
5. naloga

Parcialne diferencialne enačbe: robni problemi in relaksacija

MIHA SRDINŠEK

I. POISEUILLOV KOEFICIENT ZA PRETOK VISKOZNE TEKOČINE (SOR)

Zdaj se bomo začeli ukvarjati s parcialnimi diferencialnimi enačbami, ki jih lahko v splošnem zapišemo kot

$$av_{xx} + bv_{xy} + cv_{yy} + dv_x + ev_y + fv = Q \quad (1)$$

, pri čemer je v splošnem $v = v(x, y)$ in indeksi predstavljajo parcialne odvode funkcije v . V večini primerov so le določeni koeficienti neničelni in v splošnem delimo parcialne enačbe na tri vrste glede na vrednost izraza

$$\mathcal{D} = b^2 - 4ac. \quad (2)$$

Tiste s pozitivno \mathcal{D} imenujemo hiperbolične, tiste z ničelno parabolične in tiste z negativno eliptične. Enačba, z reševanjem katere se bomo ukvarjali mi, spada pot eliptične enačbe. Numerično bomo poskušali izračunati hitrostni profil viskozne tekočine v cevi. Opazovali bomo presek na katerem je tlačni gradient konstanten in enak.

$$\rho \left(\frac{\partial \vec{u}}{\partial t} + \vec{u} \cdot \nabla \vec{u} \right) = -\nabla p + \mu \nabla^2 \vec{u} + \frac{1}{3} \mu \nabla (\nabla \cdot \vec{u}). \quad (3)$$

S fizikalno intuicijo to enačbo močno poenostavimo. Najprej rečemo da je tlačni gradient različen od nič le v z smeri, potem iz tega ocenimo, da se hitrost spreminja le v ravnini pravokotni na smer toka. Ker je primer stacionaren je tudi odvod po času enak nič. Gradient toka torej kaže pravokotno na smer toka in je tudi ta člen enak nič. Tako nam ostane močno poenostavljena enačba

$$\frac{\partial^2 u_z}{\partial x^2} + \frac{\partial^2 u_z}{\partial y^2} = \frac{1}{\mu} \frac{\partial p_z}{\partial z} = konst. \quad (4)$$

POiskali bi lahko tudi analitično rešitev, a zaplete se pri robnih pogojih. Za grde robne pogoje je analitično rešitev zelo težko najti, tam pride na vrsto zopet numerika. Tako kot bo v nadalje postalo povsem enostavno, najprej zapišemo mrežo, ki definira celotno področje, kjer iščemo

rešitev. Z mrežo določene natančnosti vsaki točki priredimo dva indeksa. Tako lahko enačbo prepišemo v diskretno obliko

$$-\frac{u_{j-1,k} - 2u_{j,k} + u_{j+1,k}}{\Delta x^2} - \frac{u_{j,k-1} - 2u_{j,k} + u_{j,k+1}}{\Delta y^2} = q_{j,k}. \quad (5)$$

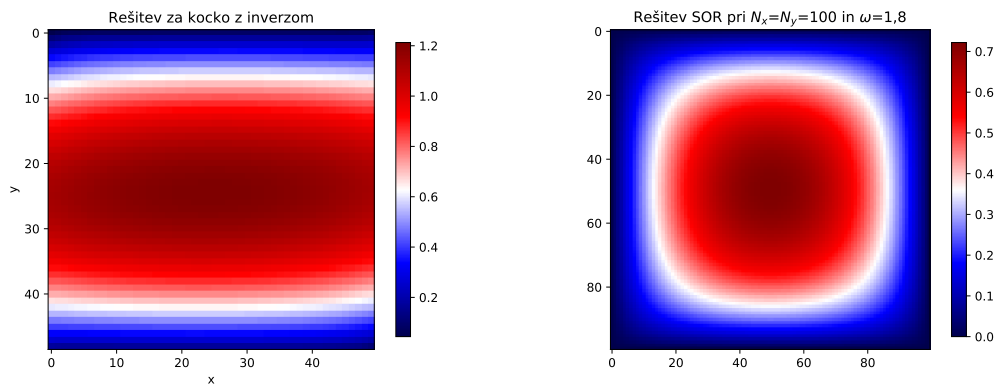
S tem pa smo na nek način naredili očitno, da lahko zapišemo operator nabla kot linearni operator oziroma matriko velikosti $N_x N_y \times N_x N_y$, ki jo bomo označevali z A . Hitrost lahko potem v tem smislu iz matrike problemujemo v vektor na katerem deluje ta preslikava A ,

$$\vec{u} = (u_{00}, u_{01}, \dots, u_{10}, u_{11}, \dots, u_{N_y N_x}), \quad (6)$$

tako da dobimo matrično enačbo

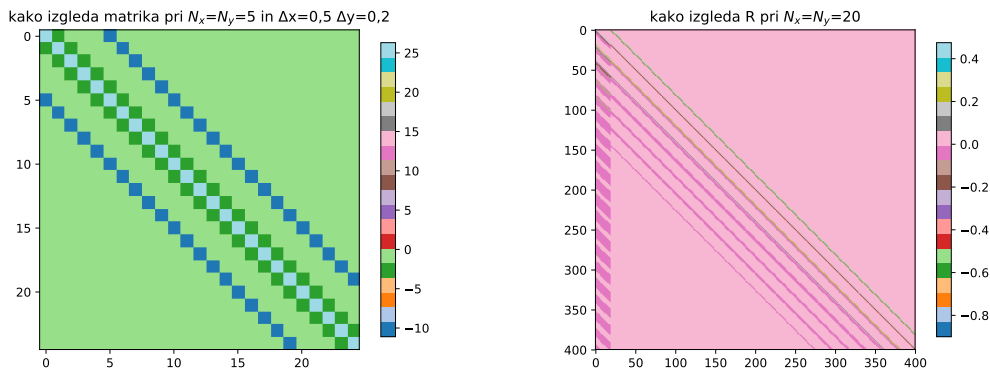
$$A\vec{u} = \vec{q}. \quad (7)$$

Enostavno računanje inverza seveda, razen z analitičnimi postopki reševanja, ne pride v poštev. Če izračunamo inverz in pogledamo rešitev, dobimo napačen rezultat prikazan na sliki 1. Tu smo izračunali hitrostni profil za cev s kvadratnim presekom. Pri tem sem mogoče zgolj kako narobe zapisal matriko A , a v vsakem primeru so bile prisotne komplikacije in algoritem je neverjetno počasen za večje N .



Slika 1: Slika levo prikazuje rešitev, če enačbo rešujemo z inverzom matrike A , tista desno pa pravo rešitev.

Zato si pogledajmo matriko, da bomo videli od kje pride ideja za naš novi algoritem. Matrika A je prikazana na sliki 2 levo.



Slika 2: Slika levo prikazuje matriko A , desna pa matriko R .

Vidimo, da je zelo lepa matrika in da bi lahko morda sistem reševanja močno poenostavili z upoštevanjem tega, da vedno upoštevamo zgolj nekaj komponent matrike. Poglejmo si torej poseben algoritem, ki smo ga spoznali na predavanjih. Uvedemo približek \vec{w} za \vec{u} in potem zmanjšujemo napako tega približka.

$$\vec{r} = A\vec{w} - \vec{q} \quad (8)$$

nam potem predstavlja residual. Rešitev izboljšujemo po postopku opisanem z matriko B

$$\vec{w}^{n+1} = \vec{w}^n + B\vec{r}^n \quad (9)$$

Uporabili bi lahko Jacobijev postopek pri katerem razdelimo matriko A na diagonalni, spodnje in zgornje trikoten del po postopku

$$A = L + D + U, \quad (10)$$

in rečemo $B = D^{-1}$, kar pomeni

$$\vec{w}^{n+1} = D^{-1}(\vec{q} - (L + U))\vec{w}^n, \quad (11)$$

otiroma potem po dolgem razpisvanju recept za nov algoritem

$$u_{j,k}^{n+1} = \frac{1}{d}(q_{j,k} + \frac{1}{(\Delta x)^2}(u_{j+1,k}^n + u_{j-1,k}^n) + \frac{1}{(\Delta y)^2}(u_{j,k+1}^n + u_{j,k-1}^n)). \quad (12)$$

Druga možna metoda je Gauss-Seidelova metoda, za katero lahko zapišemo matriko metode kot

$$B = (L + D)^{-1} \quad (13)$$

in potem algoritem kot

$$u_{j,k}^{n+1} = \frac{1}{d}(q_{j,k} + \frac{1}{(\Delta x)^2}(u_{j+1,k}^n + u_{j-1,k}^{n+1}) + \frac{1}{(\Delta y)^2}(u_{j,k+1}^n + u_{j,k-1}^{n+1})). \quad (14)$$

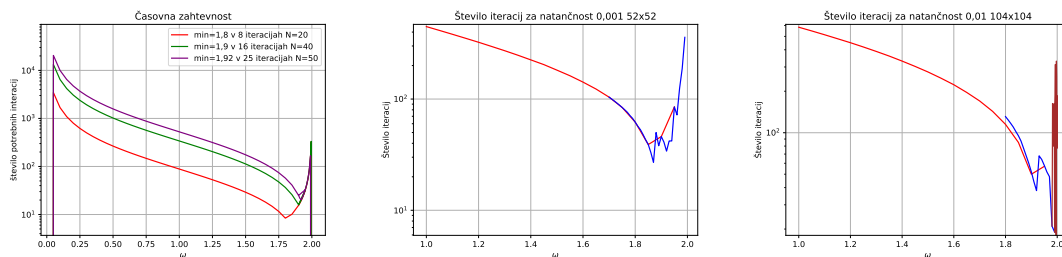
Ta dva postopka sta prepočasna, zato uvedemo nov postopek imenovan SOR s prehodno matriko

$$B = \omega(D + \omega L)^{-1}. \quad (15)$$

Tega postopka se lahko lotimo s pomočjo Jacobijevega nastavka ali pa Gauss-Seidlovega, tako rečemo $u_{j,k}^* = u_{j,k}^{n+1}$ in potem izvedemo nov algoritem odvisen od parametra $\omega \in (0, 2)$

$$u_{j,k}^{n+1} = u_{j,k}^n + \omega(u_{j,k}^* - u_{j,k}^n). \quad (16)$$

Zdaj si lahko pogledamo kakšen ω je najboljši - najbolj pohitri reševanje. To naredimo tak, da poiščemo po absolutni vrednosti največjo lastno vrednost matrike R $\rho(R)$, če



Slika 3: Slika levo prikazuje odvisnost števila iteracij potrebnih za konvergenco metode od parametra ω v teoriji in sliki desno v praksi.

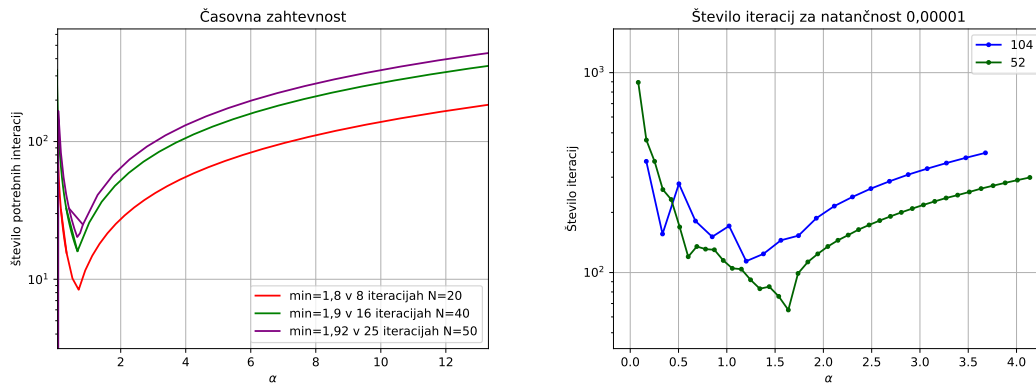
$$R = I - BA. \quad (17)$$

Potem velja, da je število iteracij potrebnih da metoda konvergira k pravi rešitvi $-1/\log(\rho(R))$. Tako matriko R vidimo na sliki 2 desno. Odvisnost števila iteracij potrebnih za konvergenco od parametra ω pa vidimo na slikah 3. Ko potem zaženemo metodo in opazujemo, kdaj postane napaka poljubno majhna, lako poustvarimo tak graf, kar vidimo na slikah desno.

Preverimo sedaj še trditev, da lahko dobro oceno za ω dobimo z enačbo

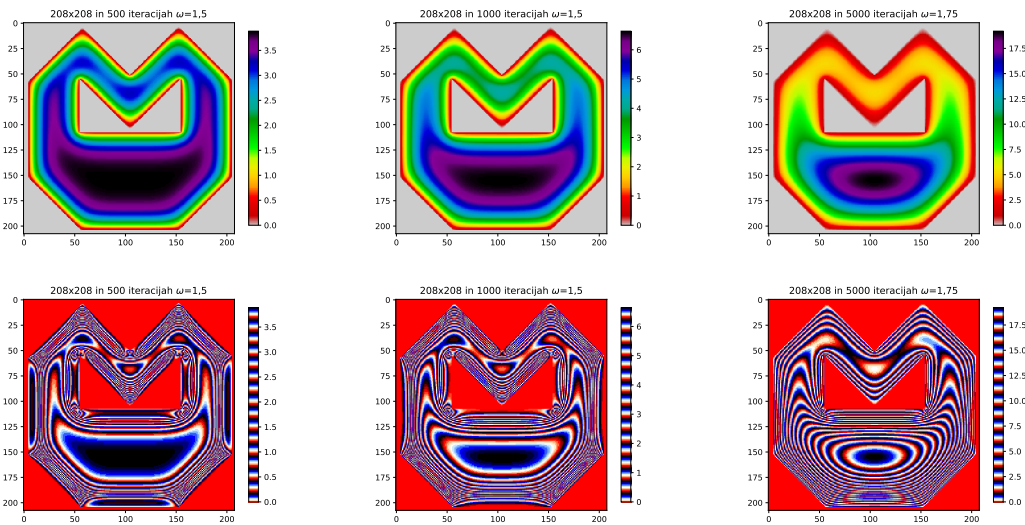
$$\frac{2}{1 + \alpha \frac{\pi}{J}}, \quad (18)$$

kjer je J število točk po enem robu. Če si izrišemo to, bi morali videti, da bomo pri različnih N dobili enak α . To vidimo na slikah 4.



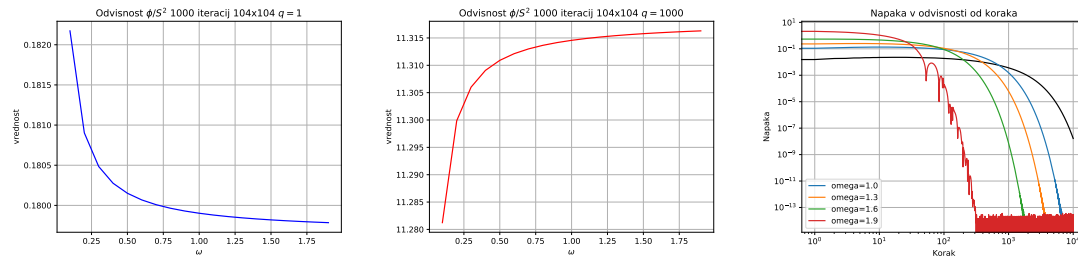
Slika 4: Slika levo prikazuje odvisnost števila iteracij potrebnih za konvergenco metode od parametra α v teoriji in slika desno v praksi. Na sliki levo so v legendi zapisane ω .

Vidimo, da se precej pokirrejo pri različnih N . Ko smo to preverili v praksi se je tudi izkazalo za dobro, ampak za lepši rezultat bi morali malo povečati natančnost. Slike ki smo jih dobili v praksi so slike, ki jih dobimo za primer prav posebnega profila, ki smo ga dobili za nalogo. Ima obliko mačke in je prikazan na slikah spodaj.



Na slikah zgoraj tudi vidimo rešitev z metodo *SOR*. Vidimo, kako s povečevanjem števila iteracij počasi konvergiramo k pravi rešitvi. Pri tem si lahko pogledamo kako se ne spreminjajo samo velikosti, ampak tudi struktura območja z enako hitrostjo.

Zdaj si lahko pogledamo še kako hitro se približujemo pravi rešitvi. Torej razlika zaporednih rešitev v odvisnosti od števila iteracij. V tem primeru dobimo lepo pričakovano konvergirajočo odvisnost. To lahko vidimo na dveh slikah levo 5, kjer se vidi, da se približujemo neki končni vrednosti. Kako se zmanjšuje napaka vidimo na sliki desno za različne *omega*. Kot smo že prej prikazali in kot pričakujemo že iz teorije, je najhitreje konvergirajoči postopek tisti z $\omega = 1,9$.



Slika 5: Sliki levo prikazujeta vrednost pretoka v odvisnosti od koraka pri različnih N , slika desno pa napako v odvisnosti od koraka. Črna krivulja predstavlja jakobijev algoritem.

S tem smo izčrpali prvo nalogo in se zato zaženemo v dolgo nalogo.

II. TOPLOTNA ENAČBA NA VALJU

V tej nalogi bomo samo še enkrat uporabili prejšnji postopek, pri kanček zahtevnejših robnih pogojih. Reševali bomo toplotno enačbo

$$\frac{\partial T}{\partial t} = D \nabla^2 T + \frac{q}{\rho c_p}. \quad (19)$$

Iskali bomo stacionarno stanje, tako, da je odvod temperature T po času enak nič in ne bomo imeli toplotnih virov. TO pa še ne pomeni, da ne moramo imeti virov na robu, le ne spadajo v enačbo. Če so viri toplote na robu, za to poskrbimo z robnimi pogoji. Enačba se torej prelevi na

$$\nabla^2 T = \frac{\partial^2 T}{\partial r^2} + \frac{1}{r} \frac{\partial T}{\partial r} + \frac{\partial^2 T}{\partial z^2} = 0, \quad (20)$$

ko uvedemo še cilindrični koordinatni sistem. Določimo še dimenzije valja, za katere veja, da je višina valja $h = 2R$ dvakratnik polmra valja. Notranjost, kjer ni virov bomo potem reševali zelo podobno kot prej na mreži

$$\frac{T_{j+1,k} - 2T_{j,k} + T_{j-1,k}}{(\Delta r)^2} + \frac{1}{j\Delta r} \frac{T_{j+1,k} - T_{j-1,k}}{2\Delta r} + \frac{T_{j,k+1} - 2T_{j,k} + T_{j,k-1}}{(\Delta z)^2} = 0. \quad (21)$$

Iz zgoraj zapisanih karakteristik valja lahko zapišemo $\Delta z = 2\Delta r$ in tako zahtevamo $N_r = N_z$. Zdaj lahko iz tega že vidimo, da je oblika diferenčne enačbe identična tisti prej, le koeficienti so nekoliko različni, a z lahkoto bomo zapisali matriko A . Koeficienti od prej zdaj postanejo

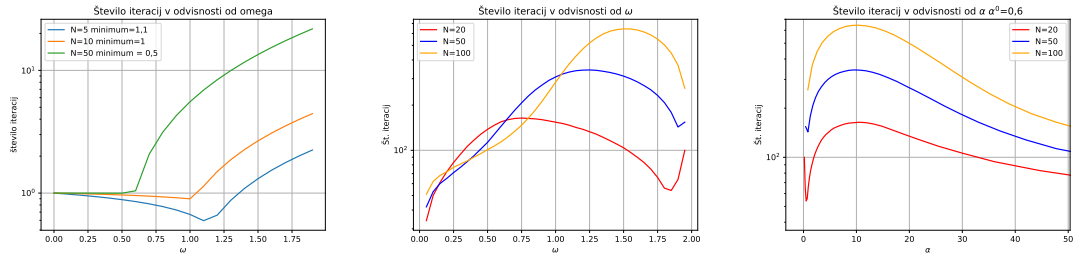
$$T_{j,k}^* = -\frac{2}{5} \left(\left(1 + \frac{1}{2j}\right) T_{j+1,k} + \left(1 - \frac{1}{2j}\right) T_{j-1,k} + \frac{1}{4} (T_{j,k+1} + T_{j,k-1}) \right). \quad (22)$$

Zaplete se le pri robnih pogojih, kjer se moramo tokrat malo bolj zamisliti o postopku reševanja. Pri prejšnji nalogi nas to sploh ni zanimalo, zgolj na robu mreže smo pustili pas ničel in ga ob vsaki iteraciji z *if* stavki ohranjali, tako da smo izračunavali vrednost algoritma zgolj na delih mreže, ki niso na robu. Na Če bi se dotaknili roba, bi imeli problem z robnimi pogoji, saj algoritem v vsaki točki (j, k) potrebuje vse štiri najbližje sosedne. S temi ničlami smo torej problem rešili. Tu pa ne moremo kar tako, saj imamo na treh robovih bolj dinamične robne pogoje.

Zanimamo se za valj, ki ga na spodnjem robu ogrevamo, zgornji rob pa držimo izoliran, zunaj plašč pa vzdržujemo pri konstantni temperaturi. Pri tem so enačbe zapisane v cilindričnih koordinatah, tako da moramo določiti tudi robni pogoj za $r = 0$, kjer se mora celoten profil prezrcaliti na drugo stran. Stacionarni robni pogoj na robu zagotovimo z lahkoto, samo ničle, ki smo jih imeli že prej, zdaj nadomestimo z neko konstanto. Ostale tri robne pogoje pa ustvarimo bolj premeteno. Oglejmo si najprej zgornji in spodnji robni pogoj. Zgoraj recimo naredimo tako, da na začetku na robni pas in na pas pred njim namestimo same ničle, na pasu pred njim nato poženemo algoritem. Po enem korgu (lahko pa tudi med krogom, kar sem naredil jaz), ko izračunamo vse vrednosti na pasu pred robom, vrstico prekopiramo na rob in tako ponavljamo v neskončnost. Isto naredimo spodaj. Tam želimo ustvariti gradient, ki ga bo algoritem začutil, zato damo na rob ničle in na sosednjo vrstico vrednost gradienta $\frac{1}{4N_r} j$ (upoštevamo, da je $\Delta z = 2\Delta r$). Vsak korak izračunamo vrednosti na sosednji vrstici in na rob prekopiramo izračunano vrednost minus gradient. Tako ustvarimo na spodnjem robu konstantni tok, na zgornjem robu pa tok uničimo. Podobno kot na zgornjem robu naredimo tudi pri $r = 0$. Tudi tam si pustimo rob prazen in v vsakem koraku prekopiramo vrednost na sosednji vrstici na rob, tako da zadostimo simetriji.

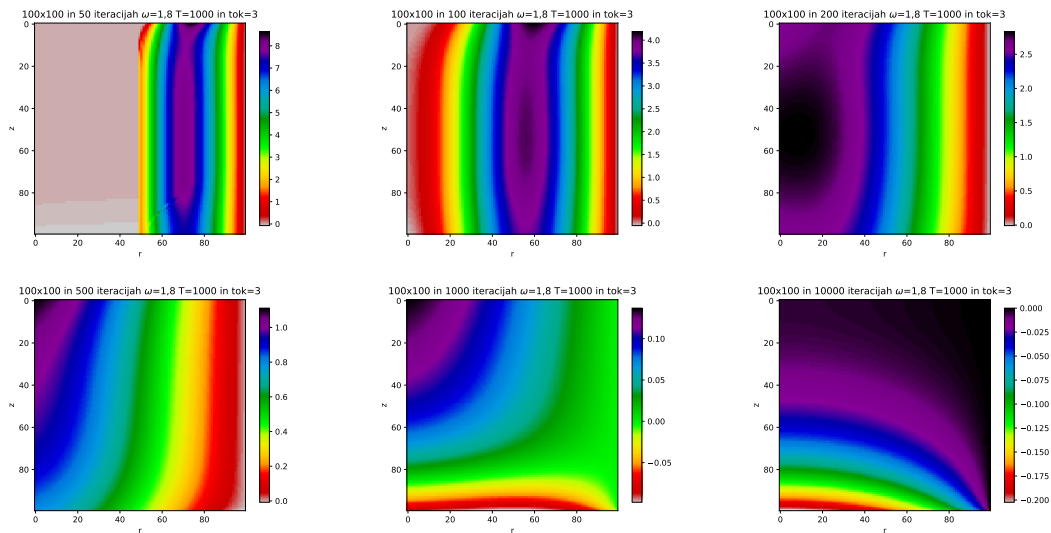
Vidimo, da so pri tem postopku zelo pomembni robni pogoji, in da morda prejšnji postopek za izračunavanje maksimalnega števila iteracij enostavno iz matrike A ne bo deloval, a vseeno

ga poskusimo, kar vidimo na slikah 6. S tem postopkom smo dobili povsem napačne rezultate v primerjavi z "eksperimentom", zato smo se v nadalje raje držali eksperimenta in računali pri tistih omegah, ki smo jih izračunali tam. Pri tem smo si pogledali še optimalni α , ki znaša približno 0,6.



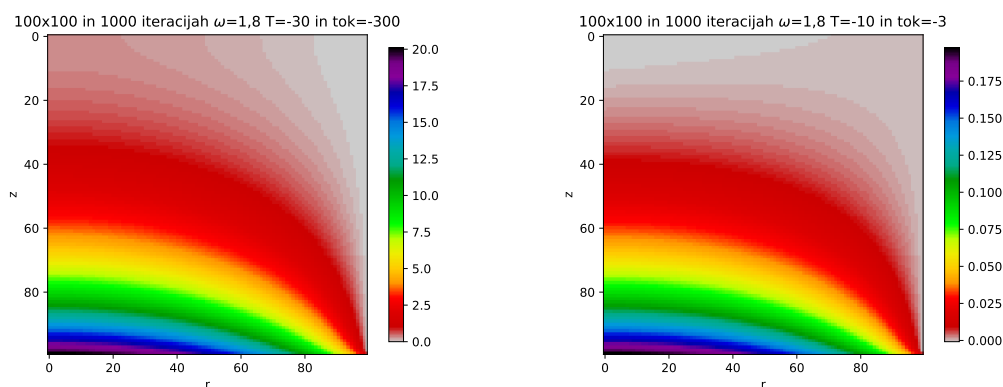
Slika 6: Slika levo prikazuje odvisnost števila iteracij potrebnih za konvergenco metode od parametra ω v teoriji in sliki desno v praksi.

Tu se zavlačevanje konča, pogledamo si lahko kako izgledajo rešitve temperaturnega profila na takem valju. To vidimo na slikah 7.



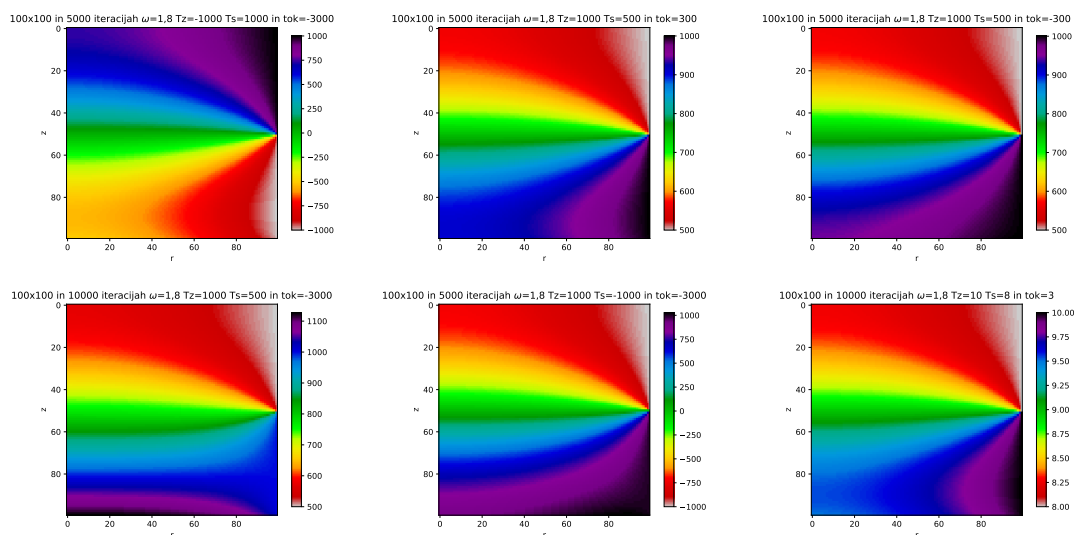
Slika 7: Slike prikazujejo kako se z iteracijami rešitev približuje končni, tako da vidimo kakšne so vmesne rešitve.

Še dve rešitvi pa vidimo na slikah 8. Vidimo torej zelo lepe slike, ki nam prikazujejo temperaturni profil na preseku valja. Očitno smo robne pogoje dobro izbrali, saj imamo recimo zelo lepo odvisnost pri $r = 0$. Pri tem je zelo pomembna opazka, da sem zaradi reglednosti rezultate izrisal tako, da sem konstantno temperaturo na robu T izenačil z 0. Tako vidimo zgledj odstopanje od vrednosti T in ne realno vrednost. Pravo vrednost dobimo tako, da vrednost na legendi prištejemo T .



Slika 8: Slike prikazujejo dve končni rešitvi.

V drugem delu naloge se na hitro vprašamo, kaj se zgodi, če polovico plašča držimo pri eni in drugo polovico pri drugi temperaturi. Za razne kombinacije emperatur in tokov sem izrisal take rešitve. Pri tem sem imel najprej težave, ker nisem dovolj strogo zapisal pogoj, zaradi česar je na robu s konstantno temperaturo ostala nedefinirana točka. To sem popravil in dobil naslednje rešitve na slikah 9.



Slika 9: Slike prikazujejo rešitve, če dele plašča držimo pri različnih temperaturah.

Vidimo, da je vse precej predvidljivo in zelo podobno.