Numeričko rješavanje i vizualizacija rješenja Laplaceove jednadžbe

Strelec, Filip

Undergraduate thesis / Završni rad

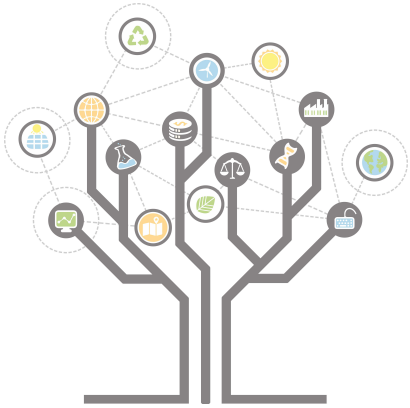
2018

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:130:701281

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-05-20**



Repository / Repozitorij:

Repository of Faculty of Geotechnical Engineering - Theses and Dissertations

zir.nsk.hr

dabar
DIGITALNI AKADEMSKI ARHIVI I REPOZITORIJI

Numeričko rješavanje i vizualizacija rješenja Laplaceove jednadžbe

Strelec, Filip

Undergraduate thesis / Završni rad

2018

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:130:701281

Rights / Prava: In copyright

Download date / Datum preuzimanja: **2020-10-27**

Repository / Repozitorij:

Repository of Faculty of Geotechnical Engineering

SVEUČILIŠTE U ZAGREBU

GEOTEHNIČKI FAKULTET

FILIP STRELEC

NUMERIČKO RJEŠAVANJE I VIZUALIZACIJA RJEŠENJA LAPLACEOVE JEDNADŽBE

ZAVRŠNI RAD

VARAŽDIN, 2018.

SVEUČILIŠTE U ZAGREBU

GEOTEHNIČKI FAKULTET

ZAVRŠNI RAD

NUMERIČKO RJEŠAVANJE I VIZUALIZACIJA RJEŠENJA LAPLACEOVE JEDNADŽBE

KANDIDAT:

FILIP STRELEC

MENTOR:

doc.dr.sc. IVAN HIP

MENTOR/KOMENTOR:

izv.prof.dr.sc. IGOR PETROVIĆ

VARAŽDIN, 2018.

Sveučilište u Zagrebu
Geotehnički fakultet

ZADATAK ZA ZAVRŠNI RAD

Pristupnik: FILIP STRELEC

Matični broj: 2640 - 2015./2016.

NASLOV ZAVRŠNOG RADA:

NUMERIČKO RJEŠAVANJE I VIZUALIZACIJA RJEŠENJA LAPLACEOVE JEDNADŽBE

Rad treba sadržati:
1. Uvod
2. Laplaceova jednadžba
3. Rješavanje Laplaceove jednadžbe metodom konačnih razlika
4. Implementacija u JavaScript-u
5. Upute za korištenje programa
6. Testiranje-usporedba s analitičkim rješenjem
7. Usporedba s drugim programskim rješenjima
8. Zaključak
9. Popis literature
10. Popis slika

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

Zadatak zadan: 16.04.2018. Rok predaje: 06.09.2018.

Mentor:	Drugi mentor/komentor:	Predsjednik Odbora za nastavu:
Doc.dr.sc. Ivan Hip	Izv.prof.dr.sc. Igor Petrović	Izv.prof.dr.sc. Igor Petrović

REPUBLIKA HRVATSKA - GEOTEHNIČKI FAKULTET - SVEUČILIŠTE U ZAGREBU - VARAŽDIN

ZADATAK ZAVRŠNOG RADA

Jednadžba kontinuiteta za nestlačivi fluid kombinirana s Darcyjevim zakonom daje Laplaceovu jednadžbu koja opisuje stacionarno tečenje. Problem stacionarnog tečenja vode u tlu svojstven je mnogim geotehničkim zahvatima. Na primjer, tečenje vode kroz temeljno tlo brane, tečenje vode kroz nasutu branu, ili pak problem tečenja podzemne vode u građevnu jamu. U svim tim slučajevima, uslijed značajnih hidrauličkih gradijenata, može doći do pojave erozije tla ili čak i do hidrauličkog sloma kada tlo u potpunosti gubi svoju nosivost. Nadalje, rješenje Laplaceove jednadžbe osnovna je polazišna osnova za modeliranje pronosa zagađivala kroz tlo (advekcija, molekularna difuzija, hidrodinamička disperzija, adsorpcija). Kada se definiraju odgovarajući rubni uvjeti Laplaceovu jednadžbu može se numerički riješiti metodom relaksacije. Kôd za numeričko rješavanje treba implementirati u programskom jeziku JavaScript, a programsko sučelje i vizualizaciju rješenja u HTML5 tehnologijama, tako da se programsko rješenje može koristiti iz standardnog web preglednika.

IZJAVA O AKADEMSKOJ ČESTITOSTI

Izjavljujem i svojim potpisom potvrđujem da je završni rad pod naslovom

NUMERIČKO RJEŠAVANJE I VIZUALIZACIJA LAPLACEOVE JEDNADŽBE

rezultat mog vlastitog rada koji se temelji na istraživanjima te objavljenoj i citiranoj literaturi te je izrađen pod mentorstvom doc.dr.sc. Ivana Hipa i izv.prof.dr.sc. Igora Petrovića.

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 visokoškolskoj, znanstvenoj ili obrazovnoj ustanovi.

U Varaždinu, 06.09.2018.

Filip Strelec

(Ime i prezime)

(Vlastoručni potpis)

SAŽETAK RADA

Tema završnog rada je razvoj računalnog programa za numeričko rješavanje i vizualizaciju Laplaceove jednadžbe. Program je ciljano razvijen u JavaScript programskom jeziku, a vizualizacija rješenja implementirana je u HTML5 element *canvas*, kako bi program bio lako dostupan putem standardnog web preglednika. U završnom radu objašnjena je Laplaceova jednadžba i metoda konačnih razlika koja je korištena za rješavanje Laplaceove jednadžbe te implementacija te metode u JavaScript u svrhu izračuna i vizualizacije Laplaceove jednadžbe. Rezultati programa uspoređeni su s analitičkim rješenjem i drugim programskim rješenjima kako bi provjerili njegovu točnost.

KLJUČNE RIJEČI

Laplaceova jednadžba, metoda konačnih razlika, JavaScript, HTML5

Sadržaj

1. Uvod 1
2. Laplaceova jednadžba 2
   2.1 Tečenje vode u tlu 2
      2.1.1. Jednadžba kontinuiteta 2
      2.1.2. Laplaceova jednadžba 3
   2.2. Elektrostatika 4
   2.3. Stacionarno vođenje topline 4
   2.4. Rubni uvjeti 5
3. Rješavanje Laplaceove jednadžbe metodom konačnih razlika 6
   3.1. Aproksimacija derivacija konačnim razlikama 6
   3.2. Primjena metode konačnih razlika za rješavanje Laplaceove jednadžbe 8
4. Implementacija u JavaScript-u 9
   4.1. Implementacija metode konačnih razlika 9
   4.2. Grafičko korisničko sučelje 12
      4.2.1. Unos parametara 12
      4.2.2. Zadavanje rubnih uvjeta 15
   4.3. Vizualizacija rješenja 17
      4.3.1. HTML5 *canvas* – grafičko programsko sučelje 19
      4.3.2. Reprezentacija rješenja bojom 19
      4.3.3. Prikaz vektorskog polja brzina 22
      4.3.4. Ekvipotencijale 24
5. Upute za korištenje programa 28
6. Testiranje – usporedba s analitičkim rješenjem 30
   6.1. Analitičko rješenje 30
      6.1.1. Opis problema 31
      6.1.2. Razdvajanje (separacija) varijabli 31
      6.1.3. Usklađivanje s rubnim uvjetima 33
      6.1.4. Superpozicija rješenja 34
      6.1.5. Rubni uvjet (IV) 35
      6.1.6. Slučaj kad je *f(x)* konstanta 36
   6.2. Konvergencija metode konačnih razlika 36
7. Usporedba s drugim programskim rješenjima 39

8. Zaključak 43
9. Popis literature 44
10. Popis slika 45

1. Uvod

Laplaceova jednadžba pojavljuje se u mnogim znanstvenim područjima, a do njezinog rješenja moguće je doći na više načina. Današnja računala imaju veliku procesorsku snagu, što ih čini idealnim kandidatima za rješavanje Laplaceove jednadžbe. Cilj rada bio je razviti kvalitetan program koji točno i brzo rješava probleme iz raznih područja znanosti i tehnike koji se mogu riješiti Laplaceovom jednadžbom, a da pritom bude lako razumljiv za korisnika. Program učinkovito koristi metodu konačnih razlika da bi riješio Laplaceovu jednadžbu.

Prvi dio rada opisuje Laplaceovu jednadžbu (poglavlje 2.) i rješavanje Laplaceove jednadžbe metodom konačnih razlika (poglavlje 3.). Drugi dio rada posvećen je opisu implementacije metode konačnih razlika u JavaScript i vizualizacije rješenja jednadžbe (poglavlje 4.). Zadnja poglavlja rada objašnjavaju problematiku analitičkog rješavanja Laplaceove jednadžbe te uspoređuju programsko rješenje s analitičkim i s rješenjima drugih programskih paketa (poglavlja 6. i 7.).

Programski kod je dostupan na GitHub-u na adresi:

https://github.com/ihip/fstrelec-Laplace

Program se može isprobati na adresi:

https://ihip.github.io/fstrelec-Laplace/

2. Laplaceova jednadžba

Laplaceova jednadžba je linearna homogena parcijalna diferencijalna jednadžba drugog reda koja se pojavljuje u mnogim znanstvenim područjima, najznačajnije u poljima elektromagnetizma, mehanici fluida i nauci o toplini. Pomoću Laplaceove jednadžbe precizno možemo opisati ponašanje elektrostatskih i hidrauličkih potencijala.

Nama je najzanimljivija primjena za opis tečenja vode u homogenom tlu, ali razvijena aplikacije može se koristiti i za izračunavanje elektrostatskog potencijala ili raspodjele temperature kod stacionarnog vođenja topline.

2.1 Tečenje vode u tlu

Tečenje vode u homogenom i izotropnom tlu opisuje Darcyjev zakon. Zakon je formulirao Henry Darcy. Zapisano u suvremenoj matematičkoj notaciji

$$\vec{v}_D = -k\,\vec{\nabla}h(x,y,z) \qquad (1)$$

pri čemu je $\vec{v}_D$ Darcyjeva odnosno prividna brzina, k koeficijent propusnosti materijala dobiven laboratorijskim ili terenskim mjerenjima. Koeficijent propusnosti se također može posredno odrediti empirijskim formulama korelacijom s podacima iz granulometrijske krivulje. Slovo h označava hidraulički potencijal te je zapravo potencijalna energija po jedinici težine tekućine

$$h = \frac{p}{\rho g} + z \qquad (2)$$

2.1.1. Jednadžba kontinuiteta

Jednadžba kontinuiteta zapravo je matematički zapis zakona očuvanja mase. Međutim, naivna formulacija koja bi samo istaknula da je suma svih masa jednaka konstanti nije dovoljna. Na primjer, kad bi dva kilograma materije nestalo na Zemlji, a istovremeno se pojavilo na Marsu, zakon očuvanja mase bio bi formalno, u svojoj najjednostavnijoj

formi, zadovoljen. Jednadžba kontinuiteta nadopunjuje zakon o očuvanju mase i nalaže da transport mase mora biti kontinuiran, stvaranje mase mora biti kompenzirano nestajanjem na nekom drugom mjestu, ali samo ako se radi o točkama u neposrednom susjedstvu između kojih postoji kontinuirani protok ili prijelaz mase. Jednadžba kontinuiteta predstavlja matematički egzaktan opis zakona očuvanja mase koji uvažava tu činjenicu. (Hip, 2014)

Diferencijalni oblik jednadžbe kontinuiteta glasi:

$$\frac{\partial \rho}{\partial t} + \vec{\nabla} \cdot (\rho \vec{v}) = 0 \qquad (3)$$

Pri čemu je ρ gustoća fluida, t vrijeme, a $\vec{v}$ vektorsko polje brzine toka. Budući da je voda praktično nestlačiva vrijedi :

$$\vec{\nabla} \cdot \vec{v} = 0 \qquad (4)$$

2.1.2. Laplaceova jednadžba

Uvrštavanjem Darcyjevog zakona (1) u jednadžbu kontinuiteta za nestlačivi fluid (4) dobiva se da hidraulički potencijal mora zadovoljavati sljedeću jednadžbu:

$$\vec{\nabla} \cdot \vec{\nabla} h(x, y, z) = \vec{\nabla}^2 \, h(x, y, z) = \Delta h(x, y, z) = 0 \qquad (5)$$

Pri čemu je

$$\Delta \equiv \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2} \qquad (6)$$

Laplaceov operator (engl. *Laplacian*), a homogena parcijalna diferencijalna jednadžba

$$\Delta h(x, y, z) = 0 \qquad (7)$$

naziva se Laplaceova jednadžba.

Rješavanjem Laplaceove jednadžbe (7) za odgovarajuće rubne uvjete može se dobiti hidraulički potencijal $h(x, y, z)$, a iz njega pomoću Darcyjevog zakona (1) možemo dobiti polje brzina $\vec{v}(x, y, z)$ te odgovarajuće protoke koji nas najčešće zanimaju u inženjerskoj praksi.

Za detaljniji opis problematike strujanja podzemnih voda vidjeti (Jović, 2006).

2.2. Elektrostatika

Električno polje $\vec{E}$ je negativni gradijent električnog potencijala U

$$\vec{E}(x,y,z) = -\vec{\nabla}U(x,y,z)$$

a za polje također mora vrijediti Gaussov zakon

$$\vec{\nabla} \cdot \vec{E}\,(x,y,z) = \frac{\rho}{\varepsilon_0}\,(x,y,z)$$

gdje ρ označava gustoću električnog naboja, a ε_0 označava dielektričnu permitivnost vakuuma.

U prostoru bez naboja vrijedi:

$$\vec{\nabla} \cdot \vec{E}(x,y,z) = 0$$

Što zajedno s formulom (1) daje

$$\vec{\nabla} * \vec{\nabla}U(x,y,z) = \vec{\nabla}^2 U(x,y,z) = \Delta U(x,y,z) = 0$$

iz čega zaključujemo da električni potencijal mora zadovoljavati Laplaceovu jednadžbu

$$\frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial y^2} + \frac{\partial^2 U}{\partial y^2} = 0.$$

Za više detalja vidjeti (Purcell, 1988).

2.3. Stacionarno vođenje topline

Iz Fourierovog zakona vođenja topline

$$\vec{q} = -\lambda\vec{\nabla}(x,y,z,t)$$

i prvog zakona termodinamike (zakon o očuvanju energije) slijedi Fourierova jednadžba:

$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\partial^2 T}{\partial y^2} = \frac{\rho c}{\lambda}\frac{\partial T}{\partial t}$$

Kada se uspostavi ravnotežno stanje, odnosno nema više promjene temperature, $\frac{\partial T}{\partial t}$ je jednak 0, pa je raspodjela temperature rješenje Laplaceove jednadžbe

$$\Delta T(x, y, z) = 0$$

koja u ovom slučaju opisuje stacionarno vođenje topline (engl. *steady-state conduction*).

Za više detalja vidjeti (Pitts i Sissom, 1998).

2.4. Rubni uvjeti

Da bi dobili jedinstveno rješenje konkretnog fizikalnog problema treba postaviti takozvane rubne uvjete.

Zadane vrijednosti potencijala ili temperature na rubovima domene zovu se Dirichletovi rubni uvjeti. Mogu se zadati i vrijednosti prve derivacije, dakle brzine ili gustoće toplinskog toka. U tom slučaju, takve rubne uvjete zovemo Neumannovim rubnim uvjetima. Također se može zadati i linearna kombinacija vrijednosti funkcije i prve derivacije. Ti rubni uvjeti zovu se Robinovi rubni uvjeti.

U mnogim fizikalnim problemima pojavljuju se miješani rubni uvjeti pri čemu je jedan dio rubova opisan Dirichletovim, a drugi Neumannovim rubnim uvjetima.

3. Rješavanje Laplaceove jednadžbe metodom konačnih razlika

Metoda konačnih razlika (diferencija) (Bronštejn i sur., 2004, Press et al., 1992) je jedna od metoda za rješavanje rubnih problema kod običnih i parcijalnih diferencijalnih jednadžbi uz pomoć računala. Derivacije se aproksimiraju konačnim razlikama, kako je to opisano u sljedećem potpoglavlju, a vrijednosti funkcije, tj. rješenja koje se traži, definirane su samo u ekvidistantnim točkama koje se nazivaju *čvorovi* i koji u 2D slučaju čine mrežu ili rešetku te se lako pohrane u memoriju računala.

3.1. Aproksimacija derivacija konačnim razlikama

Derivacija je limes kad Δx teži k nuli:

$$f'(x) = \lim_{\Delta x \to 0} \frac{f(x+\Delta x)-f(x)}{\Delta x} \qquad (1)$$

Ali što ako su nam poznate vrijednosti funkcije samo na točkama (čvorovima) s nekim konačnim razmakom Δx?

Koliko je izraz (1) bez provedenog limesa dobra aproksimacija derivacije

$$f'(x) \approx \frac{f(x+\Delta x)-f(x)}{\Delta x} \; ? \qquad (2)$$

Da bi odgovorili na ovo pitanje poslužit ćemo se razvojem u Taylorov red.

$$f(x+\Delta x) = f(x) + \Delta x\, f'(x) + \frac{1}{2}\Delta x^2\, f''(x) + \frac{1}{6}\Delta x^3\, f'''(x) + 0(\Delta x^4) \qquad (3)$$

Preslagivanjem članova prvu derivaciji možemo izraziti kao:

$$f'(x) = \frac{f(x+\Delta x)-f(x)}{\Delta x} - \frac{1}{2}\Delta x\, f''(x) - \frac{1}{6}\Delta x^2\, f'''(x) + 0(\Delta x^3) \qquad (4)$$

Usporedbom s (1) vidimo da je pogreška proporcionalna s Δx.

Međutim kombiniranjem Taylorovog reda (3) i

$$f(x-\Delta x) = f(x) - \Delta x\, f'(x) + \frac{1}{2}\Delta x^2\, f''(x) - \frac{1}{6}\Delta x^3\, f'''(x) + 0(\Delta x^4) \qquad (5)$$

može se dobiti bolja aproksimacija prve derivacije. Kad se od (5) oduzme (3) slijedi:

$$f(x+\Delta x) - f(x-\Delta x) = 2\Delta x\, f'(x) + \frac{1}{3}\Delta x^3\, f'''(x) + 0(\Delta x^5) \quad (6)$$

Preslagivanjem članova dobivamo da je prva derivacija:

$$f'(x) = \frac{f(x+\Delta x)-f(x-\Delta x)}{2\Delta x} - \frac{1}{6}\Delta x^2\, f'''(x) - 0(\Delta x^4) \qquad (7)$$

Dakle kod aproksimacije

$$f'(x) \approx \frac{f(x+\Delta x)-f(x-\Delta x)}{2\Delta} \qquad (8)$$

Pogreška je proporcionalna s Δx^2 što je povoljnije nego kod „naivne“ aproksimacije (2).

Na sličan način možemo dobiti i aproksimaciju za drugu derivaciju $f''(x)$.

Zbrajanjem (3) i (5) dobije se:

$$f(x+\Delta x) + f(x-\Delta x) = 2\, f(x) + \Delta x^2\, f''(x) + \frac{1}{12}\Delta x^4\, f^{(4)}(x) + 0(\Delta x^6) \qquad (9)$$

Iz čega slijedi:

$$f''(x) = \frac{f(x+\Delta x)-2f(x)+f(x-\Delta x)}{\Delta x^2} - \frac{1}{12}\Delta x^2 f^{(4)}(x) - 0(\Delta x^4) \qquad (10)$$

Dakle, drugu derivaciju možemo aproksimirati s:

$$f''(x) \approx \frac{f(x+\Delta x)-2f(x)+f(x-\Delta)}{\Delta x^2} \qquad (11)$$

pri čemu je pogreška proporcionalna s Δx^2.

Isti izraz dobili bi i dvostrukom primjenom aproksimacije prve derivacije (8).

Za opširniju i rigorozniju diskusiju pogrešaka vidjeti (Press i sur., 1992) i (Burden i Faires, 1989).

3.2. Primjena metode konačnih razlika za rješavanje Laplaceove jednadžbe

Naš program omogućit će rješavanje problema koji je definiran u dvije dimenzije. Potrebna nam je Laplaceova jednadžba za potencijal koji je funkcija dvije koordinate:

$$\frac{\partial^2 h}{\partial x^2} + \frac{\partial^2 h}{\partial y^2} = 0 \qquad (1)$$

U poglavlju 3.1. izveli smo formulu (11) za aproksimaciju druge derivacije:

$$\frac{\partial^2 f(x)}{\partial x^2} \approx \frac{f(x+\Delta x)+f(x-\Delta x)-2f(x)}{\Delta x^2} \qquad (2)$$

Uvrštavanjem (2) u (1) uz jednake razmake između čvorova u x i y smjeru

$$a = \Delta x = \Delta y$$

dolazimo do jednadžbe za izračun potencijala pojedinog čvora (Lang i Pucker, 2005):

$$h(x,y) = \frac{1}{4}(h(x+a,y) + h(x-a,y) + h(x,y+a) + h(x,y-a)) \qquad (3)$$

Potencijal svakog čvora u iteraciji $n+1$ izračunava se na temelju vrijednosti susjednih čvorova u n-toj iteraciji na sljedeći način:

$$h^{(n+1)}(x,y) = \frac{1}{4}(h^{(n)}(x+a,y) + h^{(n)}(x-a,y) + h^{(n)}(x,y+a) + h^{(n)}(x,y-a)) \quad (4)$$

Da bi iteriranje bilo smisleno i uspješno potrebno je definirati rubne uvjete koji se tokom iteracije ne mijenjanju. Iteriranje traje proizvoljno, odnosno iteriramo sve dok iznos potencijala pojedinog čvora ne konvergira do željene preciznosti.

Radi jednostavnosti implementacije Neumannove rubne uvjete ograničit ćemo na taj način da Neumannovim rubnim uvjetima možemo definirati vrijednost derivacije samo s vrijednosti 0. To u stvarnosti predstavlja nepropusnu barijeru jer je u tom smjeru i protok jednak nuli.

4. Implementacija u JavaScript-u

JavaScript, skraćeno JS, je programski jezik koji se izvršava u web pregledniku. Najčešće se koristi za animaciju i dinamičko pozicioniranje objekata u pregledniku. JS je vrlo moćan, ali i relativno jednostavan programski jezik što nam omogućava efikasnu implementaciju našeg koda u web preglednik te time program postaje lako dostupan na svim platformama. Sam JavaScript nema mogućnost vizualizacije programske logike, no HTML5 tehnologija omogućuje vizualizaciju našeg koda putem *canvas* elementa. HTML5 *canvas* je jednostavno ali moćno programsko sučelje za rad s grafikom pomoću kojeg možemo vizualizirati rješenja.

4.1. Implementacija metode konačnih razlika

Za implementaciju metode konačnih razlika u programski jezik JavaScript bilo je potrebno definirati dva 2D polja, jedno 2D polje koje će u sebi sadržavati vrijednosti potencijala u čvorovima koje sam nazvao *potencijal*, te drugo 2D polje kojim definiramo da li je pojedini čvor sadrži Dirichletov (*"d"*) ili Neumannov (*"n"*) rubni uvjet, ili je pak običan čvor koji sadrži vrijednost potencijala (*"h"*). To polje sam nazvao *rubniUvjet*. JavaScript sam po sebi nema mogućnost definiranja 2D polja te je bilo potrebno svako 1D polje staviti u *for* petlju, koja se izvršava onoliko puta koliko naša rešetka ima redova, te se svakim izvršenjem stvara novo 1D polje u oba 2D polja *potencijal* i *rubniUvjet*. Funkcija koja inicijalizira ta dva 2D polja zove se *inicijalizacija():*

```
function inicijalizacija () {
    for (var i = 0; i < brojRedova; i++) {
        potencijal[i] = new Array();
        rubniUvjet[i]= new Array();
        for (var j = 0; j < brojStupaca; j++) {
                rubniUvjet[i][j]="h"
               potencijal[i][j] = 0.0;
                rubniUvjet[0][j] = "n";
                potencijal[0][j] = 0.0;
```

```
                    rubniUvjet[i][0] = "n";
                    potencijal[i][0] = 0.0;
                    rubniUvjet[i][parseInt(heightG-1)] = "n";
                    potencijal[i][parseInt(heightG-1)] = 0.0;
            }
        }
        for (var j = 0; j < brojStupaca; j++) {
                rubniUvjet[parseInt(widthG-1)][j] = "n";
                potencijal[parseInt(widthG-1)][j] = 0.0;
        }

}
```

Kao što je vidljivo, uloga funkcije *inicijalizacija()* je također i definirati početne vrijednosti svakog elementa u polju. U početku svi elementi polja *potencijal* su predefinirani na vrijednost 0, a elementi polja *rubniUvjet* su predefinirani slovom *"h"*, koje označava potencijal. Svaki član koji se nalazi na rubu predefiniran je slovom *"n"* koje označava Neumannov rubni uvjet. Nakon funkcije *inicijalizacija()* potrebno je implementirati funkciju koja bi rješavala naš problem metodom konačnih razlika. Vrijednost potencijala u pojedinom čvoru dobivamo na sljedeći način:

1. Definiranje okolnih čvorova oko trenutnog čvora za koji računamo potencijal:

   ```
   var desno = potencijal[i+1][j];
   var dolje = potencijal[i][j+1];
   var lijevo = potencijal[i-1][j];
   var gore = potencijal[i][j-1];
   ```

2. Provjera da li naš čvor ne sadrži Dirichletov ili Neumannov uvjet:
   `if (rubniUvjet[i][j]==="d" || rubniUvjet[i][j]==="n" ) { ... }`

3. Provjera da li je Neumannov rubni uvjet u neposrednoj blizini trenutnog čvora za koji računamo potencijal, te ako je, ne uzimamo vrijednost tog čvora jer to polje je Neumannov rubni uvjet, već uzimamo vrijednost trenutnog čvora (već je u

potpoglavlju 3.2. spomenuto da ćemo se u implementaciji Neumannovih rubnih uvjeta ograničiti na vrijednost 0 koja predstavlja nepropusnu barijeru, a to ujedno znači da je prva derivacija jednaka nuli, tj. u tom smjeru nema promjene potencijala pa se čvor u kojem je definiran Neumannov rubni uvjet privremeno postavlja na istu vrijednost potencijala s trenutnim čvorom):

```
if (rubniUvjet[i+1][j]===“n“ )
                                {
                                desno = potencijal[i][j];
                                }
if (rubniUvjet[i-1][j]===“n“ )
                                {
                                lijevo = potencijal[i][j];
                                }
if (rubniUvjet[i][j+1]===“n“ )
                                {
                                dolje = potencijal[i][j];
                                }
if (rubniUvjet[i][j-1]===“n“ )
                                {
                                gore = potencijal[i][j];
                                }
```

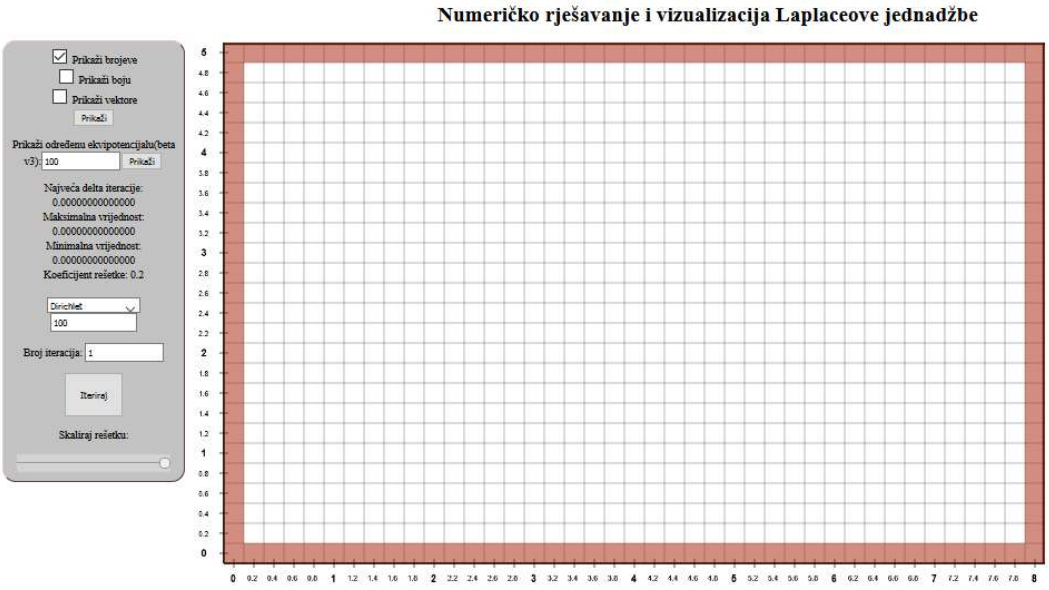
4. Implementacija formule (4) iz potpoglavlja 3.2. za izračun potencijala za trenutno polje:

   ```
   potencijal[i][j]=0.25*(gore+dolje+lijevo+desno);
   ```

Potrebno je, dakako, sav kod napisati u *for* petlji ugniježđenoj unutar još jedne *for* petlje kako bi prošli kroz sve čvorove.

4.2. Grafičko korisničko sučelje

Grafičko korisničko sučelje (engl. *graphical user interface*) je tip korisničkog sučelja koji omogućuje korisnicima interakciju s elektronskim uređajima putem ikona, prozora, gumbova i ostalih vizualnih elemenata koji omogućuju jednostavan način interakcije s programom. U našem slučaju od velike je važnosti da grafičko sučelje bude jednostavno i intuitivno kako bi korisniku pružili efikasan način unosa i čitanja podataka.



Slika 1. Grafičko korisničko sučelje našeg programa

4.2.1. Unos parametara

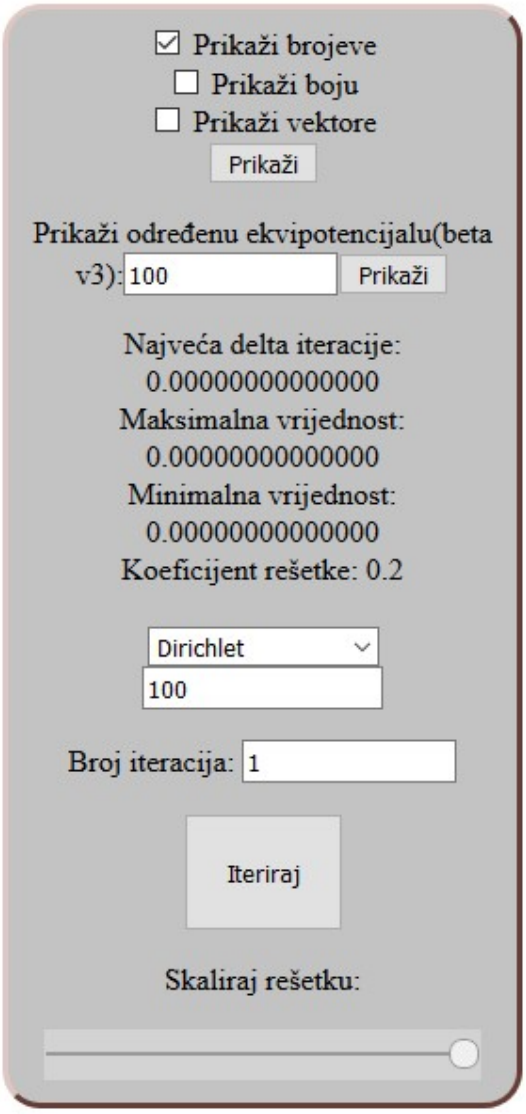
Kod problema numeričkog rješavanja i vizualizacije Laplaceove jednadžbe dolazimo do potrebe definiranja niza parametara koji su potrebni za rješavanje jednadžbe. Trebao sam naći jednostavan i brz način za unos parametara poput visine i širine presjeka, razmaka između čvorova, iznosa Dirichletovih rubnih uvjeta i broja iteracija. Odabir visine i širine presjeka, kao i broj čvorova po jedinici duljine iz kojeg se izračunava razmak između čvorova unosimo na samom početku učitavanja programa. Iskoristio sam JS *prompt()* metodu kako bi omogućio korisniku da unese dimenzije presjeka.

Funkcija za definiranje rešetke zove se *velicinaResetke()*:

```
function velicinaResetke() {
                visinaP = prompt("Visina presjeka", "5");
                sirinaP = prompt("Širina presjeka", "8");
                brojCvorova = prompt("Broj čvorova po jedinici visine/širine", "5");
                function daLiJeBroj(n) { return !isNaN(parseFloat(n)) && !isNaN(n - 0) }
                 visinaR = visinaP*brojCvorova+1;
                sirinaR = sirinaP*brojCvorova+1;
                razmakCvorova=1/brojBoxova;

                document.getElementById("razmakC").innerHTML = "Razmak čvorova: "+razmakCvorova;
        if(dubina % 1 === 0 && duzina % 1 === 0 && brojBoxova % 1 === 0  ) {
                }
                else {
                        if(daLiJeBroj(visinaP)==false
                        ||daLiJeBroj(sirinaP)==false
                        ||daLiJeBroj(brojCvorova)==false ) {
                                window.alert("Upisite broj, a ne slova!");
                                location.reload();

                        }
                                window.alert("Upisite cijeli broj!");
                                location.reload();

        }
  }
```

Sve ostale parametre korisnik može odrediti putem HTML elemenata za unos teksta (oznaka <input type="text">) i odabir (oznake *<input type="checkbox"*> i *<input type="button"* >) te sam ih stavio u jedan HTML *div* element koji uz unos parametara sadrži i niz drugih HTML elemenata potrebnih za interakciju s korisnikom i ispis važnih informacija poput minimalne i maksimalne vrijednost potencijala. *Div* element sam

pomoću CSS-a fiksirao u gornji lijevi kut ekrana te sam ga uredio radi lakše preglednosti informacija potrebnih za rad s programom.



Slika 2. Prikaz elemenata za interakciju s korisnikom

4.2.2. Zadavanje rubnih uvjeta

Numerička analiza Laplaceove jednadžbe ne bi bila moguća bez postavljanja uvjeta na granicama problema. Rješenje Laplaceove jednadžbe je jedinstveno, što pak ne bi bilo moguće da jednadžba nema postavljene rubne uvjete. Bez rubnih uvjeta nije moguće dobiti rješenje.

Da bi omogućio korisniku unos rubnih uvjeta u HTML5 *canvas* element bilo je potrebno implementirati slušatelje događaja (engl. *event listener*) (Powers, 2010) kako bi omogućili korisniku interakciju s programom pomoću miša te napisati funkciju koja će pozvati drugu funkciju nazvanu *lokacijaClicka()* kada korisnik pomakne miš, što možemo i vidjeti u JS kodu:

```
canvas.addEventListener('mousedown', clickDolje);
canvas.addEventListener('mouseup', clickGore);


function clickDolje(){
        canvas.addEventListener('mousemove', lokacijaClicka);
}


function clickGore(){
        canvas.removeEventListener("mousemove", lokacijaClicka);
}
```

Funkcija *lokacijaClicka()* služi kako bi od korisnika dobili koordinate njegovog miša kada drži lijevi klik i pomiče miša. X i Y koordinatu miša podijelimo s varijablom *velicinaKvadrata* kako bi znali u kojem čvoru se miš nalazi jer svakom čvoru odgovara predefinirani kvadrat veličine 30x30 piksela.

Funkcija lokacijaClicka():

```
function lokacijaClicka(e) {
        var i = parseInt(Math.floor(e.offsetX / velicinaKvadrata));
        var j = parseInt(Math.floor(e.offsetY / velicinaKvadrata));
        promjena(i,j);
}
```

Kao što je vidljivo, kad dobijemo i preradimo informaciju lokacije korisnikovog miša koji pokušava definirati rubni uvjet poziva se funkcija nazvana *promjena()* s parametrima koje smo dobili od korisnika. Funkcija *promjena()* služi kako bi provjerili parametre koje je korisnik odlučio definirati te da bi te parametre unijeli u 2D polja *rubniUvjet* i *potencijal*, te unesene parametre korisniku grafičkom interpretacijom prikazali na HTML5 *canvas* elementu.

```
function promjena(){
        var odaberi = document.getElementById("odaberi");
        var unos = odaberi.options[odaberi.selectedIndex].value;
    if (unos == "1") {
        broj = document.getElementById("broj").value;
        var BrojUGriduPomaki=velicinaKvadrata*i+1;
        var
        BrojUGriduPomakj=velicinaKvadrata*j+(0.9*velicinaKvadrata)
        ;
        rubniUvjet[i][j] = "d";
        potencijal[i][j] = parseFloat(broj);
        ctx.fillStyle = "rgba(104, 180, 247, 1)";
        ctx.fillRect((i * velicinaKvadrata)+1.5, (j *
        velicinaKvadrata)+1.5, velicinaKvadrata-3, velicinaKvadrata-3);
        ctx.fillStyle ="black";
        ctx.font = "15px Arial";
        ctx.fillText(potencijal[i][j],BrojUGriduPomaki,BrojUGriduPoma
        kj);
        ctx.font = "bold 9px Arial";
                                    }
    else if (unos == "2") {
        rubniUvjet[i][j] = "n";
        potencijal[i][j] = 0.0;
        ctx.fillStyle = "rgba(167, 30, 6, 1)";
        ctx.fillRect((i * velicinaKvadrata)+1.5, (j *
        velicinaKvadrata)+1.5, velicinaKvadrata-3, velicinaKvadrata-3);
        }
```

```
else if (unos=="3") {
        ctx.fillStyle = "white";
        rubniUvjet[i][j] = "h";
        potencijal[i][j] = 0.00;
        ctx.fillRect((i * velicinaKvadrata)+1.5, (j * velicinaKvadrata)+1.5, velicinaKvadrata-3, velicinaKvadrata-3);
                }
        }
```

Varijabla *unos* ovisi o tome da li je korisnik odabrao definirati Neumannov, Dirichletov rubni uvjet, ili je pak odlučio izbrisati rubni uvjet.

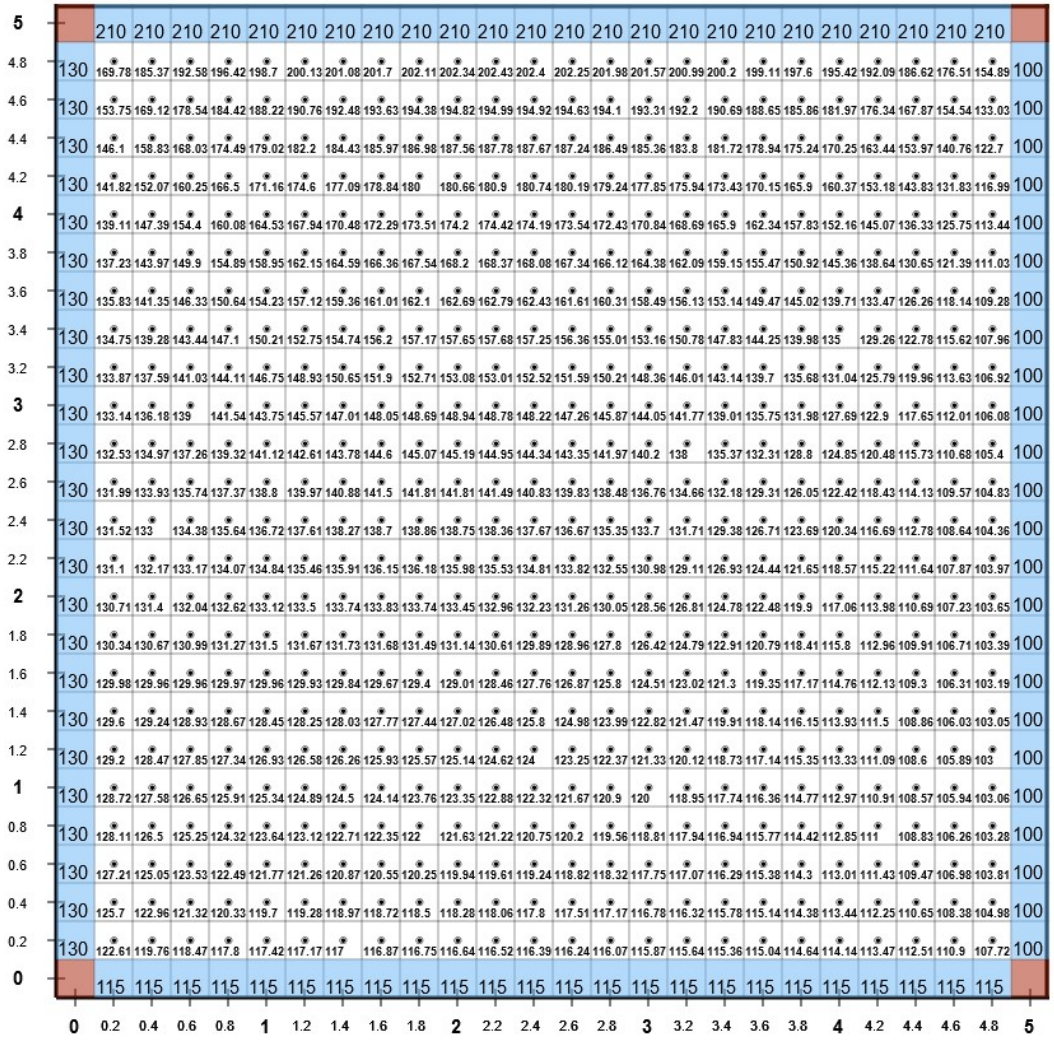
4.3. Vizualizacija rješenja

Svakom iteracijom dobivamo točniju vrijednost za pojedini čvor. Tijek iteracija, te konačno rješenje moguće je prikazati na razne načine te je nužno korisniku omogućiti što veću mogućnost interpretacije podataka.

Omogućio sam u HTML5 elementu *canvas* ispis i vizualizaciju rješenja na sljedeće načine:

1. Ispisati vrijednost potencijala u pojedinom čvoru (vrijednost je zaokružena na dvije decimale).
2. Raspon minimalne i maksimalne vrijednosti potencijala prikazati gradijentom boja, gdje plava boja prikazuje minimalnu, a crvena maksimalnu vrijednost.
3. Prikazati vektorsko polje brzina.
4. Prikazati određene ekvipotencijale.

Objasnit ću svaku pojedinu vizualizaciju na primjeru profila duljine 5, širine 5 s razmakom između čvorova 0,2. Na slici 3. definirani su rubni uvjeti za spomenuti profil te je izračunata krajnja vrijednost svakog čvora profila.



Slika 3. Problem dužine 5, širine 5 s razmakom između čvorova 0,2

4.3.1. HTML5 *canvas* – grafičko programsko sučelje

Canvas element je dio HTML5 jezika te nam omogućava skriptno iscrtavanje i prikazivanje 2D oblika i *bitmap* slika. *Canvas* je početno predstavljen 2004. godine za upotrebu unutar Mac OS X *WebKit* komponente 2004. godine (Hixie, 2004) te se ubrzo proširio na web preglednike. Element *canvas* može se definirati programski JavaScriptom ili direktno u HTML5 kodu atributima *width* i *height*, koji definiraju širinu i visinu *canvas* elementa odnosno prostor namijenjen za crtanje. JavaScript kod može pristupiti *canvas* elementu putem niza funkcija za crtanje te time omogućuje dinamički generiranu grafiku. *Canvas* element se najčešće koristi za izradu grafova, animacija i računalnih igara. Velika prednost *canvas* elementa je upravo njegova kompatibilnost s većinom modernih internetskih preglednika.

4.3.2. Reprezentacija rješenja bojom

Reprezentacija rješenja bojom (engl. *heat map*) je grafička reprezentacija podataka gdje je vrijednost svakog pojedinog čvora reprezentirana odgovarajućom bojom. Za kvalitetnu reprezentaciju bojom potrebno je znati iznos najmanjeg i najvećeg potencijala te na temelju tih informacija napraviti gradijent boje gdje crvena boja odgovara najvećoj vrijednosti pojedinog čvora, a plava boja najmanjoj vrijednosti.

Tokom svake iteracije funkcija *konacneRazlikeIteracija()* izračuna vrijednost najvećeg i najmanjeg elementa u polju *potencijal*, ako i samo ako je to isto pojedino polje definirano slovom *"h"* u polju *rubniUvjet*. Razlikom najvećeg i najmanjeg elementa dobivamo varijablu koju sam nazvao *deltaExtrema* koja služi za definiranje raspona najveće i najmanje vrijednosti potencijala. Funkcija *prikaziVrijednostiBojom()* sadrži dvije *for* petlje, ugniježdene jedna u drugu i primarna joj je zadaća proći kroz svaku vrijednost polja *potencijal* i izračunati iznos varijabli nazvanih *iznosRazNajmanjeg, rgbPostotak, rgbPostotakIntezitet* i *rgbPostotakIntezitetSuprotno,* deklarirati varijablu *bojaString,* te *canvas* funkcijama *fillStyle* i *fillRect* obojati čvor u boju koja odgovara vrijednosti tog čvora.

Varijabla *iznosRazNajmanjeg* je definirana kao razlika trenutnog potencijala i vrijednosti najmanjeg potencijala u trenutnoj iteraciji.

iznosRazNajmanjeg=potencijal[i][j]-najmanji;

Varijabla *rgbPostotak* je definirana kao omjer *iznosRazNajmanjeg* i *deltaExtrema,* drugim riječima, ako je trenutni čvor za koje računamo *rgbPostotak* ujedino i čvor najmanje vrijednosti, varijabla *rgbPostotak* poprimit će vrijednost 0, a čvor najvećeg potencijala imat će vrijednost *rgbPostotak* 1.

rgbPostotak=iznosRazNajmanjeg/deltaExtrema;

Varijabla *rgbPostotakIntezitet* je umnožak varijable *rgbPostotak* i broja 255, zaokružen na cijeli broj.

Varijabla *rgbPostotakIntezitetSuprotno* je razlika broja 255 i varijable *rgbPostotakIntezitet.*

Canvas funkcija *fillStyle* služi za definiranje boje ispune pojedinog polja. Primjer i obrazloženje sintakse *canvas* funkcije *fillStyle*:

ctx.fillStyle = 'rgba(255, 165, 0, 0.55)';

Slova rgba označuje da želimo definirati boju *red-green-blue-alpha* metodom odnosno prva tri broja u zagradi definiraju intenzitet crvene, zelene i plave brojem od 0 do 255, a zadnji broj definira prozirnost boje te je u rasponu između 0 i 1.

Da bi definirao sintaksu *canvas* funkcije *fillStyle* morao sam napraviti varijablu *bojaString* koja generira tekst za funkciju *fillStyle* za svaki čvor pomoću varijabli *rgbPostotakIntezitet* i *rgbPostotakIntezitetSuprotno*. Metodom pokušaja i pogreške odlučio sam definirati varijablu *bojaString* na sljedeći način:

*bojaString="rgba("+Math.round(rgbPostotakIntezitet*1)+","+(Math.round(rgbPostotakIntezitetSuprotno*0.5))+","+Math.round(rgbPostotakIntezitetSuprotno*1) +"," +1+ ")";*

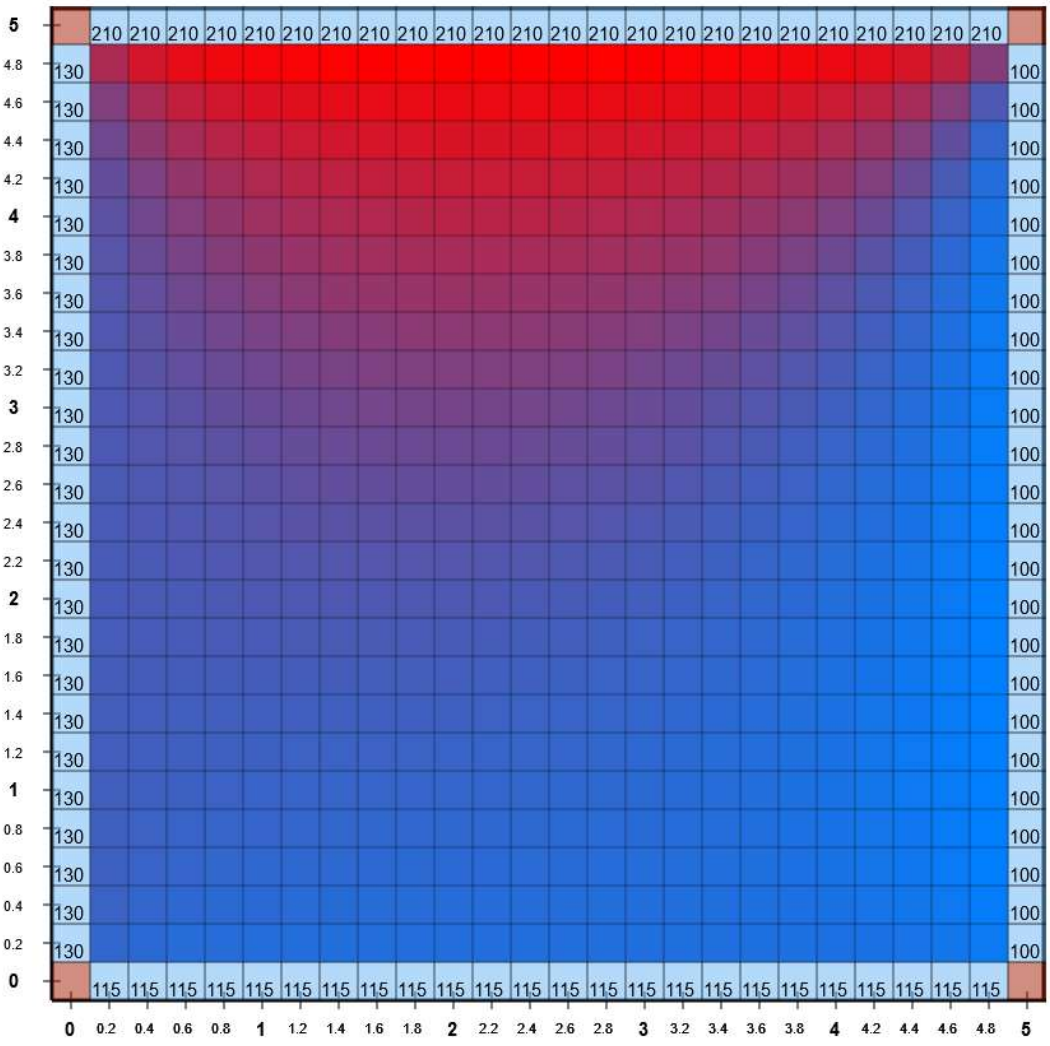
Zadaća varijable *bojaString* je proizvesti pravilnu sintaksu *canvas* funkcije *fillStyle* za svaki čvor *RubniUvjet,* ukoliko taj element ima vrijednost "h", to jest, ukoliko to polje nije Dirichletov ili Neumannov rubni uvjet.

Na kraju svega, funkcijama

ctx.fillStyle = bojaString;

*ctx.fillRect(i * velicinaKvadrata, j * velicinaKvadrata, velicinaKvadrata, velicinaKvadrata);*

bojimo svaki čvor.



Slika 4. Vizualizacija problema visine 5, širine 5 s razmakom između čvorova 0,2 bojom

4.3.3. Prikaz vektorskog polja brzina

Poznavanje potencijala svakog čvora u profilu omogućuje nam da na temelju tih podataka izračunamo i prikažemo vektorsko polje brzina, odnosno vektor brzine za svaki pojedini potencijal. Poznavanjem okolnih čvorova promatranog čvora omogućuje nam izračun iznosa vektora trenutnog čvora u X i Y smjeru, te njihovim zbrajanjem dobivamo smjer, intenzitet i iznos vektorske brzine čvora. Funkcija zadužena za izračun i vizualizaciju pojedinog vektora brzine čvora zove se *pokaziVektore().*

Iskoristio sam istu metodologiju koju sam primijenio i na sam problem rješavanja Laplaceove jednadžbe metodom konačnih razlika da bi definirao varijable *dolje, gore, lijevo, desno* koje odgovaraju pozicijama okolnih čvorova promatranog čvora.

Vektor brzine promatranog čvora definiran je na sljedeći način.

vektorX= -(desno-lijevo)/2

vektorY= -(dolje-gore)/2

Da bi vektor prikazao grafički potrebno je znati njegovu početnu i završnu točku. Njegova početna točka nalazi se u sredini predefiniranog kvadrata koji predstavlja taj određeni čvor u rešetki, a udaljenost završne točke proporcionalna je s vrijednosti okolnih čvorova, drugim riječima bilo je potrebno uvesti algoritam čiji je zadatak izračunati novu varijablu koja bi služila kao referentan iznos prema kojem definiramo najmanji i najveći vektor. Krajnji iznos nove varijable bit će jednak maksimalnom iznosu varijable *vektorX ili vektorY,* ovisno koja varijabla je veća. Ta nova varijabla zove se *vektorMaxAbsolutni* te nam služi kao referentan broj kojeg možemo staviti u omjer svaki puta kada crtamo pojedini vektor da raspon vizualizacije ne bi bio limitiran samo na uzak raspon zadavanja rubnih uvjeta gdje vektori nisu preveliki ili premali.

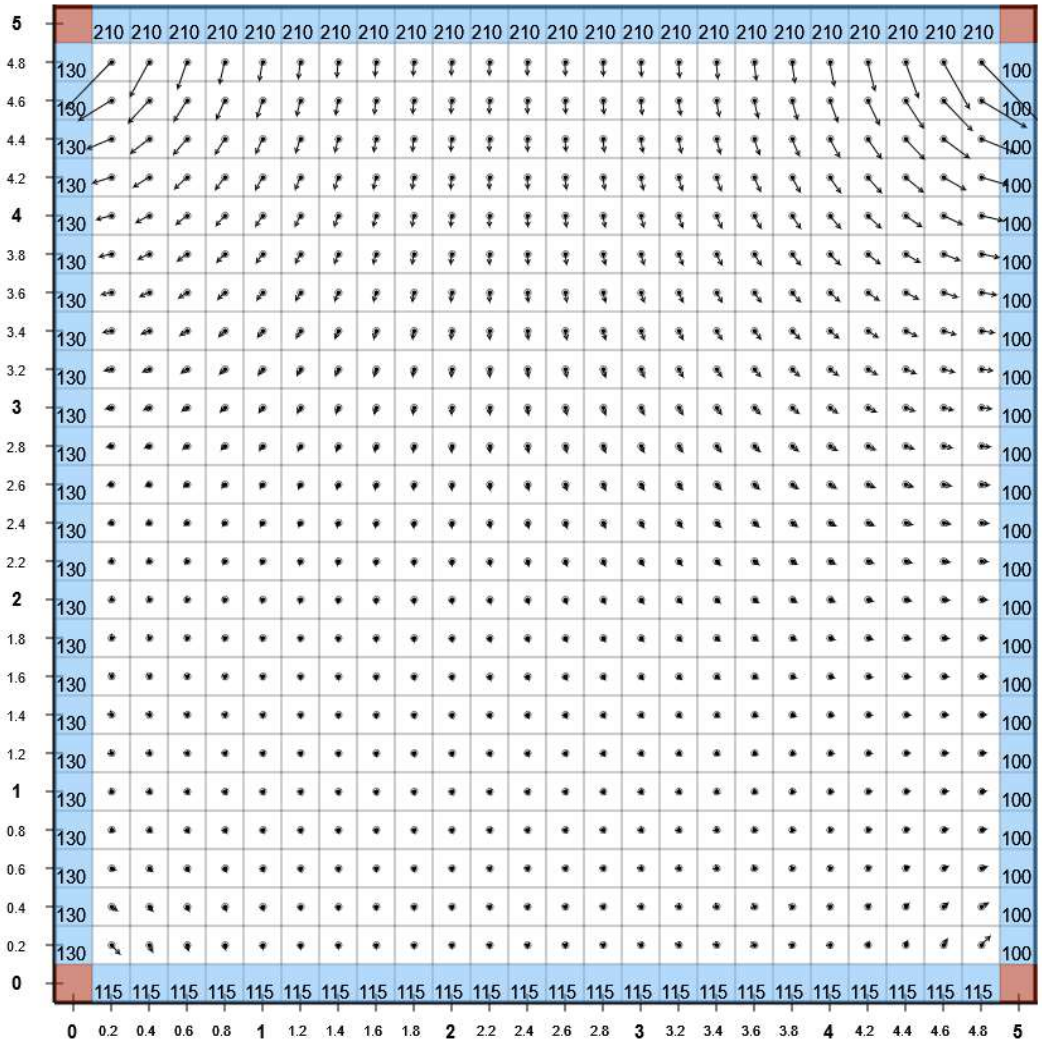
vektorXPostotak=(vektorX/vektorMaxAbsolutni);
vektorYPostotak=(vektorY/vektorMaxAbsolutni);
Canvas element sadrži funkcije *moveTo* i *lineTo* koje omogućuju crtanje ravnih crta no za crtanje vrha vektora potrebno je napisati novu funkciju. Funkciju za vrhove vektora preuzeo sam od (Cieslewski, 2011). Ime funkcije je: *canvas_strelica().*

Prikaz svih argumenata funkcije *canvas_strelica()* tokom njenog pozivanja u funkciji *pokaziVektore()*.

*canvas_strelica(ctx,BrojUGriduPomaki+3,BrojUGriduPomakj-3,BrojUGriduPomaki+3+vektorXPostotak*50,BrojUGriduPomakj-3+vektorYPostotak*50);*

Argumenti funkcije *canvas_strelica()*:

1. Kontekst
2. Lokacija početka vektora po X osi
3. Lokacija početka vektora po Y osi
4. Lokacija završetka vektora po X osi
5. Lokacija završetka vektora po Y osi



Slika 5. Vizualizacija problema visine 5, širine 5 s razmakom između čvorova 0,2 vektorskim poljem brzina

4.3.4. Ekvipotencijale

Ekvipotencijale su krivulje koje spajaju točke jednakih hidrauličkih potencijala. One su sastavni dio strujne mreže kao jedan skup krivulja koji predstavlja rješenje Laplaceove diferencijalne jednadžbe. Ekvipotencijale moraju biti okomite na nepropusne granice.

Program ima mogućnost vizualizacije pojedine ekvipotencijale. Korisnik u HTML element za unos teksta upisuje vrijednost ekvipotencijale koju želi prikazati, te pritiskom na gumb "Prikaži" inicijalizira se funkcija *ekvipotencijale()*. Argument funkcije *ekvipotencijale()* je broj kojeg je korisnik upisao u HTML element za unos teksta te je spremljen u varijablu pod imenom *meta*. Funkcija za svaku točku potencijala stvara varijable *centarInterpolacijei*, *centarInterpolacijej* i *intezitet* na sljedeći način:

```
for (var i = 0; i < brojRedova; i++) {
for (var j = 0; j < brojStupaca; j++) {
                    var
                    centarInterpolacijei=velicinaKvadrata*i+(0.5*velicinaKvadrata-
                    2);
                    var
centarInterpolacijej=velicinaKvadrata*j+(0.5*velicinaKvadrata);
                    var intezitet=potencijal[i][j];
                                                    ...
```

varijable *centarInterpolacijei* i *centarInterpolacijej* služe kao točke oko kojih se provjerava da li je u neposrednoj blizini iznos potencijala koji bi odgovarao broju koji je korisnik upisao u HTML element za unos teksta. Funkcija za svaku točku prolazi kroz niz uvjetnih naredbi (engl. *if statement*) kako bi provjerili da li oko naše trenutne točke potencijala postoji točka potencijala koja je veća ili jednaka varijabli *meta*, te ako je, da inicijalizira funkciju *nasliSmoGa()* koja interpolacijom dolazi do udaljenosti između centaraInterpolacije i točke potencijala *meta.* Ta udaljenost je pospremljena u varijablu nazvanu *udaljenost*. Polja *xMeta* i *yMeta* su polja koja sadržavaju X i Y koordinate svake točke kojoj je iznos potencijala *meta.*

Dio koda funkcije *ekvipotencijale()* koji je zaslužan za pronalazak koordinata točaka iznosa potencijala *meta*:

```
...     if (rubniUvjet[i][j]=="h"){
        if(potencijal[i][j]<=meta){
                if(potencijal[i+1][j]>=meta){
                        var intezitetR=potencijal[i+1][j];
                        nasliSmoGa(intezitetR);
                        xMeta.push(centarInterpolacijei+udaljenost);
                        yMeta.push(centarInterpolacijej);
                                }
                if(potencijal[i-1][j]>=meta){
                         var intezitetL=potencijal[i-1][j];
                        nasliSmoGa(intezitetL);
                        xMeta.push(centarInterpolacijei-udaljenost);
                        yMeta.push(centarInterpolacijej);
                                }
                if(potencijal[i][j+1]>=meta){
                        var intezitetD=potencijal[i][j+1];
                        nasliSmoGa(intezitetD);
                        xMeta.push(centarInterpolacijei);
                        yMeta.push(centarInterpolacijej+udaljenost);
                         }
                 if (potencijal[i][j-1]>=meta){

                        var intezitetU=potencijal[i][j-1];
                        nasliSmoGa(intezitetU);
                        xMeta.push(centarInterpolacijei);
                        yMeta.push(centarInterpolacijej-udaljenost);

                }}}...
```

Kod funkcije *nasliSmoGa():*

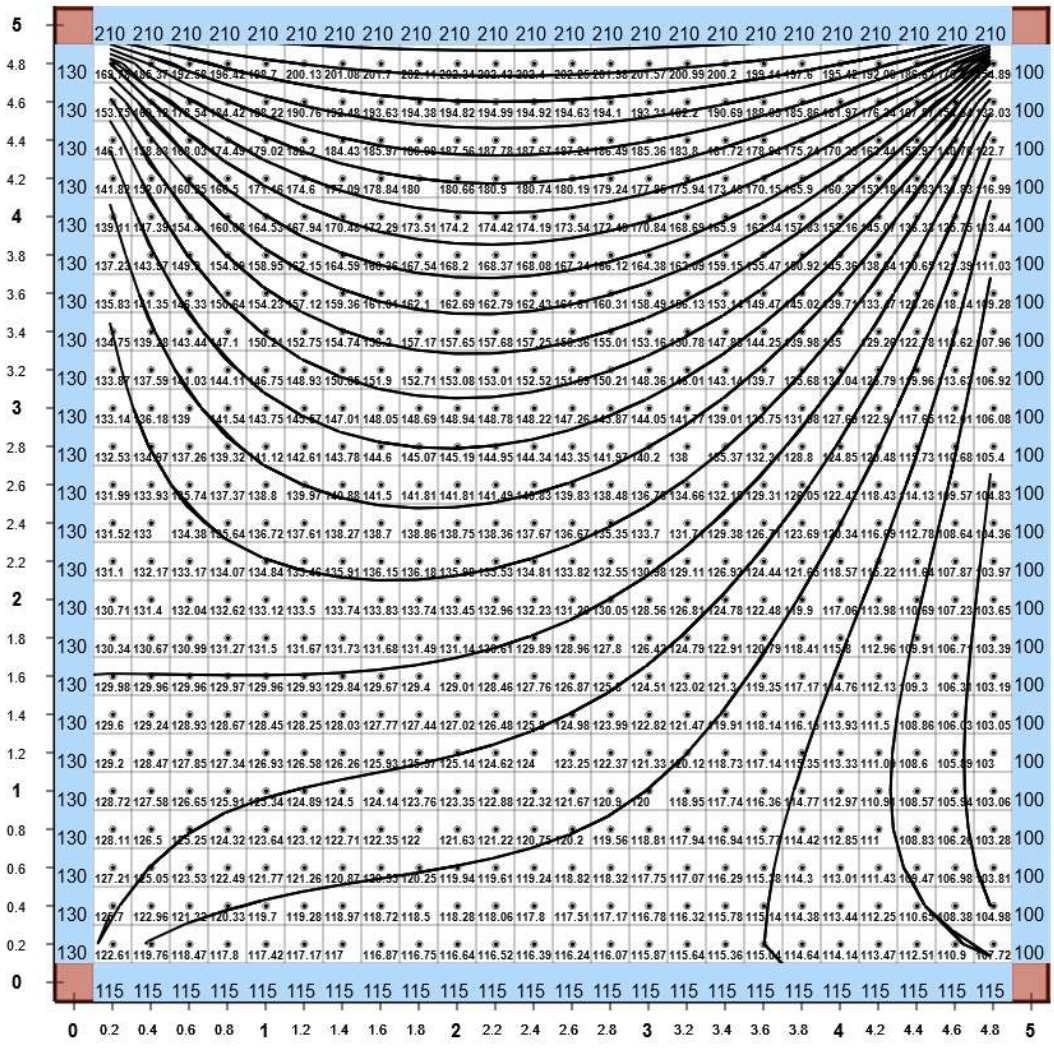
```
function nasliSmoGa(smjer){
        razlikaPolja=Math.abs(smjer-intezitet);
```

```
razlikaMeteIPolja=meta-intezitet;
omjer=razlikaMeteIPolja/razlikaPolja;
udaljenost=velicinaKvadrata*omjer;
        }
```

Funkcija *nasliSmoGa()* nalazi se unutar funkcije *ekvipotencijale()* kao i funkcija *crtaj()* koja služi za pravilno spajanje točaka spremljenih u poljima *xMeta* i *yMeta*. Kod funkcije *crtaj()*:

```
function crtaj(){
        ctx.beginPath();
        var prviBroj=1;
    for (var i = 0; i < xMeta.length; i++) {
    for (var ii = 0; ii < xMeta.length; ii++) {
        var xUdaljenostSuseda=Math.abs(xMeta[i]-xMeta[ii])
        var yUdaljenostSuseda=Math.abs(yMeta[i]-yMeta[ii]);
    //za bolju verziju nekad
        var formulaUdaljenost=Math.abs(Math.sqrt(Math.pow(xMeta[i]-
        xMeta[ii],2)+Math.pow(yMeta[i]-yMeta[ii],2)));
        if ( spojilaSe[i]=="da" && spajaloSeUNju=="da"){}
    //
        else {
            if (xUdaljenostSuseda<(velicinaKvadrata+(1)) &&
        yUdaljenostSuseda<(velicinaKvadrata+1) ){
                ctx.beginPath();
                ctx.lineWidth = 2;
                ctx.fillStyle="black"
                spojilaSe[i]="da"
                spajaloSeUNju[ii]="da";
                ctx.moveTo(xMeta[i], yMeta[i]);
                ctx.lineTo(xMeta[ii], yMeta[ii]);
                ctx.stroke();
                    }}}}}
```

Valja napomenuti da funkcija *crtaj()* nije optimalno napisana te još ima prostora za poboljšanja.



Slika 6. Vizualizacija problema visine 5, širine 5 s razmakom između čvorova 0,2 ekvipotencijalama ekvidistance 5.

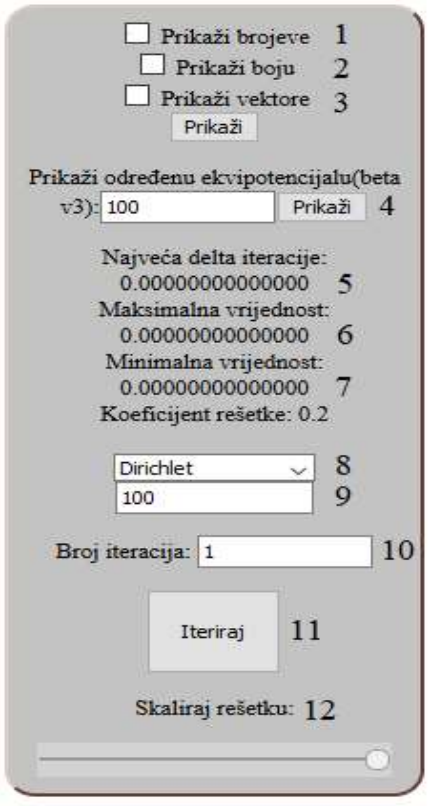
5. Upute za korištenje programa

Kako bi program funkcionirao na predviđeni način potrebno je definirati parametre presjeka te rubne uvjete u profilu. Parametri presjeka definiraju se na sljedeći način:

1. Unos visine presjeka
2. Unos širine presjeka
3. Definiranje broja čvorova po jedinici duljine

Napomena: Mogućnost unosa parametara rešetke odnosno presjeka limitirana samo na cijele brojeve.

Nakon uspješno definiranog presjeka, korisniku se prikazuje HTML5 *canvas* element s iscrtanom rešetkom profila, zajedno sa trakom elemenata za interakciju s korisnikom koja je čvrsto „zalijepljena“ na lijevu stranu prozora internetskog preglednika.



Sadržaj trake s elementima za interakciju s korisnikom:

1. Prikaz iznosa potencijala pojedinog čvora
2. Prikaz vizualizacije rješenja bojom
3. Prikaz vektorskog polja brzina
4. Prikaz određene ekvipotencijale
5. Najveća razlika iteracije
6. Maksimalna vrijednost čvora
7. Minimalna vrijednost čvora
8. Odabir rubnog uvjeta
9. Upis iznosa Dirichletovog rubnog uvjeta
10. Broj iteracija
11. Gumb "Iteriraj"
12. Mogućnost smanjenja i povećanja rešetke

Slika 7. Traka s elementima za interakciju s korisnikom

Da bi definirali rubne uvjete potrebno je:

1. Kliknuti na rubni uvjet koji želimo definirati (Dirichlet, Neumann ili Očisti rubni uvjet) (element 8)
2. Upisati iznos rubnog uvjeta (Ukoliko definiramo Dirichletov rubni uvjet) (element 9)
3. Pritisnutim klikom miša povlačimo strelicu po elementu *canvas*, po čvorovima na kojima želimo definirati rubni uvjet

Napomena: Na rubovima presjeka moraju biti zadani Dirichletovi ili Neumannovi rubni uvjeti. Unaprijed je zadan Neumannov rubni uvjet (smeđi kvadrati), a korisnik ga može zamijeniti Dirichletovim (plavi kvadrati).

Gumbom "Iteriraj" (element 11) iteriramo rješavanje problema metodom konačnih razlika onoliko puta koliko je korisnik definirao upisom broja iteracija (element 10).

Program omogućuje korisniku prikaz pojedine ekvipotencijale upisom vrijednosti ekvipotencijale te klikom na gumb "Prikaži" (element 4). Korisnik ima mogućnost unijeti bilo koju vrijednost ekvipotencijale, to jest nije limitiran na unos cijelih brojeva.

Ostale napomene: Korisniku je omogućen ispis točnog iznosa pojedinog potencijala dvoklikom na određeni čvor.

6. Testiranje – usporedba s analitičkim rješenjem

Programi koji izvode numeričke proračune obično su vrlo složeni i zbog velike opasnosti od pogrešaka potrebno ih je prije upotrebe temeljito testirati. U 7. poglavlju usporedit ćemo vizualna rješenja sa SEEP/W programskim paketom. No ključni je test usporedba s nekim jednostavnijim problemom koji se može egzaktno analitički riješiti.

Ukoliko program ima poteškoća s reprodukcijom analitičkog rješenja to ukazuje na ozbiljne pogreške u metodi ili jednu ili više pogrešaka tijekom implementacije metode u kod.

Usporedba s jednostavnim analitičkim rješenjem važan je i nezaobilazan korak razvoja programa.

6.1. Analitičko rješenje

Stacionarno vođenje topline u trodimenzionalnom Kartezijevom koordinatom sustavu, uz nepromjenjiviju konstantu toplinske vodljivosti opisuje Laplaceova jednadžba (Pitts i Sissom, 1998)

$$\frac{\partial^2 \theta}{\partial x^2} + \frac{\partial^2 \theta}{\partial y^2} + \frac{\partial^2 \theta}{\partial z^2} = 0$$

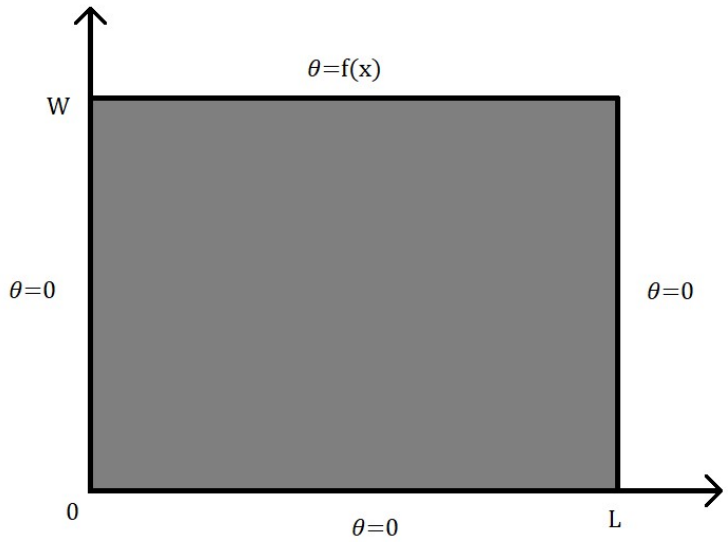
Ukoliko proučavamo neki problem u kojem je jedna dimenzija daleko veća od drugih, temperaturni gradijent u tom smjeru je zanemariv. Primjer iz svakodnevnog života bila bi šipka kojoj je dimenzija u smjeru z mnogo veća nego u ostalim smjerovima.

Budući da je $\frac{\partial^2 \theta}{\partial z^2} = 0$ problem se svodi na nalaženje 2D polja temperature $\theta(x, y)$ na presjeku šipke.

Mora vrijediti:

$$\frac{\partial^2 h}{\partial x^2} + \frac{\partial^2 h}{\partial y^2} = 0 \qquad (1)$$

6.1.1. Opis problema



Slika 8. Presjek šipke

Na tri strane temperatura je $\theta = 0$, a na gornjoj strani je neka proizvoljna funkcija od x: $\theta = f(x)$

Rubni uvjeti:

(I) Lijeva strana: $\theta(0, y) = 0 \ (0 < y < W)$

(II) Desna strana: $\theta(L, y) = 0 \ (0 < y < W)$

(III) Donja strana: $\theta(x, 0) = 0 \ (0 < x < L)$

(IV) Gornja strana: $\theta(x, W) = f(x) \ (0 < y < W)$

6.1.2. Razdvajanje (separacija) varijabli

Pretpostavimo da polje temperature $\theta(x, y)$ ima rješenje oblika

$$\theta(x, y) = X(x)Y(y) \qquad (2)$$

Uvrštavanjem izraza (2) u Laplaceovu jednadžbu (1) dobivamo:

$$Y(y)\frac{d^2X(x)}{dx^2} + X(x)\frac{d^2Y(y)}{dy^2} = 0$$

Kad prebacimo i presložimo tako da x bude na jednoj a y na drugo strani dobivamo:

$$\frac{1}{X(x)}\frac{d^2X(x)}{dx^2} = -\frac{1}{Y(y)}\frac{d^2Y(y)}{dy^2}$$

Lijeva strana ovisi samo o x, a desna samo o y koordinati. S obzirom da su x i y neovisni, izraz bude jednak samo ako su obje strane jednake nekoj istoj konstanti

$$\frac{1}{X(x)}\frac{d^2X(x)}{dx^2} = -\frac{1}{Y(y)}\frac{d^2Y(y)}{dy^2} = konstanta.$$

U slučajevima kad je $konstanta = 0$ ili $konstanta > 0$, jednadžbu nije moguće uskladiti s rubnim uvjetima (Pitts i Sissom, 1998).

U slučaju da je $konstanta < 0$ uzimamo da je $konstanta = -\lambda^2$ pri čemu je $\lambda \in R$ i slijedi:

$$\frac{1}{X(x)}\frac{d^2X(x)}{dx^2} = -\frac{1}{Y(y)}\frac{d^2Y(y)}{dy^2} = -\lambda^2$$

Dobivamo dvije obične diferencijalne jednadžbe:

$$\frac{d^2X(x)}{dx^2} = -\lambda^2X(x) \text{ i } \frac{d^2Y(y)}{dy^2} = \lambda^2Y(y)$$

Obična diferencijalna jednadžba

$$\frac{d^2X(x)}{dx^2} = -\lambda^2X(x)$$

je ekvivalentna jednadžbi harmonijskog oscilatora, ali s x, umjesto t kao parametrom, što se bolje vidi ako je napišemo na uobičajeni način.

$$\frac{d^2X(x)}{dx^2} + \lambda^2X(x) = 0$$

Rješenja jednadžbe su harmonijske funkcije:

$$X(x) = C_1 \sin \lambda x + C_2 \cos \lambda x$$

Rješenje diferencijalne jednadžbe

$$\frac{d^2Y(y)}{dy^2} = \lambda^2Y(y)$$

je funkcija koja je nakon dva deriviranja opet ista ta funkcija, iz čega zaključujemo da je to eksponencijalna funkcija, općenito:

$$Y(y) = C_3 e^{\lambda y} + C_4 e^{-\lambda}$$

Spajanje ta dva izraza vodi do rješenja Laplaceove jednadžbe oblika

$$\theta(x,y) = X(x)Y(y) = (C_1 \sin \lambda x + C_2 \cos \lambda x)(\, C_3 e^{\lambda y} + C_4 e^{-\lambda y}) \qquad (3)$$

6.1.3. Usklađivanje s rubnim uvjetima

Gore definirane rubne uvjete potrebno je ubaciti u rješenje da bismo provjerili da li ga je moguće uskladiti s zadanim rubnim uvjetima i odredili konstante C_1, C_2, C_3 i C_4.

Rubni uvjet (I): $\theta(0,y) = 0$

$$(C_1 \sin 0 + C_2 \cos 0)\big(C_3 e^{\lambda y} + C_4 e^{-\lambda}\ \big) = C_2\big(C_3 e^{\lambda y} + C_4 e^{-\lambda y}\big) = 0 \qquad (4)$$

Faktor $\left(C_3 e^{\lambda y} + \mathrm{C}_4 e^{-\lambda y}\right)$ mora biti različit od nule jer odabir $C_3 = C_4 = 0$ vodi ka trivijalnom rješenju $\theta(x,y) = 0$, tj. temperatura je nula na čitavom presjeku, a to se ne može uskladiti s rubnim uvjetom (IV). Ako pak pretpostavimo da su konstante C_3 i C_4 različite od nule, tada iz $C_3 e^{\lambda y} + C_4 e^{-\lambda y} = 0$ slijedi $\frac{C_4}{C_3} = -e^{2\lambda y}$, tj. omjer konstanti ovisi o koordinati y što je nemoguće ako su C_3 i C_4 doista konstante. Dakle, izraz (4) može biti zadovoljen jedino ako je $C_2 = 0$ pa rubni uvjet (I) može biti zadovoljen jedino rješenjem oblika

$$\theta(x,y) = C_1\big(C_3 e^{\lambda y} + C_4 e^{-\lambda y}\big) \sin \lambda x.$$

Rubni uvjet (II): $\theta(L,y) = 0$

$$C_1\big(C_3 e^{\lambda y} + C_4 e^{-\lambda y}\big) \sin \lambda L = 0$$

Odabir $C_1 = 0$ opet vodi do trivijalnog rješenja $\theta(x,y) = 0$ kojeg je nemoguće uskladiti s rubnim uvjetom (IV). Preostaje da mora biti $\sin \lambda L = 0$ iz čega slijedi da je $\lambda L = n\pi$, $n \in Z$. Rješenje koje zadovoljava rubne uvjete (I) i (II) je oblika

$$\theta(x,y) = C_1\left(C_3 e^{\frac{n\pi y}{L}} + C_4 e^{-\frac{n\pi y}{L}}\right)\sin\frac{n\pi x}{L}$$

Rubni uvjet (III): $\theta(x,0) = 0$

$$C_1(C_3 e^0 + C_4 e^0) = C_1(C_3 + C_4)\sin\frac{n\pi x}{L} = 0$$

Odabir $C_1 = 0$ opet vodi na $\theta(x,y) = 0$, $\sin\frac{n\pi}{L} = 0$ vrijedi samo kad je $\frac{x}{L} \in \mathrm{Z}$, dakle preostaje samo da mora vrijediti $C_3 + C_4 = 0$ iz čega slijedi da je $C_4 = -C_3$ i

$$\theta(x,y) = C_1 C_3\left(e^{\frac{n\pi y}{L}} - e^{-\frac{n\pi y}{L}}\right)\sin\frac{n\pi x}{L}.$$

Sinus hiperbolički definira se kao: $\sinh(\varphi) = \frac{e^{\varphi} - e^{-\varphi}}{2}$

Slijedi:

$$\theta(x,y) = 2C_1 C_3 \sinh\frac{n\pi y}{L}\sin\frac{n\pi x}{L}$$

$$C_n = 2C_1 C_3$$

$$\theta(x,y) = C_n \sinh\frac{n\pi y}{L}\sin\frac{n\pi x}{L}$$

za svaki cijeli broj n.

6.1.4. Superpozicija rješenja

Uvrštavanjem rubnih uvjeta (I), (II) i (III) dobili smo da je

$$\theta(x,y) = C_n \sinh\frac{n\pi y}{L}\sin\frac{n\pi x}{L}$$

rješenje Laplaceove jednadžbe za svaki n koji je element skupa cijelih brojeva.

Međutim, za linearne i homogene diferencijalne jednadžbe suma rješenja je također rješenje, dakle i sljedeći beskonačni red je rješenje Laplaceove jednadžbe (1)

$$\theta(x,y) = \sum_{n=1}^{\infty} C_n \sinh\frac{n\pi y}{L}\sin\frac{n\pi x}{L}.$$

Sumaciju počinjemo s $n = 1$ jer je izraz $\sin\frac{0\pi x}{L}$ jednak nuli, a negativne vrijednosti n su nepotrebne jer imamo produkt dviju neparnih funkcija:

$$\sinh(-\varphi)\sin(-\varphi) = [-\sinh(\varphi)]\,[-\sin(\varphi)] = \sinh(\varphi)\sin(\varphi).$$

6.1.5. Rubni uvjet (IV)

Rubni uvjet (4): $\theta(x, W) = f(x)$

$$\theta(x, W) = \sum_{n=1}^{\infty} C_n \sinh\frac{n\pi W}{L}\sin\frac{n\pi x}{L} = f(x)$$

$$f(x) = \sum_{n=1}^{\infty} C_n \sinh\frac{n\pi W}{L}\sin\frac{n\pi x}{L} \quad (5)$$

Korištenjem trigonometrijskih transformacija može se izračunati

$$\int_0^L \sin\frac{n\pi x}{L}\sin\frac{m\pi x}{L}\,dx = \frac{L}{2}\delta_{mn}$$

što znači da je integral jednak $\frac{L}{2}$ kada je $m = n$, a za različite m i n jednak je 0 (simbol δ_{mn} je Kroneckerov delta).

Projicirat ćemo m-ti član iz (5) tako da pomnožimo obje strane sa $\sin\frac{m\pi x}{L}$ i integriramo po x od 0 do L:

$$\int_0^L f(x)\,\sin\frac{m\pi x}{L}\,dx = \sum_{n=1}^{\infty} C_n \sinh\frac{n\pi W}{L}\frac{L}{2}\delta_{mn} = C_m\frac{L}{2}\sinh\frac{m\pi W}{L}$$

Iz čega slijedi:

$$C_m = \frac{2}{L}\frac{1}{\sinh\frac{m\pi W}{L}}\int_0^L f(x)\sin\frac{m\pi x}{L}\,dx$$

Konačno rješenje Laplaceove jednadžbe uvažajući sva četiri rubna uvjeta:

$$\theta(x, y) = \frac{2}{L}\sum_{n=1}^{\infty}\frac{\sinh\frac{n\pi y}{L}}{\sinh\frac{n\pi W}{L}}\sin\frac{n\pi x}{L}\int_0^L f(x)\sin\frac{n\pi x}{L}\,dx \qquad (6)$$

Dakle, ukoliko znamo riješiti integral $\int_0^L f(x) \sin\frac{n\pi x}{L} dx$, možemo dobiti analitičko rješenje u obliku beskonačnog reda.

6.1.6. Slučaj kad je *f(x)* konstanta

Rješavanjem integrala $\int_0^L f(x) \sin\frac{n\pi x}{L} dx$ za slučaj $f(x) = \theta_c = konstanta$ dobivamo:

$$\int_0^L \theta_c \sin\frac{n\pi x}{L} dx = \theta_c \int_0^L \sin\frac{n\pi x}{L} dx$$

$$\mathrm{t} = \frac{n\pi}{L} x, \quad \mathrm{dt} = \frac{n\pi}{L} dx, \quad \mathrm{dx} = \frac{\mathrm{L}}{n\pi} dt$$

$$\text{donja granica: } \mathrm{t} = \frac{n\pi}{L} 0 = 0$$

$$\text{gornja granica: } \mathrm{t} = \frac{n\pi}{L} L = n\pi$$

Iz čega slijedi:

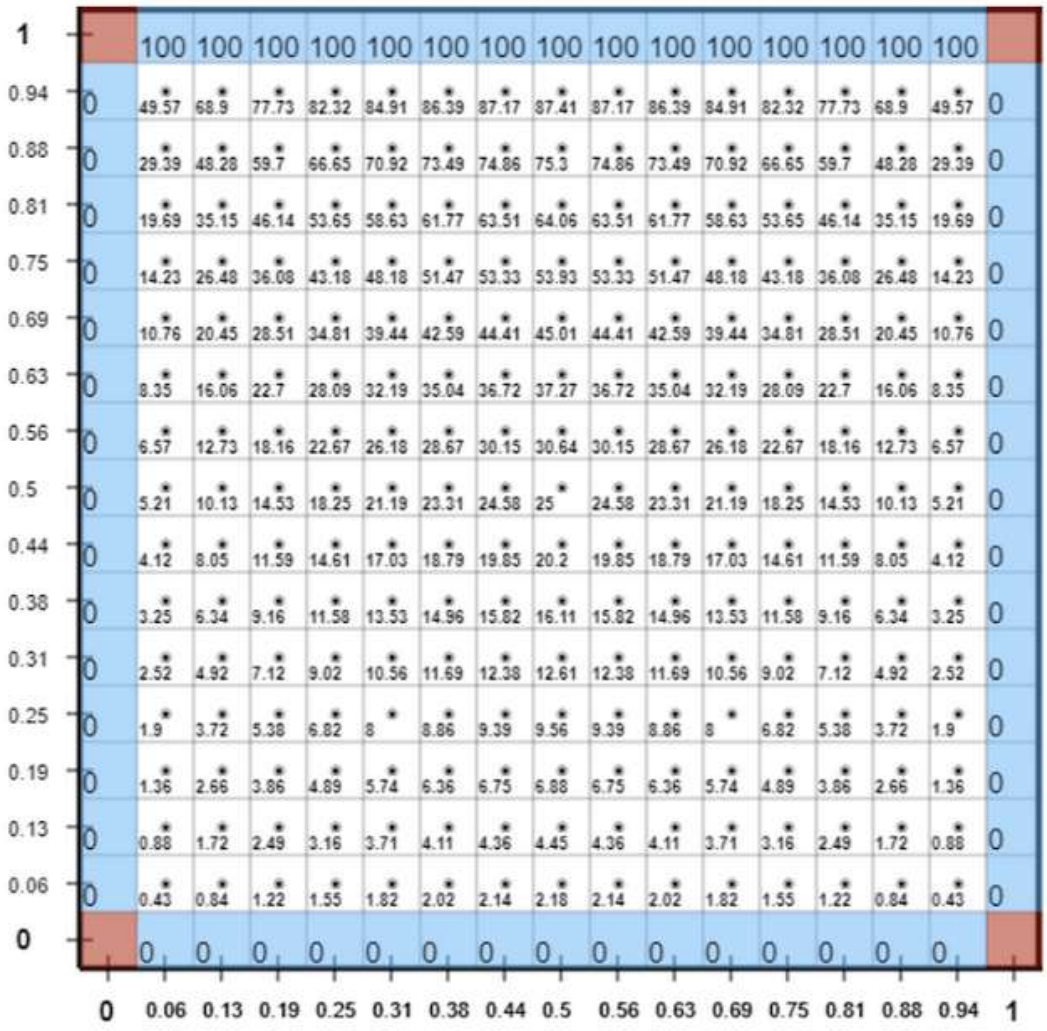
$$\frac{\theta_c L}{n\pi} \int_0^{n\pi} \sin t \;\; dt = \frac{\theta_c L}{n\pi} (\cos 0 - \cos n\pi) = \theta_c \frac{L}{\pi} \frac{(-1)^{n+1} + 1}{n} \qquad (7)$$

Preostaje uvrstiti (7) u rješenje (6) te dobivamo:

$$\theta(x, y) = \theta_c \frac{2}{\pi} \sum_{n=1}^{\infty} \frac{-1^{n+1} + 1}{n} \frac{\sinh\frac{n\pi y}{L}}{\sinh\frac{n\pi W}{L}} \sin\frac{n\pi x}{L} \qquad (8)$$

6.2. Konvergencija metode konačnih razlika

Da bi ustanovili kako odstupanja od analitičkog rješenja ovise o finoći rešetke, tj. o razmaku između čvorova, proveli smo usporedbu rješenja iz programa s različito definiranim razmakom čvorova s analitičkim rješenjem (8) za $\theta_c = 100$, dimenzije presjeka $L = W = 1$ i točku $(\frac{1}{4}, \frac{3}{4})$, vidjeti Sliku 9.

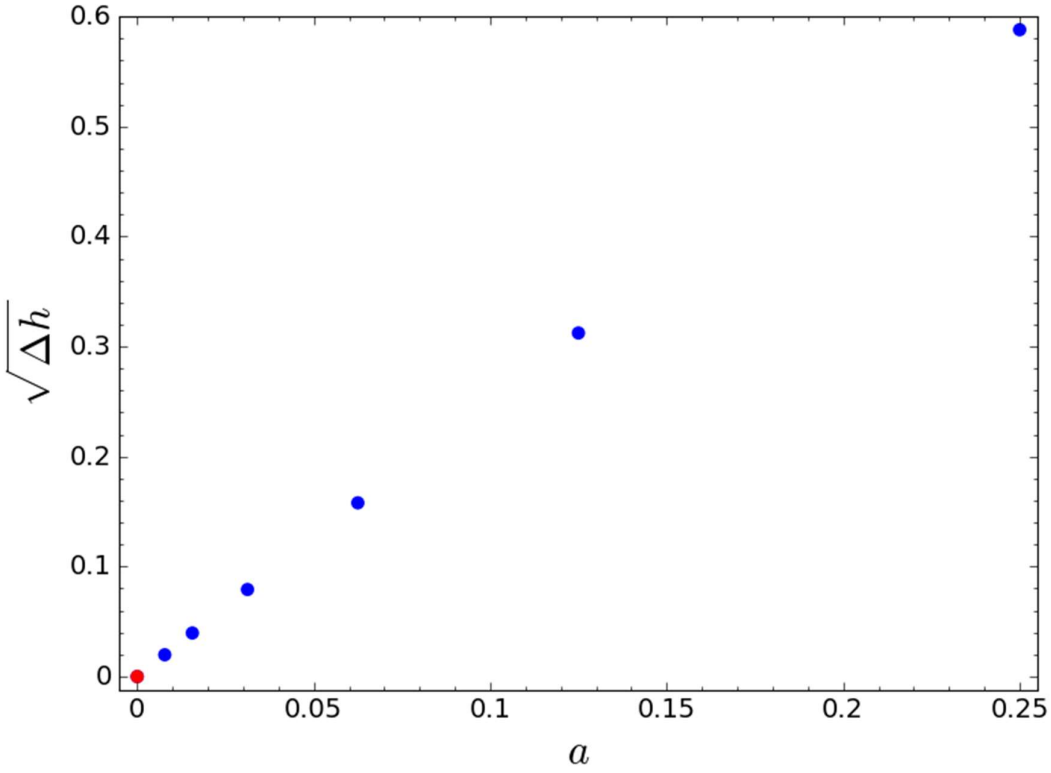


Slika 9. Problem definiran u programu, s razmakom između čvorova 0,0625

Analitičko rješenje je izračunato zbrajanjem prvih 20 neparnih članova reda korištenjem programa SageMath 8.3., a u našem programu problem je riješen metodom konačnih razlika s razmacima između čvorova: 0,25, 0,125, 0,0625, 0,03125, 0,015625 i 0,0078125.

Broj čvorova po jedinici duljine	Razmak između čvorova a	Vrijednost potencijala	Odstupanje Δh	Omjer odstupanja
4	0,25	42,8571429	0,3456903	
8	0,125	43,1053057	0,0975275	3,5445419
16	0,0625	43,1778991	0,0249341	3,9114108
32	0,03125	43,1965711	0,0062621	3,9817513
64	0,015625	43,2012660	0,0015672	3,9956981
128	0,0078125	43,2024413	0,0003919	3,9989408
analitičko rješenje		43,2028332	0	

Tablica 1. Odnos odstupanja od analitičkog rješenja Δh i razmaka između čvorova a



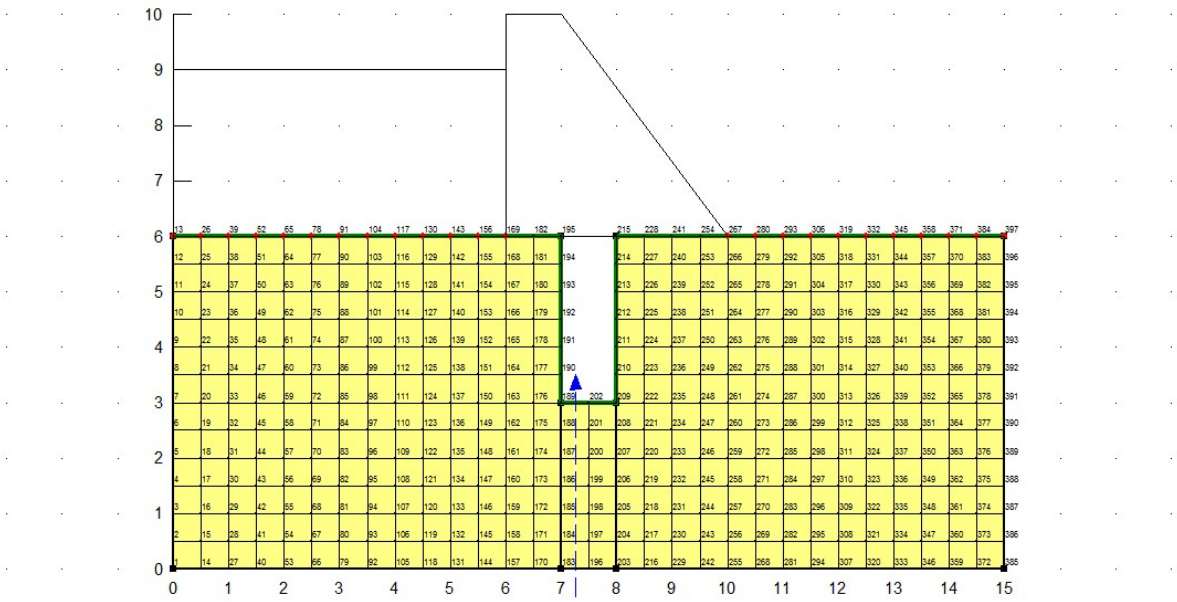
Graf 1. Odnos razmaka između čvorova a i korijena vrijednosti odstupanja Δh

Kao što je vidljivo iz grafikona 1. odnos korijena iznosa vrijednosti odstupanja $\sqrt{\Delta h}$ i razmaka između čvorova a je gotovo linearan, tj. $\sqrt{\Delta h}$ je proporcionalan s a, što znači da je odstupanje Δh proporcionalno a^2, potpuno u skladu s razmatranjem u potpoglavlju 3.1.

Dakle, ako se prepolovi razmak između čvorova dobiva se četiri puta manje odstupanje i kako se lijepo vidi iz grafikona 1. numerička rješenja (plave točke) konvergiraju prema analitičkom rješenju (crvena točka).

7. Usporedba s drugim programskim rješenjima

Kako bi usporedio točnost izračuna svog programa odlučio sam usporediti njegova rješenja s rješenjima identično definiranog profila u SEEP/W programskom paketu.



Slika 10. Skica numeričkog modela gravitacijske brane (Stanojević, 2012)

Crvene točke na slici označavaju Dirichletov potencijal.

- Uzvodna strana brane

  H = 9 m;

  u/γ_w(tlačna visina) = 3 m

  z (geodetska visina) = 6 m

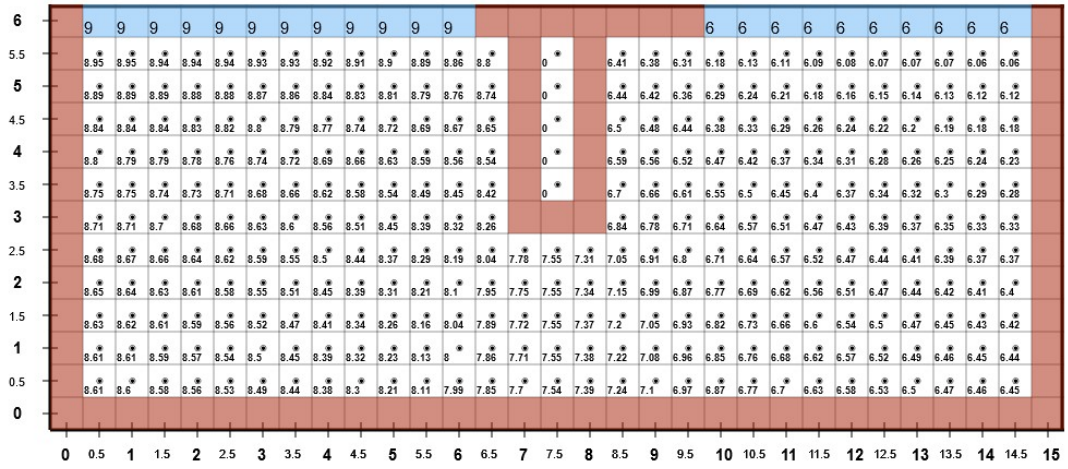
- Nizvodna strana brane

  H = 6 m;

  u/γ_w(tlačna visina) = 0 m

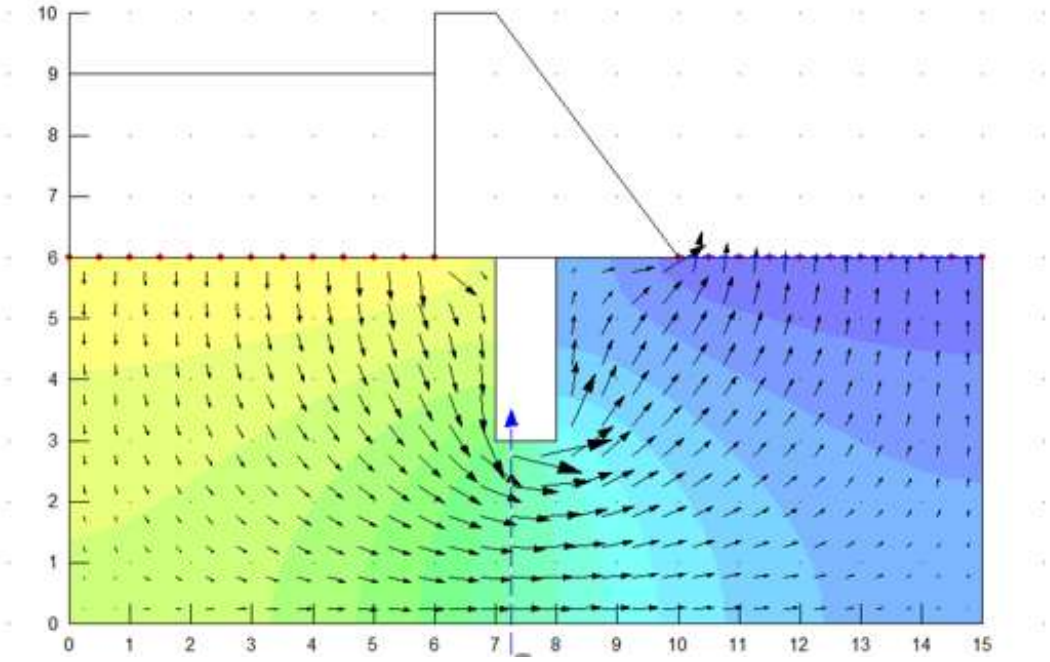
  z (geodetska visina) = 6 m

Na slici 11 možemo vidjeti taj isti profil definiran u našem programu s razmakom između čvorova 0,5 metara.

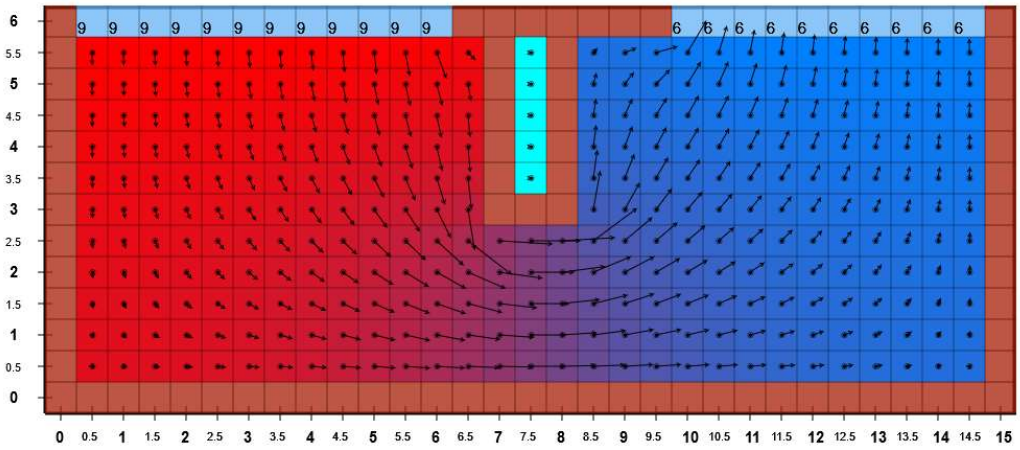


Slika 11. Profil modela gravitacijske brane

Usporedba rješenja vektorskog polja brzina:

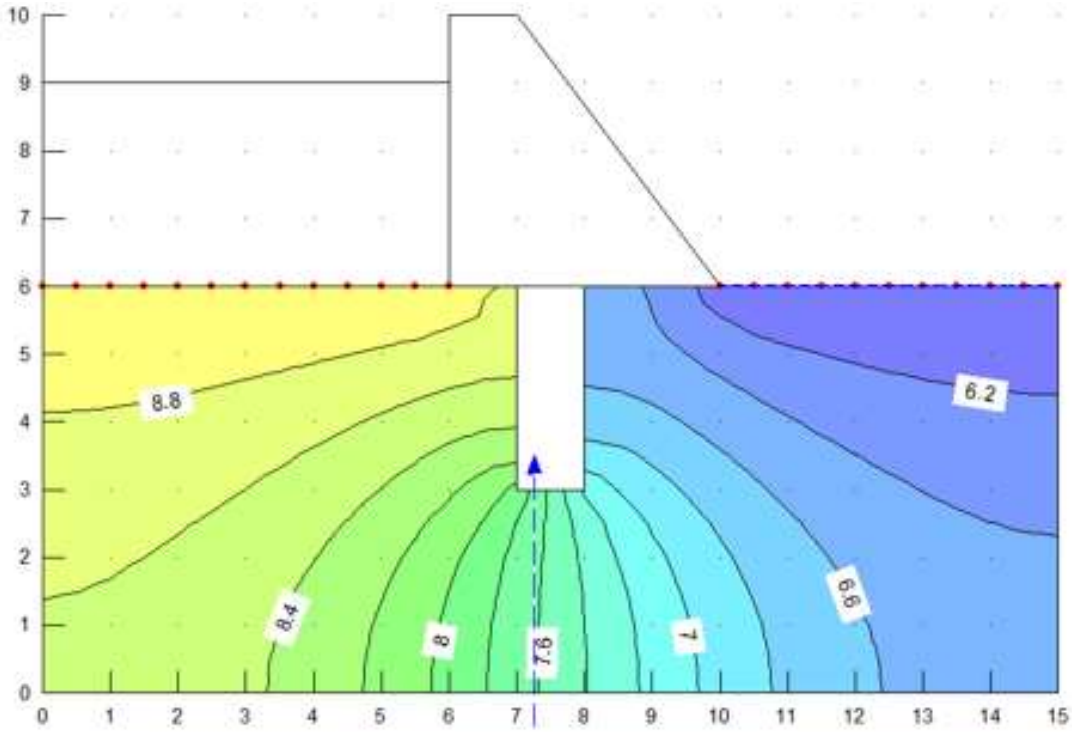


Slika 12. Prikaz vektora brzine toka (Stanojević, 2012)

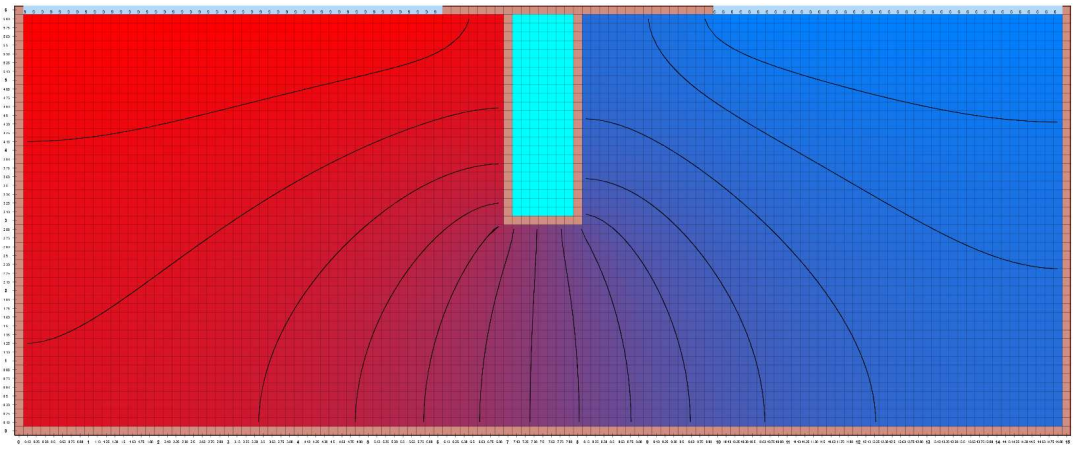


Slika 13. Prikaz vektora brzine toka (razmak između čvorova: 0,5)

Usporedba ekvipotencijala:



Slika 14. Ekvipotencijale(Stanojević, 2012)



Slika 15. Ekvipotencijale (razmak između čvorova: 0,125)

8. Zaključak

Rad je započet s ciljem razvoja programa za numeričko rješavanje i vizualizaciju Laplaceove jednadžbe koji bi bio jednostavan i dostupan svima. HTML5 element *canvas* omogućuje vizualizaciju rješenja u bilo kojem modernom web pregledniku.

Upravo zbog svoje lake dostupnosti putem web preglednika i jednostavnosti program je prikladan za korištenje u edukativne svrhe. Studenti ga s lakoćom mogu koristiti kako bi stekli intuiciju za rješenja Laplaceove jednadžbe i odgovarajućih problema iz raznih područja znanosti i tehnike kao što su vođenje topline i stacionarno tečenje vode u tlu.

Usporedbom rezultata programa s analitičkim rješenjem i rješenjima drugih programskih paketa potvrđeno je da je program ispravno implementiran te je postavljeni cilj ostvaren.

9. Popis literature

Burden, R.L., Faires, J.D. (1989). *Numerical Analysis*, 4th Ed. Boston: PWS-KENT Publishing Company

Cieslewski, T. (13.06.2011.) *Draw arrow on canvas tag* [Online] Dostupno na: https://stackoverflow.com/questions/808826/draw-arrow-on-canvas-tag [09.08.2018.]

Hip, I. (2014). *Dinamika fluida.* Varaždin: Sveučilište u Zagrebu, Geotehnički fakultet (skripta).

Hixie, I. (12.07.2004.) Extending HTML [Online] Dostupno na: http://ln.hixie.ch/?start=1089635050 [30.07.2018.]

Jović, V. (2006). Osnove hidromehanike. Zagreb: Element. 9. pogl.

Lang, C.B., Pucker N. (2005). Mathematische Methoden in der Physik, 2. Auflage. München: Elsevier GmbH (Spektrum Akademischer Verlag)

Pitts, D.R.; Sissom, L.E. (1998). Heat Transfer, 2nd Ed. New York: McGraw-Hill / Schaum's Outline Series.

Powers, S. (2010). *Naučite JavaScript.* 2. izdanje. Zagreb: Dobar plan (O'Reilly).

Press, W.H. et al. (1992) *Numerical Recipes in FORTRAN*, 2nd Ed. Cambridge: Cambridge University Press

Purcell, E.M. (1988). Elektricitet i magnetizam. Zagreb: Tehnička knjiga. 2. pogl.

Stanojević I. (2012). *PROGRAMSKI PAKET SEEP/W.* Završni rad. Varaždin: Sveučilište u Zagrebu, Geotehnički fakultet.

10. Popis slika

Slika 1. Grafičko korisničko sučelje našeg programa

Slika 2. Prikaz elemenata za interakciju s korisnikom

Slika 3. Problem dužine 5, širine 5 s razmakom između čvorova 0,2

Slika 4. Vizualizacija problema dužine 5, širine 5 s razmakom između čvorova 0,2 bojom

Slika 5. Vizualizacija problema dužine 5, širine 5 s razmakom između čvorova 0,2 vektorskim poljem brzina

Slika 6. Vizualizacija problema visine 5, širine 5 s razmakom između čvorova 0,2 ekvipotencijalama ekvidistance 5.

Slika 7. Traka s elementima za interakciju s korisnikom

Slika 8. Presjek šipke

Slika 9. Problem definiran u programu, s razmakom između čvorova 0,0625

Tablica 1. Odnos odstupanja Δh i razmaka između čvorova a

Graf 1. Odnos razmaka između čvorova a i korijena vrijednosti odstupanja Δh

Slika 10. Skica numeričkog modela gravitacijske brane (Stanojević 2012)

Slika 11. Profil modela gravitacijske brane

Slika 12. Prikaz vektora brzine toka (Stanojević 2012)

Slika 13. Prikaz vektora brzine toka (razmak između čvorova: 0,5)

Slika 14. Ekvipotencijale(Stanojević 2012)

Slika 15. Ekvipotencijale (razmak između čvorova: 0,125)