Math.Pow(trenutnaDubinaZ, 2)));
         double thirdFraction = (double)(((1) / (Math.Pow(lengthL, 2) + Math.Pow(trenutnaDubinaZ, 2))) + ((1) / (Math.Pow(widthB, 2) + Math.Pow(trenutnaDubinaZ, 2))));
         double steinbrennerIzracun = (double)((1) / (2 * Math.PI)) * (firstFraction + secondFraction * thirdFraction);
```

//Dodavanje utjecajnog faktura za trenutnu dubinu u prethodno deklarirane varijable

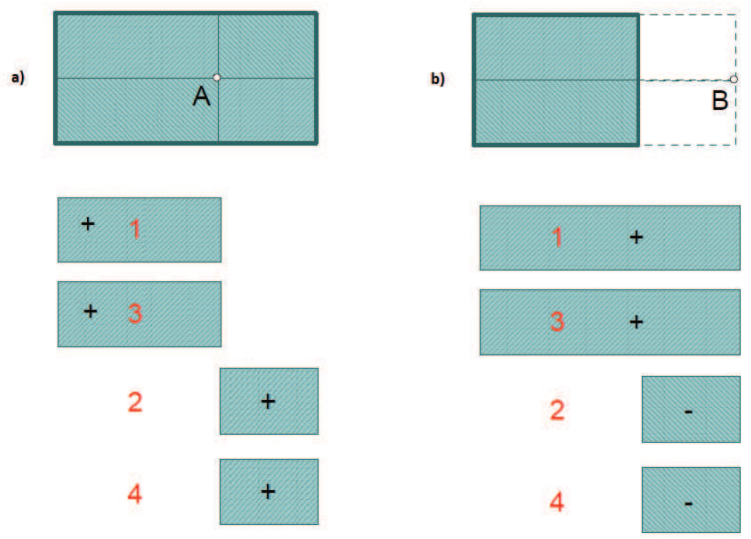
```
            depthList.Add(trenutnaDubinaZ);
            valueList.Add(steinbrennerIzracun);
            if (razlikaDubine <= programState.inkrementMjerenjaZ)
            {
                trenutnaDubinaZ = programState.dubinaZ;
                firstFraction = (double)Math.Atan((lengthL * widthB) / (trenutnaDubinaZ *
Math.Sqrt(Math.Pow(lengthL, 2) + Math.Pow(widthB, 2) + Math.Pow(trenutnaDubinaZ, 2))));
                secondFraction = (double)(lengthL * widthB * trenutnaDubinaZ) /
(Math.Sqrt(Math.Pow(lengthL, 2) + Math.Pow(widthB, 2) + Math.Pow(trenutnaDubinaZ, 2)));
                thirdFraction = (double)(((1) / (Math.Pow(lengthL, 2) + Math.Pow(trenutnaDubinaZ,
2))) + ((1) / (Math.Pow(widthB, 2) + Math.Pow(trenutnaDubinaZ, 2))));
                steinbrennerIzracun = (double)((1) / (2 * Math.PI)) * (firstFraction + secondFraction
* thirdFraction);
                depthList.Add(trenutnaDubinaZ);
                valueList.Add(steinbrennerIzracun);
            }
        }
        return (depthList, valueList);
    }
}
```

Pozivanjem funkcije *CalculateSteinbrenner(),* funkcija vraća dvije varijable tipa *„List of doubles“* koje sadrže Steinbrennerove utjecajne faktore i pripadajuće dubine mjerenja.

6.2.4.2. Izračun utjecajnog faktora *I* za točku odabranih koordinata principom *superpozicije*

Da bismo izračunali Steinbrennerov utjecajni faktor za definirani temelji i točku odabranih koordinata temelj odnosno pravokutnu površinu je potrebno podijeliti na više pravokutnika. Te pravokutnike zvat ćemo fiktivni temelji. Rubna točka svakog fiktivnog temelja mora biti točka odabranih koordinata. Zbrajanjem odnosno oduzimanjem vrijednosti utjecajnog faktora *I* za svaki fiktivni temelj dobivamo utjecajni faktor *I* za prvobitno definiran temelj u točki odabranih

koordinata. Način definiranja i zbrajanja odnosno oduzimanja fiktivnih temelja uvjetuje lokacija točke odabranih koordinata. Na slici 12 dan je shematski prikaz izračuna utjecajnog faktora za točku unutar i izvan temelja.



Slika 12. *Shematski prikaz izračuna utjecajnog faktora I za a) točku A unutar temelja i b) vanjsku točku B na površini, izvan temelja*

Algoritam odabire jedan od četiri načina izračuna utjecajnog faktora I na temelju lokacije točke odabranih koordinata u odnosu na definirani temelj:

- Točka je unutar temelja
- Točka je izvan temelja po x i y koordinati
- Točka je izvan temelja, samo po x koordinati
- Točka je izvan temelja, samo po y koordinati

...

//Referenca na klasu SteinbrennerFormula u kojoj se nalazi funkcija CalculateSteinbrenner()

SteinbrennerFormula initializeSteinbrennerCalculation =
programManager.GetComponent<SteinbrennerFormula>();

//Deklaracija i inicijalizacija varijable u koju spremamo konačan rezultat

```
(List<double> depthList, List<double> valueList) SteinBrennerRezultat;

    SteinBrennerRezultat.depthList = new List<double>();
    SteinBrennerRezultat.valueList = new List<double>();

//koordinate odabrane tocke

    double[] odabranaTocka = programState.koordinateIzracuna;
    odabranaTocka[0] = Math.Abs(odabranaTocka[0]);
    odabranaTocka[1] = Math.Abs(odabranaTocka[1]);
    programState.maxIValue = 0;

//Provjera da li je odabrana točka unutar temelja

    if (odabranaTocka[0] <= programState.sirinaB / 2
    && odabranaTocka[1] <= programState.duzinaL / 2)
    {
        double[] FiktivniTemeljPrvi = { ((programState.sirinaB / 2) + odabranaTocka[0]),
((programState.duzinaL / 2) - odabranaTocka[1]) };
        IndividualFictionalSquare TemeljPrviObjekt = new
IndividualFictionalSquare(FiktivniTemeljPrvi.Min(), FiktivniTemeljPrvi.Max(),
initializeSteinbrennerCalculation.CalculateSteinbrenner(FiktivniTemeljPrvi.Min(),
FiktivniTemeljPrvi.Max()));

        double[] FiktivniTemeljDrugi = { ((programState.sirinaB / 2) - odabranaTocka[0]),
((programState.duzinaL / 2) - odabranaTocka[1]) };
        IndividualFictionalSquare TemeljDrugiObjekt = new
IndividualFictionalSquare(FiktivniTemeljDrugi.Min(), FiktivniTemeljDrugi.Max(),
initializeSteinbrennerCalculation.CalculateSteinbrenner(FiktivniTemeljDrugi.Min(),
FiktivniTemeljDrugi.Max()));
        double[] FiktivniTemeljTreci = { ((programState.sirinaB / 2) + odabranaTocka[0]),
((programState.duzinaL / 2) + odabranaTocka[1]) };
```

```
        IndividualFictionalSquare TemeljTreciObjekt = new
IndividualFictionalSquare(FiktivniTemeljTreci.Min(), FiktivniTemeljTreci.Max(),
initializeSteinbrennerCalculation.CalculateSteinbrenner(FiktivniTemeljTreci.Min(),
FiktivniTemeljTreci.Max()));
        double[] FiktivniTemeljCetvrti = { ((programState.sirinaB / 2) - odabranaTocka[0]),
((programState.duzinaL / 2) + odabranaTocka[1]) };
        IndividualFictionalSquare TemeljCetvrtiObjekt = new
IndividualFictionalSquare(FiktivniTemeljCetvrti.Min(), FiktivniTemeljCetvrti.Max(),
initializeSteinbrennerCalculation.CalculateSteinbrenner(FiktivniTemeljCetvrti.Min(),
FiktivniTemeljCetvrti.Max()));
```

//for petlja za spremanje konačne vrijednosti utjecajnog faktora u varijablu SteinBrennerRezultat

```
        for (var i = 0; i < TemeljPrviObjekt.SteinbrennerResult.depthList.Count; i++)
        {

SteinBrennerRezultat.depthList.Add(TemeljPrviObjekt.SteinbrennerResult.depthList[i]);
            if (Double.IsNaN(TemeljPrviObjekt.SteinbrennerResult.valueList[i]))
            {
                TemeljPrviObjekt.SteinbrennerResult.valueList[i] = 0;
            }
            if (Double.IsNaN(TemeljDrugiObjekt.SteinbrennerResult.valueList[i]))
            {
                TemeljDrugiObjekt.SteinbrennerResult.valueList[i] = 0;
            }
            if (Double.IsNaN(TemeljTreciObjekt.SteinbrennerResult.valueList[i]))
            {
                TemeljTreciObjekt.SteinbrennerResult.valueList[i] = 0;
            }
            if (Double.IsNaN(TemeljCetvrtiObjekt.SteinbrennerResult.valueList[i]))
            {
```

```
            TemeljCetvrtiObjekt.SteinbrennerResult.valueList[i] = 0;
        }
```

//Zbrajanje pojedinih fiktivnih temelja

```
        double ValueSum = TemeljPrviObjekt.SteinbrennerResult.valueList[i] +
TemeljDrugiObjekt.SteinbrennerResult.valueList[i] +
TemeljTreciObjekt.SteinbrennerResult.valueList[i] +
TemeljCetvrtiObjekt.SteinbrennerResult.valueList[i];
        SteinBrennerRezultat.valueList.Add(ValueSum);
      }
    }
```

//Logika za slučaj kad je točka odabranih koordinata van temelja

```
    else
    {
```

//Tip točke 0 : točka je van temelja po x i y osi

```
        int dotTypeLocation = 0;
        double[] FiktivniTemeljPrvi = { ((programState.sirinaB / 2) + odabranaTocka[0]),
((programState.duzinaL / 2) + odabranaTocka[1]) };
        IndividualFictionalSquare TemeljPrviObjekt = new
IndividualFictionalSquare(FiktivniTemeljPrvi.Min(), FiktivniTemeljPrvi.Max(),
initializeSteinbrennerCalculation.CalculateSteinbrenner(FiktivniTemeljPrvi.Min(),
FiktivniTemeljPrvi.Max()));
        double[] FiktivniTemeljDrugi = { ((programState.sirinaB / 2) + odabranaTocka[0]),
((odabranaTocka[1] - (programState.duzinaL / 2))) };
        IndividualFictionalSquare TemeljDrugiObjekt = new
IndividualFictionalSquare(FiktivniTemeljDrugi.Min(), FiktivniTemeljDrugi.Max(),
initializeSteinbrennerCalculation.CalculateSteinbrenner(FiktivniTemeljDrugi.Min(),
FiktivniTemeljDrugi.Max()));
```

```
        double[] FiktivniTemeljTreci = { ((odabranaTocka[0] - (programState.sirinaB / 2))),
((programState.duzinaL / 2) + odabranaTocka[1]) };
        IndividualFictionalSquare TemeljTreciObjekt = new
IndividualFictionalSquare(FiktivniTemeljTreci.Min(), FiktivniTemeljTreci.Max(),
initializeSteinbrennerCalculation.CalculateSteinbrenner(FiktivniTemeljTreci.Min(),
FiktivniTemeljTreci.Max()));
        double[] FiktivniTemeljCetvrti = { ((odabranaTocka[0] - (programState.sirinaB / 2))),
(odabranaTocka[1] - (programState.duzinaL / 2)) };
        IndividualFictionalSquare TemeljCetvrtiObjekt = new
IndividualFictionalSquare(FiktivniTemeljCetvrti.Min(), FiktivniTemeljCetvrti.Max(),
initializeSteinbrennerCalculation.CalculateSteinbrenner(FiktivniTemeljCetvrti.Min(),
FiktivniTemeljCetvrti.Max()));

        if (odabranaTocka[0] > programState.sirinaB / 2 && odabranaTocka[1] <=
(programState.duzinaL / 2))
        {

//Tip točke 1: točka je van temelja samo po x osi

            dotTypeLocation = 1;
            FiktivniTemeljPrvi = new double[] { ((programState.sirinaB / 2) +
odabranaTocka[0]), ((programState.duzinaL / 2) - odabranaTocka[1]) };
            TemeljPrviObjekt = new IndividualFictionalSquare(FiktivniTemeljPrvi.Min(),
FiktivniTemeljPrvi.Max(),
initializeSteinbrennerCalculation.CalculateSteinbrenner(FiktivniTemeljPrvi.Min(),
FiktivniTemeljPrvi.Max()));
            FiktivniTemeljDrugi = new double[] { ((programState.sirinaB / 2) +
odabranaTocka[0]), (programState.duzinaL - ((programState.duzinaL / 2) - odabranaTocka[1]))
};
            TemeljDrugiObjekt = new IndividualFictionalSquare(FiktivniTemeljDrugi.Min(),
FiktivniTemeljDrugi.Max(),
```

```
initializeSteinbrennerCalculation.CalculateSteinbrenner(FiktivniTemeljDrugi.Min(),
FiktivniTemeljDrugi.Max()));
            FiktivniTemeljTreci = new double[] { (odabranaTocka[0] - (programState.sirinaB /
2)), ((programState.duzinaL / 2) - odabranaTocka[1]) };
            TemeljTreciObjekt = new IndividualFictionalSquare(FiktivniTemeljTreci.Min(),
FiktivniTemeljTreci.Max(),
initializeSteinbrennerCalculation.CalculateSteinbrenner(FiktivniTemeljTreci.Min(),
FiktivniTemeljTreci.Max()));
            FiktivniTemeljCetvrti = new double[] { (odabranaTocka[0] - (programState.sirinaB /
2)), (programState.duzinaL - ((programState.duzinaL / 2) - odabranaTocka[1])) };
            TemeljCetvrtiObjekt = new IndividualFictionalSquare(FiktivniTemeljCetvrti.Min(),
FiktivniTemeljCetvrti.Max(),
initializeSteinbrennerCalculation.CalculateSteinbrenner(FiktivniTemeljCetvrti.Min(),
FiktivniTemeljCetvrti.Max()));
        }
        else if (odabranaTocka[0] <= (programState.sirinaB / 2) && odabranaTocka[1] >
(programState.duzinaL / 2))
        {

//Tip točke 2: točka je van temelja samo po y osi

            dotTypeLocation = 2;
            FiktivniTemeljPrvi = new double[] { ((programState.sirinaB / 2) +
odabranaTocka[0]), ((programState.duzinaL) + (odabranaTocka[1] - (programState.duzinaL /
2))) };
            TemeljPrviObjekt = new IndividualFictionalSquare(FiktivniTemeljPrvi.Min(),
FiktivniTemeljPrvi.Max(),
initializeSteinbrennerCalculation.CalculateSteinbrenner(FiktivniTemeljPrvi.Min(),
FiktivniTemeljPrvi.Max()));
            FiktivniTemeljDrugi = new double[] { (programState.sirinaB - ((programState.sirinaB
/ 2) + odabranaTocka[0])), ((programState.duzinaL) + (odabranaTocka[1] -
(programState.duzinaL / 2))) };
```

```
        TemeljDrugiObjekt = new IndividualFictionalSquare(FiktivniTemeljDrugi.Min(),
FiktivniTemeljDrugi.Max(),
initializeSteinbrennerCalculation.CalculateSteinbrenner(FiktivniTemeljDrugi.Min(),
FiktivniTemeljDrugi.Max()));
        FiktivniTemeljTreci = new double[] { ((programState.sirinaB / 2) +
odabranaTocka[0]), (odabranaTocka[1] - (programState.duzinaL / 2)) };
        TemeljTreciObjekt = new IndividualFictionalSquare(FiktivniTemeljTreci.Min(),
FiktivniTemeljTreci.Max(),
initializeSteinbrennerCalculation.CalculateSteinbrenner(FiktivniTemeljTreci.Min(),
FiktivniTemeljTreci.Max()));
        FiktivniTemeljCetvrti = new double[] { (programState.sirinaB -
((programState.sirinaB / 2) + odabranaTocka[0])), ((odabranaTocka[1] -
(programState.duzinaL / 2))) };
        TemeljCetvrtiObjekt = new IndividualFictionalSquare(FiktivniTemeljCetvrti.Min(),
FiktivniTemeljCetvrti.Max(),
initializeSteinbrennerCalculation.CalculateSteinbrenner(FiktivniTemeljCetvrti.Min(),
FiktivniTemeljCetvrti.Max()));
    }

//Pomoćna funkcija AddValue(int type) za zbrajanje fiktivnih temelja u slučaju kad je točka
van temelja
//AddValue funkcija kao argument prima tip definirane točke

    void AddValue(int type)
    {

//for petlja za spremanje konačne vrijednosti utjecajnog faktora u varijablu
SteinBrennerRezultat

        for (var i = 0; i < TemeljPrviObjekt.SteinbrennerResult.depthList.Count; i++)
        {
            double ValueSum = new double();
```

```
SteinBrennerRezultat.depthList.Add(TemeljPrviObjekt.SteinbrennerResult.depthList[i]);
            if (Double.IsNaN(TemeljPrviObjekt.SteinbrennerResult.valueList[i]))
            {
               TemeljPrviObjekt.SteinbrennerResult.valueList[i] = 0;
            }
            if (Double.IsNaN(TemeljDrugiObjekt.SteinbrennerResult.valueList[i]))
            {
               TemeljDrugiObjekt.SteinbrennerResult.valueList[i] = 0;
            }
            if (Double.IsNaN(TemeljTreciObjekt.SteinbrennerResult.valueList[i]))
            {
               TemeljTreciObjekt.SteinbrennerResult.valueList[i] = 0;
            }
            if (Double.IsNaN(TemeljCetvrtiObjekt.SteinbrennerResult.valueList[i]))
            {
               TemeljCetvrtiObjekt.SteinbrennerResult.valueList[i] = 0;
            }
```

//Zbrajanje utjecajnih faktora I za tip točke 0

```
            if (type == 0)
            {
            ValueSum = TemeljPrviObjekt.SteinbrennerResult.valueList[i] -
TemeljDrugiObjekt.SteinbrennerResult.valueList[i] -
TemeljTreciObjekt.SteinbrennerResult.valueList[i] +
TemeljCetvrtiObjekt.SteinbrennerResult.valueList[i];
            }
```

//Zbrajanje utjecajnih faktora I za tip točke 1

```
            else if (type == 1)
```

```
            {

                ValueSum = TemeljPrviObjekt.SteinbrennerResult.valueList[i] +
TemeljDrugiObjekt.SteinbrennerResult.valueList[i] -
TemeljTreciObjekt.SteinbrennerResult.valueList[i] -
TemeljCetvrtiObjekt.SteinbrennerResult.valueList[i];
            }
//Zbrajanje utjecajnih faktora I za tip točke 2
            else if (type == 2)
            {
                ValueSum = TemeljPrviObjekt.SteinbrennerResult.valueList[i] +
TemeljDrugiObjekt.SteinbrennerResult.valueList[i] -
TemeljTreciObjekt.SteinbrennerResult.valueList[i] -
TemeljCetvrtiObjekt.SteinbrennerResult.valueList[i];
            }
            SteinBrennerRezultat.valueList.Add(ValueSum);
          }
        }
        AddValue(dotTypeLocation);
    } ...
```

6.2.5. Provedba i vizualizacija izračuna dodatnog naprezanja

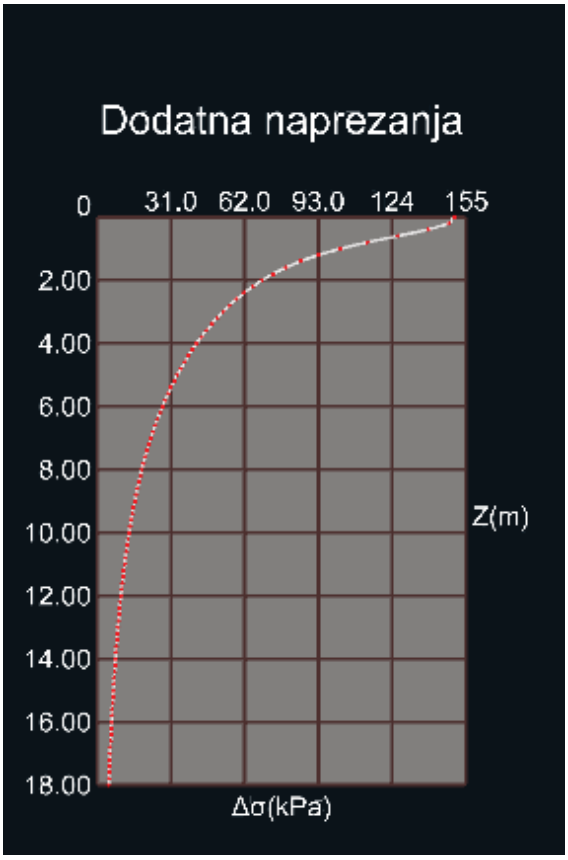
Množenjem utjecajnog faktora *I* za određenu dubinu s površinskim naprezanjem (kPa) dobivamo dodatno naprezanje za odabranu dubinu (slika 13).

```
...
//Inicijalizacija varijable dodatnaNaprezanja

  programState.dodatnaNaprezanja = new List<double>();
```

//for petlja za izračun dodatnog naprezanja za svaki inkrement mjerenja

```
  for (int i = 0; i < SteinBrennerRezultat.valueList.Count; i++)
      {
         programState.dodatnaNaprezanja.Add(SteinBrennerRezultat.valueList[i] *
programState.povrsinskoNaprezanje);
      } ...
```

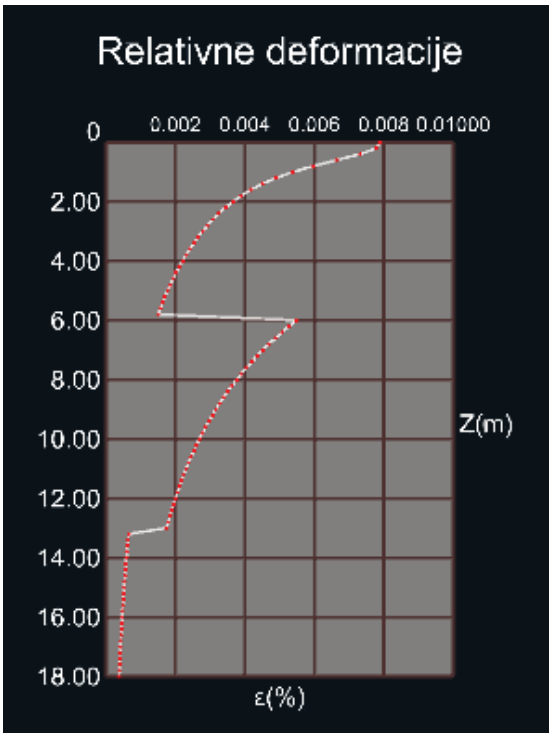


Slika 13. *Vizualizacija dodatnih naprezanja grafom*

6.2.6. Provedba i vizualizacija izračuna relativnih deformacija

Omjer dodatnog naprezanja Δσ za određenu dubinu s modulom stišljivosti u toj dubini tla daje relativnu deformaciju tla za zadanu dubinu (slika 14). Korištenjem modula stišljivosti zanemarujemo bočne deformacije.

```
...
//Inicijalizacija varijable relativneDeformacije
  programState. relativneDeformacije= new List<double>();
//for petlja za izračun relativne deformacije za svaki inkrement mjerenja
  for (int i = 0; i < programState.dodatnaNaprezanja.Count; i++)
      {
         programState.relativneDeformacije.Add(programState.dodatnaNaprezanja[i] /
programState.modulStisljivostiPoDubini[i]);
      }...
```



Slika 14. *Vizualizacija relativnih deformacija grafom*

6.2.7. Provedba i vizualizacija izračuna slijeganja tla

Proces izračuna vrijednosti slijeganja tla:

1. Definiranje vrijednosti početnog slijeganja tla

   double currentSoilSettlement = 0;

2. Iteracija kroz vrijednosti relativnih deformacija počevši od zadnje te za svaku vrijednost:
   - Izračun razlike slijeganja
   - Kumulativno zbrajanje trenutne razlike slijeganja s prošlom
   - Dodavanje vrijednosti trenutnog slijeganja u varijablu *ukupnoSlijeganje*

…

```
//Inicijalizacija varijable ukupnoSlijeganje

programState.ukupnoSlijeganje = new List<double>();

//for petlja za iteraciju kroz vrijednosti relativnih deformacija počevši od zadnje

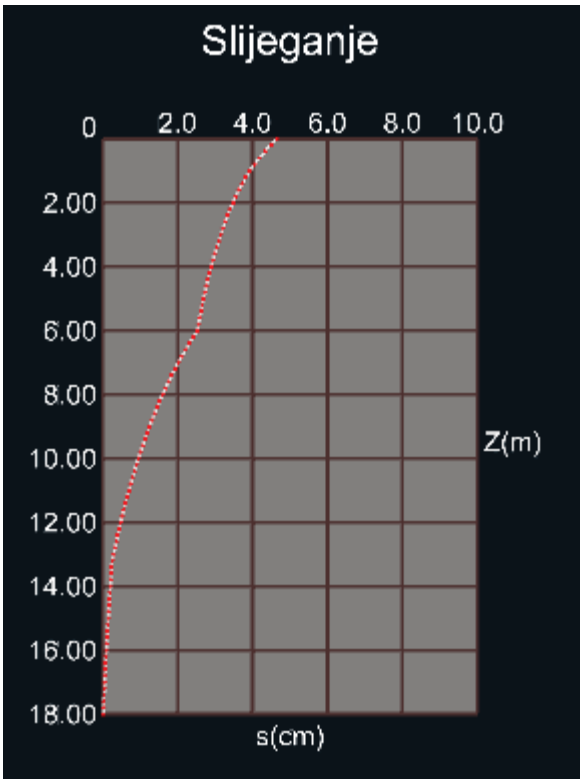
for (int i = programState. relativneDeformacije.Count - 1; i >= 0; i--)
    {
        double deltaSettlement = programState.relativneDeformacije[i] *
        programState.inkrementMjerenjaZ;
        currentSoilSettlement += deltaSettlement;

//množenje s 100 radi dobivanja vrijednosti u centimetrima

        programState.ukupnoSlijeganje.Add(currentSoilSettlement * 100);
    }...
```

3. Korištenje *for* petlje kako bi preokrenuli redoslijed vrijednosti varijable *ukupnoSlijeganje* tipa *„List of doubles“*

```
for (int i = 0; i < programState.ukupnoSlijeganje.Count; i++)
    {
        double slijeganje =
programState.ukupnoSlijeganje[(programState.ukupnoSlijeganje.Count - 1) - i];
        programState.ukupnoSlijeganje.RemoveAt((programState.ukupnoSlijeganje.Count -
1) - i);
        programState.ukupnoSlijeganje.Add(slijeganje);
    }
```



Slika 15. *Vizualizacija slijeganja tla grafom*

7. KORIŠTENJE PROGRAMA

Program je dizajniran da bude jednostavan za korištenje i dostupan što većem broju platforma sposobnih pokretanja Unity projekata. Njegova jednostavna izvedba i kompatibilnost s Android i iOS mobilnim uređajima omogućuje korisniku brz uvid u slijeganje odabranog temelja čak i na lokaciji istražnih radova.

7.1. Upute za korištenje programa

Parametri korišteni u svrhu demonstracije korištenja programa su:

- Dužina temelja: 6m
- Širina temelja: 4m
- Dubina mjerenja: 18m
- Inkrement mjerenja: 0,2m
- Površinsko naprezanje: 150kPa
- Broj slojeva: 3

Tablica 3. *Odabrani parametri slojeva*

Raspon sloja (m)	Modul stišljivosti *Ms* (MPa)
0-6	19
6-13	5
13-18	13

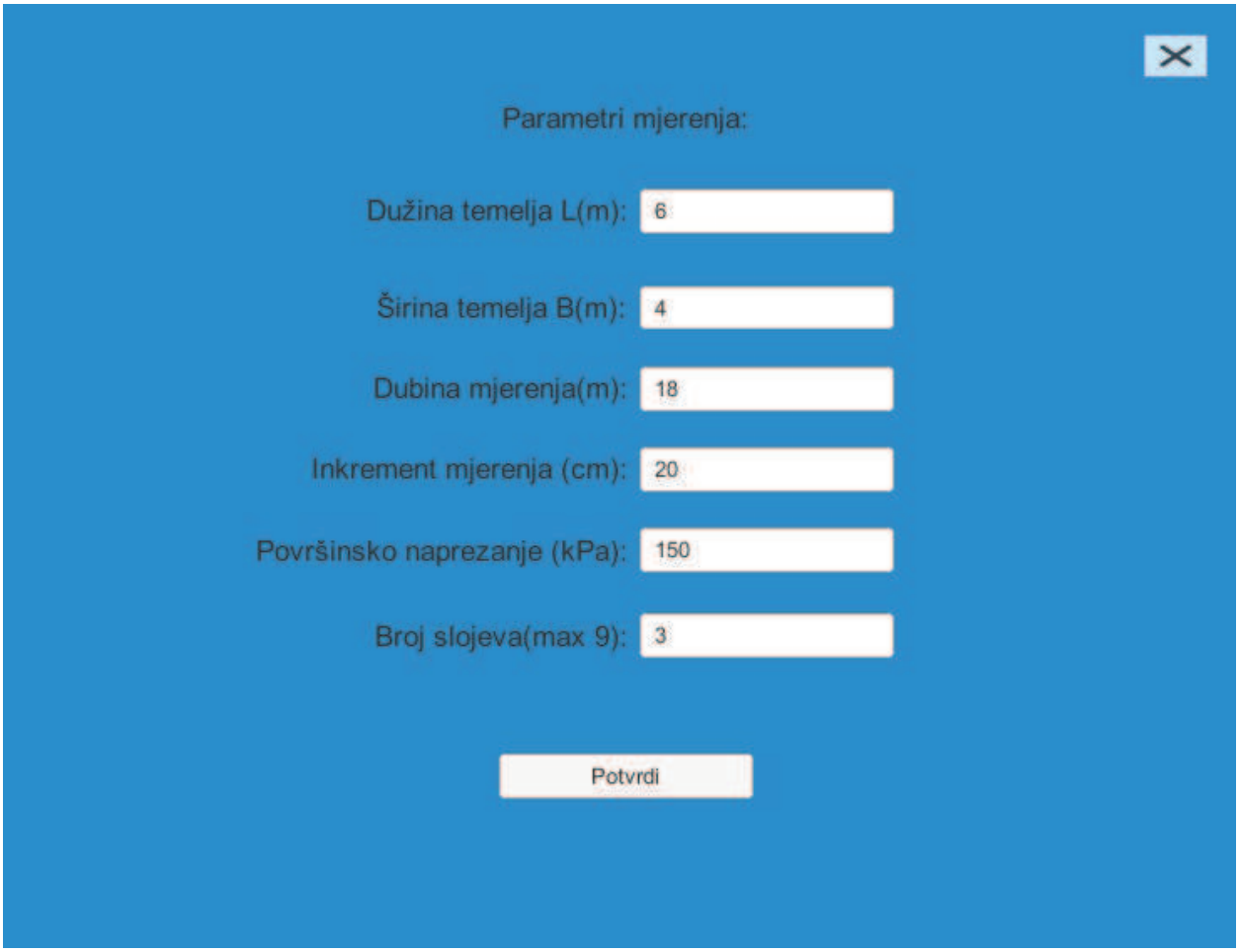
Za sve izračune odabrana je karakteristična točka temelja.

7.1.1. Unos parametara mjerenja

Kako bi program funkcionirao na predviđen način, korisnik mora definirati parametre mjerenja (slika 16):

1. Unos dužine temelja L (m)
2. Unos širine temelja B(m)
3. Unos dubine mjerenja(m)
4. Definiranje inkrementa mjerenja(cm)
5. Unos površinskog naprezanja (kPa)
6. Unos broja slojeva

Parametri mjerenja ne moraju nužno biti cijeli brojevi. Ukoliko dužina ili širina temelja nije definirana, program neće dopustiti korisniku potvrdu podataka.



Slika 16. *Unos parametara mjerenja*

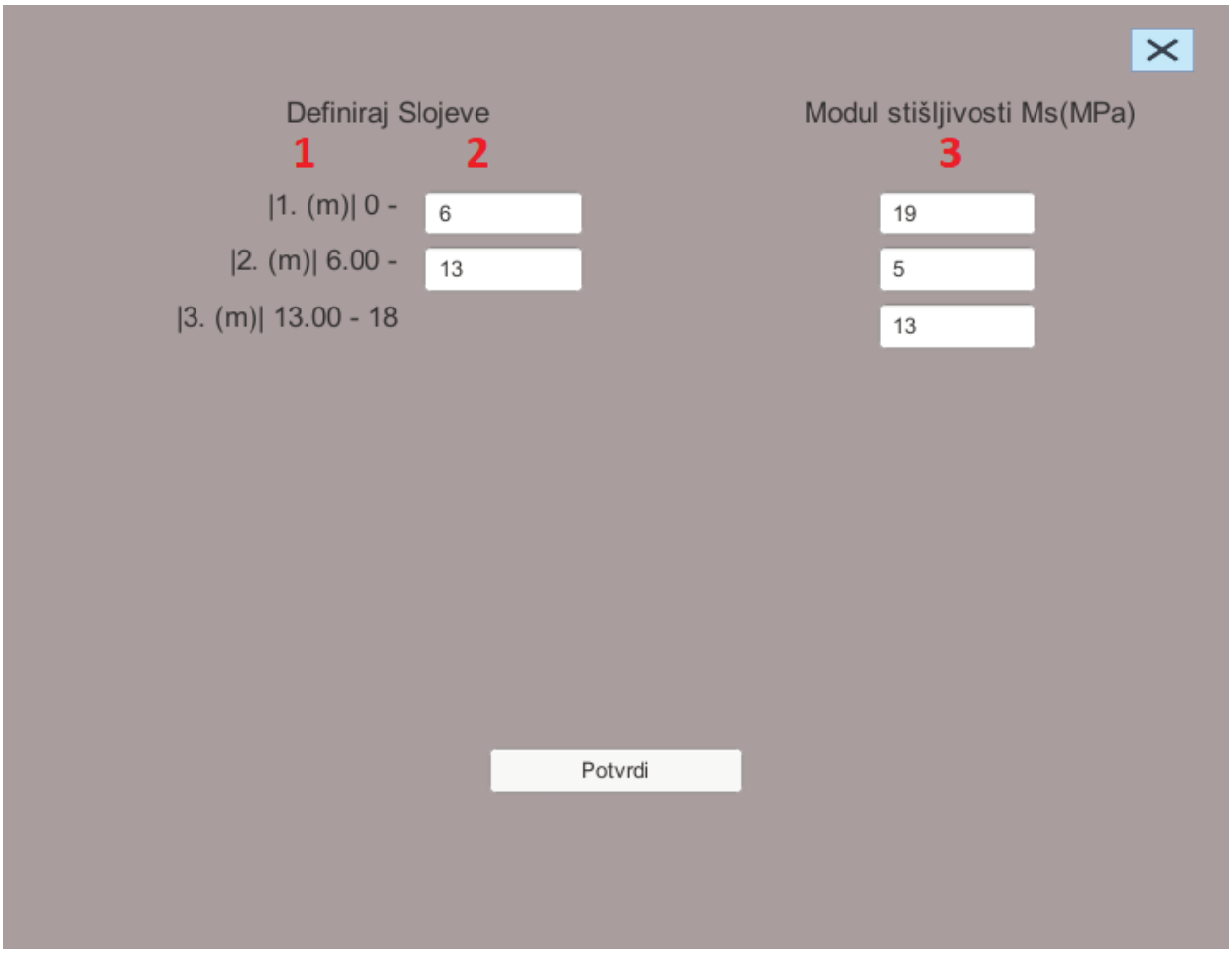
7.1.2. Unos raspona dubine i modula stišljivosti slojeva

Ekran za unos raspona dubine i modula stišljivosti slojeva sadrži 3 stupca (slika 17):

1. Gornja granica sloja
2. Donja granica sloja
3. Modul stišljivosti sloja

Gornja granica sloja automatski se popunjava ovisno o donjoj granici prethodnog sloja.

Napomena: Program će dopustiti definiranje istog raspona sloja različitim modulom stišljivosti te će uzeti u obzir modul stišljivosti sloja s manjim rednim brojem.



Slika 17. *Unos parametara slojeva*

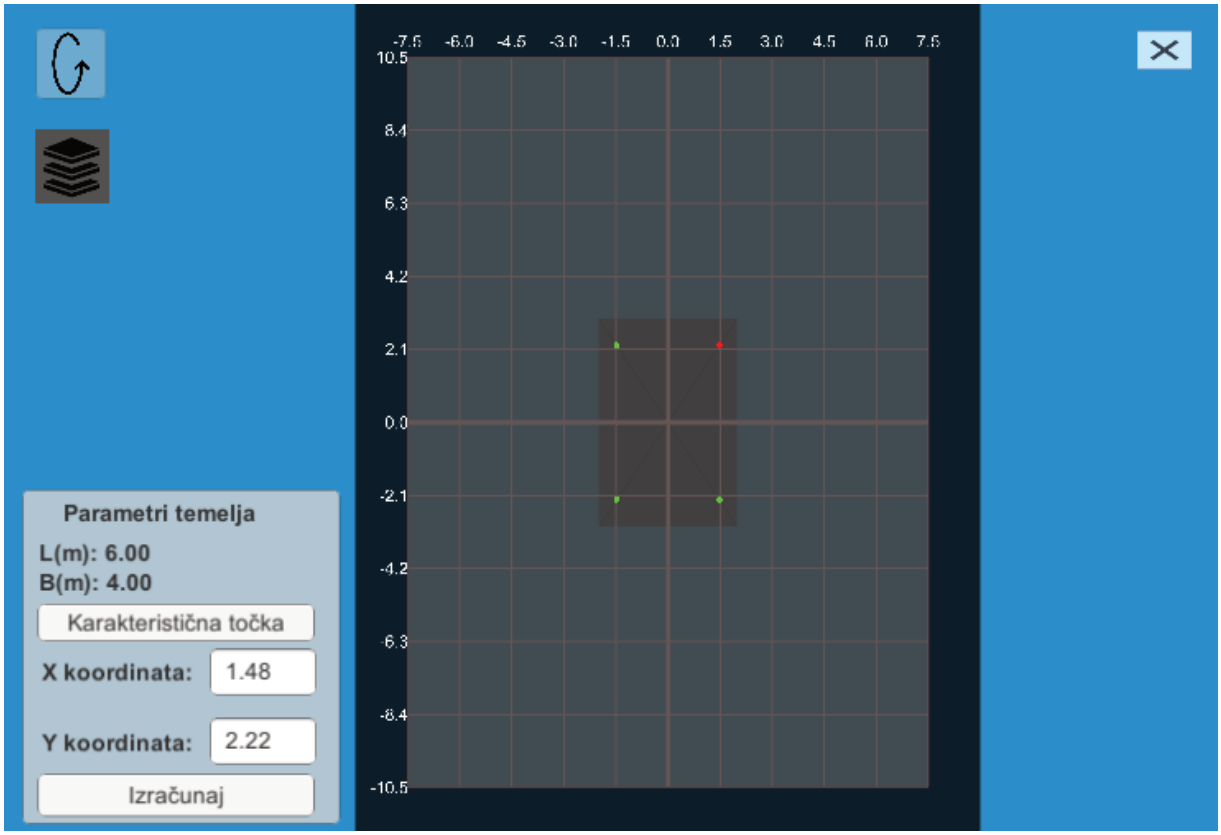
7.1.3. Unos koordinate izračuna

Nakon uspješno definiranih parametara mjerenja i slojeva korisnik je u mogućnosti unijeti koordinatu izračuna slijeganja.

Korisnik ima tri mogućnosti definiranja koordinate (slika 18):

1. Brojčani unos koordinate X i Y

2. Lijevim pritiskom na željenu koordinatu u koordinatnom sustavu
3. Odabir koordinate karakteristične točke temelja



Slika 18. *Odabir koordinate izračuna*

7.2. Korisničko sučelje i dodatne mogućnosti programa

7.2.1. Tipkovničke kratice i navigacija kamere

Korisnik je u mogućnosti pritiskom na odabrana slova na tipkovnici vršiti sljedeće naredbe

- Slovo w: translatiraj kameru prema gore
- Slovo s: translatiraj kameru prema dolje
- Slovo a: translatiraj kameru u lijevo

- Slovo d: translatiraj kameru u desno
- Slovo q: smanji prikaz
- Slovo e: povećaj prikaz
- Slovo r: ponovno pokreni program

7.2.2. Korisničko sučelje

Korisničko sučelje sastoji se od sljedećih elemenata (slika 19):

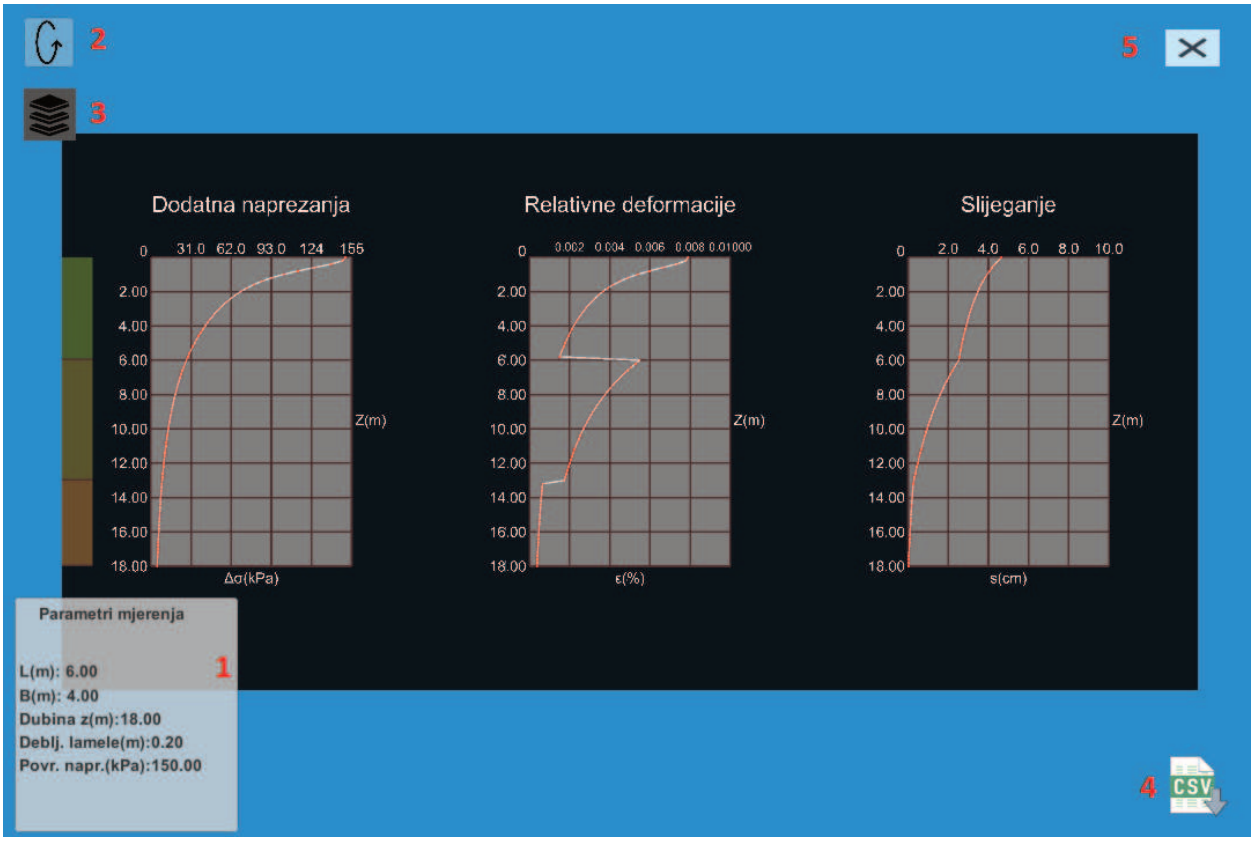
(1) Prozorčić s definiranim parametrima mjerenja

(2) Gumb za prebacivanje pogleda ekran s koordinatnim sustavom za unos koordinate izračuna

(3) Gumb za prebacivanje pogleda na ekran za definiranje slojeva

(4) Gumb za izvoz podataka u CSV datoteku

(5) Gumb za izlaz iz aplikacije



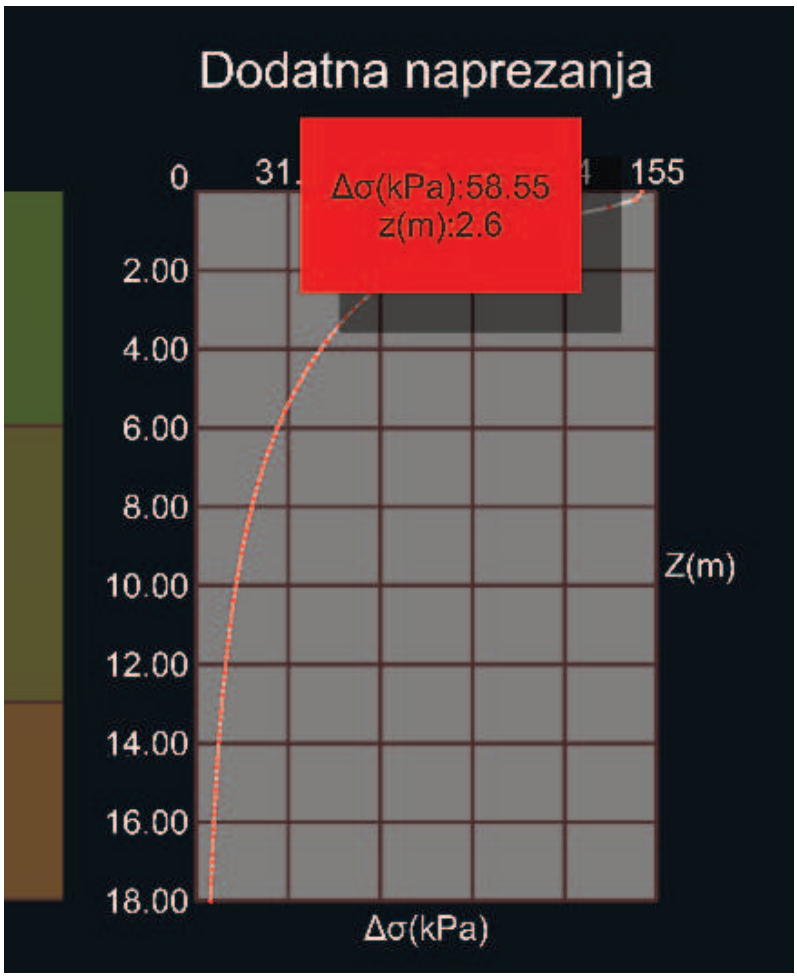
Slika 19. *Korisničko sučelje*

Napomena: Kod ekrana za unos koordinate izračuna prozorčić s definiranim parametrima mjerenja prikazuje informacije o parametrima temelja i odabiru koordinate izračuna.

7.2.3. Dodatne mogućnosti programa

Korisniku je omogućen pritisak lijevom tipkom miša na bilo koju točku grafa. Pritiskom na točku prikazuje se prozorčić s vrijednostima izračuna u toj točki kao što je prikazano za točku dubine 2.6 m na slici 20 (a). S lijeve strane grafa dodatnih naprezanja nalazi se vizualni prikaz raspona prethodno definiranih slojeva.

Vizualni prikaz raspona slojeva prikazuje slika 20 (b).



Slika 20. *Prikaz raspona dubine slojeva (a), prikaz dodatnog prozorčića s vrijednostima izračuna (b)*

Korisniku je dana mogućnost izvoza podataka mjerenja u CSV datoteku. Pritiskom na gumb za izvoz podataka u CSV datoteku (4) poziva se funkcija *AppendToReport()* unutar klase *ExportToCsv*. Informacije koje se spremaju u CSV datoteku za sve dubine mjerenja:

- Trenutna dubina
- Steinbrennerov utjecajni faktor
- Modul stišljivosti
- Dodatno naprezanje
- Relativna deformacija
- Slijeganje
- Dužina temelja
- Širina temelja
- Koordinata točke izračuna
- Debljina lamele (inkrement mjerenja)
- Vrijeme mjerenja

Program podržava spremanje više od jednog mjerenja u istu CSV datoteku.

8. PROSTOR ZA UNAPRIJEĐENJE RADA

Mogućnosti unaprjeđenja programa možemo razdvojiti u dvije kategorije.

- Unaprjeđenje iskoristivosti programa
- Unaprjeđenje mogućnosti izračuna programa

8.1. Unaprjeđenje iskoristivosti programa

Programski kod u svom trenutnom stanju može se pokrenuti na platformama:

- Windows OS
- Mac OS
- Linux
- Android
- iOS
- WebGL

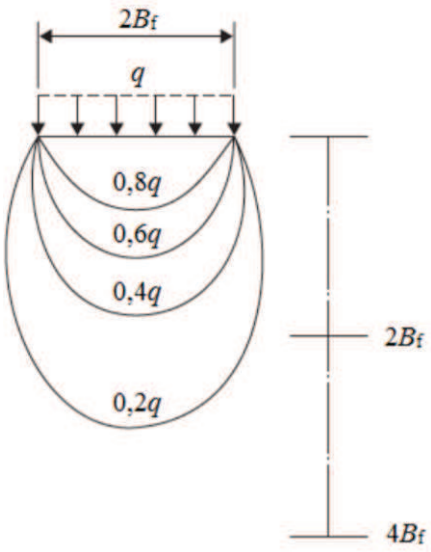
Na platformama Android, iOS i WebGl izvoz u CSV datoteku nije podržan, dodatno, na mobilnim operativnim sustavima nisu implementirane zamjene za tipkovničke kratice. Drugim riječima korisnik na mobitelu nije u mogućnosti mijenjati poziciju kamere i s lakoćom ponovno pokrenuti program kako bi unio nove parametre mjerenja.

Navedene funkcionalnosti je moguće dodati u programski kod za gore navedene platforme. Dostupnost CSV datoteke bi se mogla povećati na način da program sprema CSV datoteku na Google Drive. Na taj bi način, osoba na lokaciji istražnih radova mogla podijeliti svoja mjerenja kolegama u uredu.

8.2. Unaprjeđenje mogućnosti izračuna programa

Program nema mogućnost istovremenog izračuna slijeganja više točaka. Za izračun slijeganja nove točke, potrebno je definirati koordinate nove točke na ekranu za unos koordinate izračuna. Ukoliko korisnik nije spremio svoj prijašnji izračun u CSV datoteku, pritiskom na gumb Izračunaj gube se svi izračuni za prijašnju točku.

Implementacija mogućnosti izračuna više točaka istovremeno, ubrzala bi proces mjerenja te bi istovremeno omogućila veću mogućnost grafičke prezentacije podataka.
Izračun slijeganja više točaka pruža nam mogućnost izračuna dubine jednakih dodatnih naprezanja i slijeganja i prezentacije dobivenog izračuna krivuljom jednakih slijeganja.



Slika 21. *Prikaz krivulja jednakih dodatnih naprezanja ispod temelja (Knappett, Craig, 2012)*

Principom superpozicije moguće je izračunati utjecaj više pravokutnih temelja na odabranu točku. Zbrajanjem Steinbrennervog utjecajnog faktora *I* svakog pojedinog temelja dobivenog principom superpozicije za odabranu točku dobiva se utjecajni faktor I svih temelja u odabranoj točki. Temelju čiji se oblik može aproksimirati s više pravokutnika također je moguće izračunati principom superpozicije i Steinbrennerovom formulom.
Za sve druge oblike temelja potrebno je iskoristiti formule prikladne za zadani oblik temelja, poput Newmarkove formule za slijeganje kružno opterećene površine ili Newmarkove mreže.

Trenutna formula daje izračun slijeganja linearno elastičnog edometarskog modela tla. Implementacija proračuna drugih modela tla, poput nelinearnog modela, ili logaritamsko edometarskog modela tla uvelike bi povećala funkcionalnost programa.

Prvim opterećenjem uzorka tla u edometru dobivamo modul stišljivosti kao odnos deformacija i sile opterećenja. Rasterećenjem uzorka i ponovnim opterećenjem odnos deformacija i sile opterećenja nije ista. Parametar koji proizlazi uslijed drugog opterećenja uzorka zove se rasteretni modul.
Implementacijom rasteretnog modula u program dobili bi uvid o utjecaju rasterećenja uslijed iskopa ispod temeljne plohe.

9. DOPRINOS PRI IZRADI RADA

Razvijen je efikasan i robustan algoritam za proračune slijeganja u jednoliko uslojenom tlu. Program ponajviše može biti koristan pri početnim određivanjima vrijednosti odgovarajućih slijeganja. Istovremeno je dosegnuta visoka razina grafičke interpretacije ulaznih vrijednosti i dobivenih rezultata. Ona omogućava snažnu kontrolu procesa određivanja traženih i dobivenih veličina. Zbog svoje lake dostupnosti putem web preglednika i drugih platformi, i jednostavnosti korištenja program je prikladan za korištenje i u edukativne svrhe.

10. ZAKLJUČAK

U radu je općenito prikazana zadaća određivanja slijeganja tla, odnosno temeljnih konstrukcija. Utvrđivanje vrijednosti slijeganja pojedine točke iziskuje sveobuhvatni napor u pogledu provedbe odgovarajućih istražnih radova, te niz radnji u odabiru modela tla i konačnih proračunskih postupaka.

Ovdje je prikazana analiza za linearno elastičan edometarski model tla. Izrađen je algoritam za proračun slijeganja proizvoljne točke uslijed jednoliko opterećene pravokutne površine, za proizvoljan broj deformabilnih slojeva tla.

Izrađenim rješenjem moguće je efikasno odrediti vrijednosti slijeganja. Odgovarajućim grafičkim programskim rješenjima omogućena je kvalitetna vizualna kontrola i interpretacija dobivenih rezultata. Usporedbom rezultata s drugim rješenjima utvrđeno je da je program ispravno implementiran te da je postavljeni cilj ostvaren, no to ne znači da je ovo konačna verzija programa. Poglavlje 8. pruža uvid u potencijalna poboljšanja programa.

10. LITERATURA

1. Gotić, R., Gotić, I. (2000). Temeljenje građevina na poboljšanom tlu. Stručni rad. Građevinar 53 (2001) pp. 75-81
2. Kvasnička, P., Domitrović, D. (2007) Mehanika tla. Interna skripta. Sveučilište u Zagrebu Rudarsko geološko naftni fakultet
3. Roje-Bonacci, T. (2007) Mehanika tla. Sveučilište u Splitu Građevinsko-arhitektonski fakultet
4. Lukić Kristić, I., Szavits-Nossan, V., Mišćević, P. (2017) Direktan postupak za određivanje slijeganja plitkih temelja. Građevinar 69 pp. 467-477
5. Davis, E.H., Poulos, H.G.: The use of elastic theory for settlement prediction under three-dimensional conditions, Géotechnique, 18 (1968) 1, pp. 67–91
6. Mayne, P.W., Poulos, H.G.: Approximate displacement influence factors for elastic shallow foundations, Journal of Geotechnical and Geoenvironmental Engineering, 125 (1999) 6, pp. 453-460
7. Szavits-Nossan, V.(2013) Slijeganje plitkih temelja. Mehanika tla i stijena. 11 predavanje. Sveučilište u Zagrebu Građevinski fakultet
8. Gang , S., Ryu, D., Choi, Y., Choung, Y. (2017). Unity3D-Based Flood Simulation Visualization Web System for Efficient Disaster Management. Journal of the Korean Association of Geographic Information Studies, 20 (1), pp. 98-112
9. Khalifa, H.K., Nguyen, Q.V., Simoff, S., Catchpoole, D. (2015) AVisualization System for Analyzing Biomedical and Genomic Data Sets Using Unity3D Platform, HIKM 2015. Australia
10. Buyuksalih, G., Bayburta, S., Baskaraca, A.P., Karim, H., Rahman, A.A. (2017). Calculating solar energy potential of buildings and visualization within unity 3D game engine. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume XLII-4/W5, Malaysia
11. Ecma standard (2017) C# Language Specification; https://www.ecma-international.org/publications/files/ECMA-ST/ECMA-334.pdf Posljednji puta pristupljeno: 02.09.2020.
12. Knappett, J.A., Craig, R.F. (2012) Craig's Soil Mechanics. 8th ed. Spon Press, Oxon, UK

POPIS SLIKA

Slika 1. *Širenje dodatmih naprezanja u tlu* *2*
Slika 2. *Raspodjela dodatnih naprezanja tijekom (a) linijskog opterećenja, (b) kvadratne plohe* *3*
Slika 3*. Aksonometrijski prikaz slijeganja ispod koncentrirane sile na* *7*
Slika 4. *Aksonometrijski prikaz slijeganja ispod kružno opterećene površine* *8*
Slika 5*. Aksonometrijski prikaz slijeganja ispod ugla pravokutno opterećene površine* *9*
Slika 6. *Slijeganje kružnog temelja ukopanog u sloj tla Gibsonove vrste* *13*
Slika 7. *Utjecajni koeficijenti Isy za vertikalnih deformacija rubne točke pravokutnog temelja prema Newmark-u i Steinbrenner-u* *15*
Slika 8. *Utjecani koeficijent Iσ za dodatno vertikalno naprezanje ispod vrha pravokutne plohe na površini linearno elastičnog, izotropnog i homogenog poluprostora opterećenog jednolikim vertikalnim opterećenjem* *18*
Slika 9*. Proračun dodatnih naprezanja u točki „ispod“ temelja* *20*
Slika 10. *Proračun dodatnih naprezanja u točki „izvan“ temelja* *21*
Slika 11. *Unity – okruženje za razvoj i vizualizaciju programa* *22*
Slika 12. *Shematski prikaz izračuna utjecajnog faktora I za a) točku A unutar temelja i b) vanjsku točku B na površini, izvan temelja* *30*
Slika 13. *Vizualizacija dodatnih naprezanja grafom* *39*
Slika 14. *Vizualizacija relativnih deformacija grafom* *40*
Slika 15. *Vizualizacija slijeganja tla grafom* *42*
Slika 16. *Unos parametara mjerenja* *44*
Slika 17. *Unos parametara slojeva* *45*
Slika 18. *Odabir koordinate izračuna* *46*
Slika 19. *Korisničko sučelje* *47*
Slika 20. *Prikaz raspona dubine slojeva (a), prikaz dodatnog prozorčića s vrijednostima izračuna (b)* 48
Slika 21. *Prikaz krivulja jednakih dodatnih naprezanja ispod temelja* 51

POPIS TABLICA

Tablica 1. *Okvirne vrijednosti modula stišljivosti (Ms) za nevezana tla..... 5*
Tablica 2. *Okvirne vrijednosti modula stišljivosti (Ms) za vezana tla 5*
Tablica 3. *Odabrani parametri slojeva 43*