REŠEVANJE PDE Z METODO GALERKINA

Žiga Kmecl

2. januar 2024

1 Uvod

Pri nalogi obravnavamo enakomeren laminarni tok viskozne in nestisljive tekočine po dolgi ravni cevi ob prisotnosti tlačnega gradienta p'. Problem opisuje Poissonova enačba

$$\nabla^2 v = -\frac{p'}{\eta}\,,$$

kjer je v vzdolžna komponenta hitrosti, odvisna samo od koordinat preseka cevi, η pa je viskoznost tekočine. Robni pogoji za problem so seveda nična vzdolžna komponenta hitrosti povsod ob robovih cevi.

Znana je enačba za pretok,

$$\Phi = C\,\frac{p' S^2}{8\pi\eta}\,,$$

koeficient C je odvisen le od dimenzij cevi in je za okroglo cev enak 1.

V naši nalogi pa nas zanima koeficient C za polkrožno cev, ob vpeljavi novih spremenljivk problem določajo enačbe

$$\Delta u(\xi,\phi) = -1\,, \qquad u(\xi=1,\phi) = u(\xi,0) = u(\xi,\phi=\pi) = 0\,,$$

$$C = 8\pi \iint \frac{u(\xi,\phi)\,\xi \mathrm{d}\xi \mathrm{d}\phi}{(\pi/2)^2}\,.$$

kjer je R polmer cevi, $\xi = r/R$ in $u = v\eta/(p'R^2)$

2 Rešitev

2.1 Uporaba Galerkinove metode

Pri tej nalogi bomo za reševanje parcialne diferencialne enačbe uporabili razvoj po lastnih funkcijah. V našem primeru v lastnih funkcijah nastopajo

Besselove funkcije, mi pa bomo uporabili približek s testnimi funkcijami Ψ_i in približek rešitve zapisali kot

$$\tilde{u}(\xi,\phi) = \sum_{i=1}^{N} a_i \Psi_i(\xi,\phi),$$

Ker je funkcija le približek, dobimo neko napako ϵ, ko nesemo funkcijo v diferencialno enačbo, velja

$$\nabla^2 \tilde{u}(\xi,\phi) + 1 = \varepsilon(\xi,\phi)\,.$$

S pravilno izbiro funkcij dosežemo, da je napaka ortogonalna na vse funkcije, kar je zahteva Galerkinove metode. Pogoj ortogonalnosti vodi do enačbe

$$\sum_{j=1}^{N} A_{ij} a_j = b_i\,, \qquad i = 1, 2, \ldots, N\,,$$

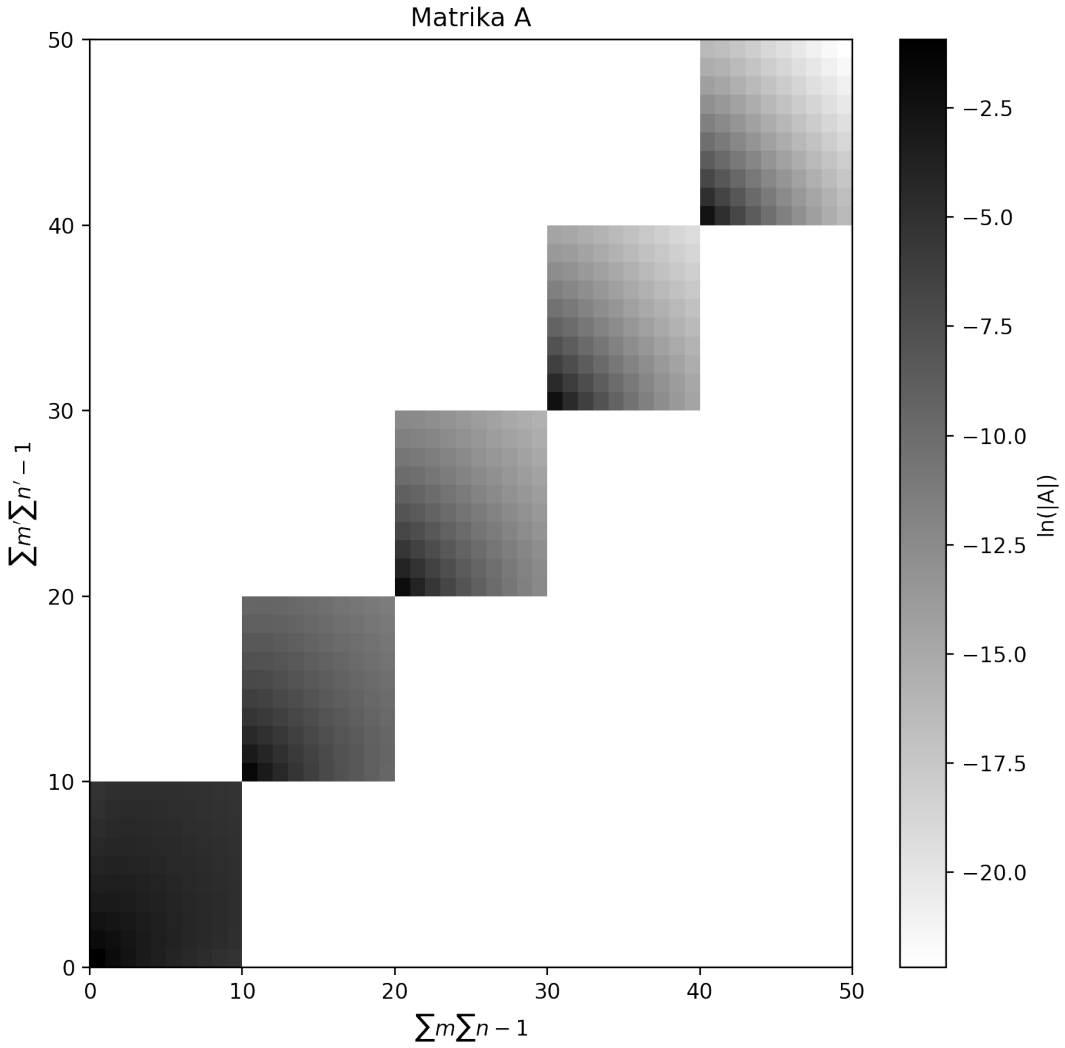
$$A_{ij} = (\nabla^2 \Psi_j, \Psi_i)\,, \qquad b_i = (-1, \Psi_i)\,,$$

a_j so uteži testnih funkcij v približku rešitve, enačba je seveda matrična enačba, koeficient za pretok pa je

$$C = -\frac{32}{\pi} \sum_{ij} b_i A^{-1}_{ij} b_j = -\frac{32}{\pi} \vec{b} \cdot \vec{a}\,.$$

Za kotni del testnih funkcij vzamemo $\sin((2m+1)\phi)$, za radialni del pa $\xi^{2m+1}(1-\xi)^n$, kjer indeks i seveda šteje tako po m kot po n.

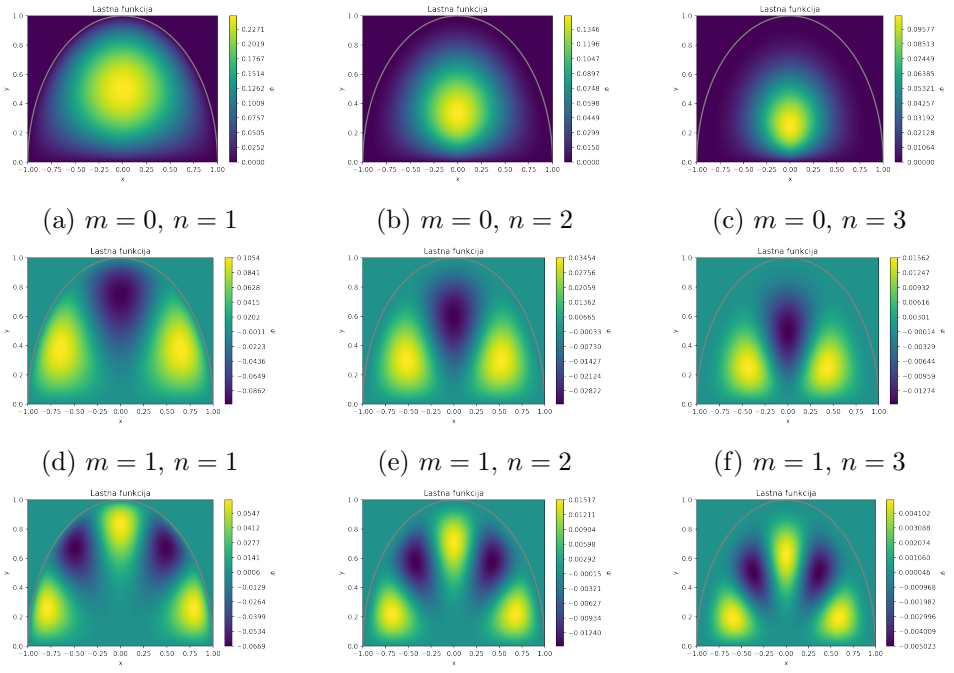
Matrika $\mathbf{A}$ bo zaradi ortogonalnosti po m-jih bločno diagonalna



Slika 1: Konstrukcija matrike $\mathbf{A}$ za $m = 4$, $n = 10$. Ker štejemo na način, da za vsak m najprej štejemo po vseh n, ortogonalnost funkcij po m zagotovi, da vzdolž diagonale matrike dobimo neničelne podmatrike, drugod pa ni ničesar.

2.2 Testne funkcije in hitrostni profil

Da bolje razumemo rešitev, si je smiselno najprej pogledati, s kakšnimi testnimi funkcijami sploh operiramo



(a) $m = 0$, $n = 1$ (b) $m = 0$, $n = 2$ (c) $m = 0$, $n = 3$

(d) $m = 1$, $n = 1$ (e) $m = 1$, $n = 2$ (f) $m = 1$, $n = 3$

(g) $m = 2$, $n = 1$ (h) $m = 2$, $n = 2$ (i) $m = 2$, $n = 3$

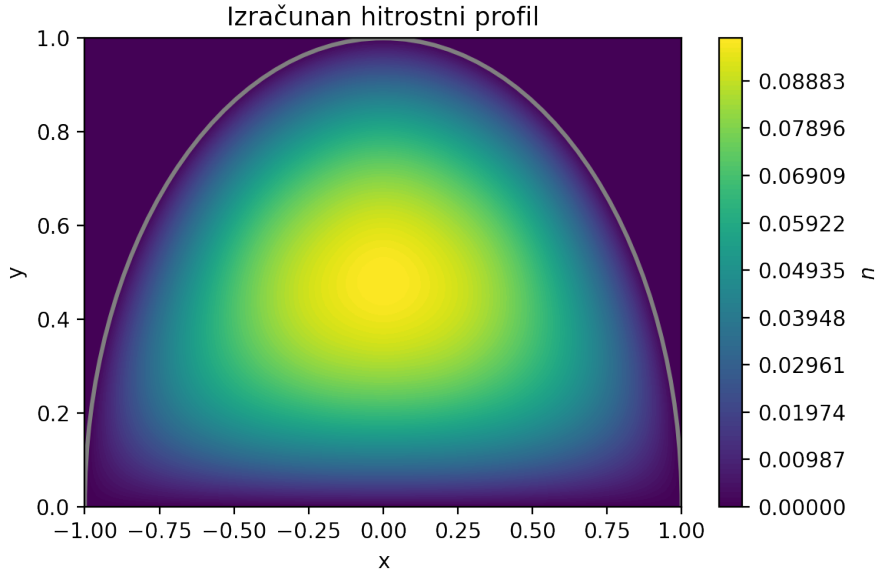
Slika 2: Uporabljene testne funkcije.

Sedaj imamo skonstruirano matriko **A** in vemo, kako izgledajo funkcije, iz katerih bo sestavljena rešitev. Iz slike matrike lahko razberemo, da so preferirane funkcije z manjšimi m-ji in n-ji in torej lahko slutimo, da bo profil v prvem približku podoben kar najbolj preprosti rešitvi z $m, n = 0, 1$.

Pa se prepričajmo. Sedaj moramo dobiti uteži, torej iščemo vektor $\vec{a}$. Iz enačbe

$$\mathbf{A}\vec{a} = \vec{b}$$

vidimo, da ga lahko najdemo z množenjem inverza matrike z vektorjem $\vec{b}$. Vendar pa zaradi lepih lastnosti matrike enačbo rešimo kar z vgrajeno metodo *solve_banded* v Pythonu.

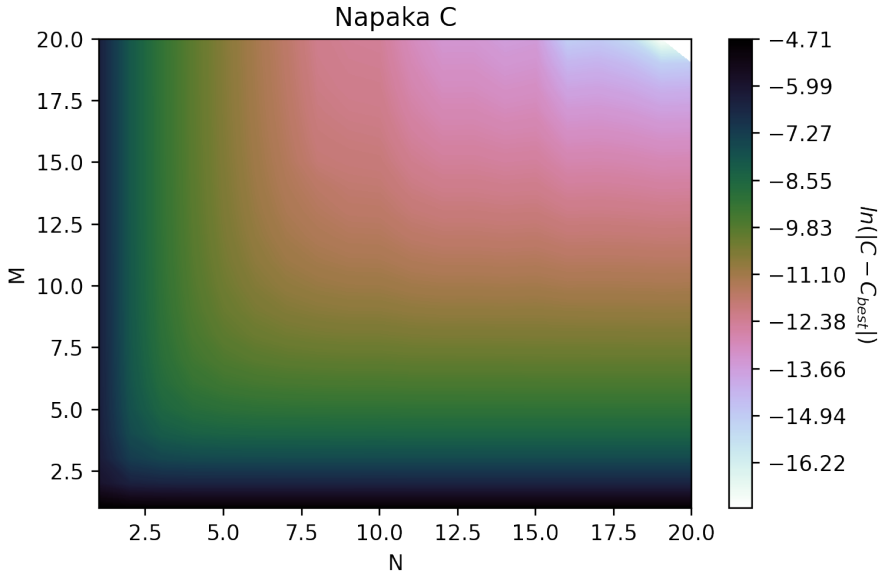


Slika 3: Izračunana rešitev za hitrostni profil (za $m_{max} = 4$, $n_{max} = 10$).

2.3 Natančnost metode

Sedaj lahko tudi izračunamo koeficient $C = 0,7577...$

Seveda nam je precej pomembno, kako dobro se Galerkinova metoda sploh obnese. Pomembno je ugotoviti še zlasti to, do katerih n in m sploh moramo iti za željeno natančnost, in ali sta oba parametra enako pomembna. Lahko se na primer izkaže, da bo rešitev veliko močneje vsebovala funkcije z velikimi n kot pa m in bi v nadaljnjih računih lahko višje m-je kar zanemarjali.



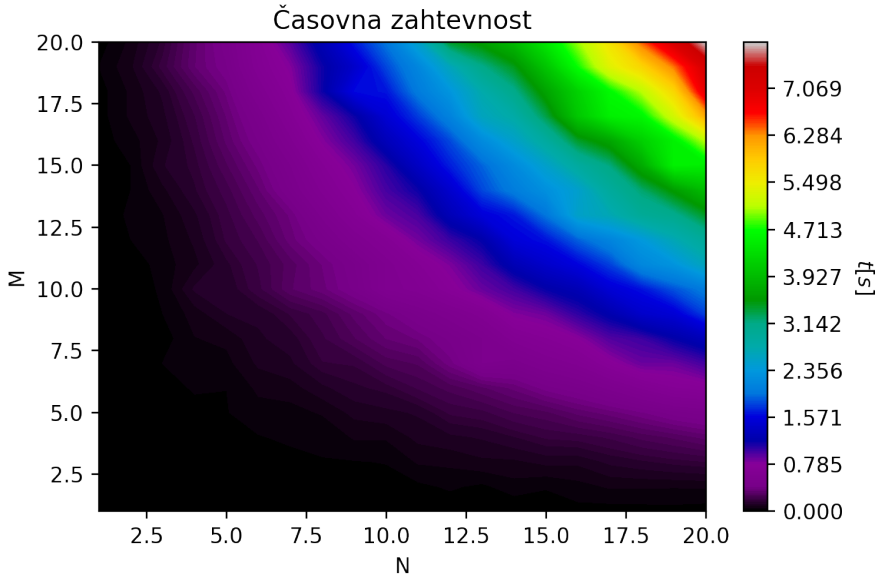
Slika 4: Izračunana rešitev za hitrostni profil (za $m_{max} = 4$, $n_{max} = 10$).

Za dani problem bi se seveda dalo poiskati tudi analitično rešitev, vendar je to zunaj obsega te naloge. Zato vse dobljene C primerjamo kar s C-jem, izračunanim z največ testnimi funkcijami. Pozoren bralec je zato morda že opazil. da je v zgornjem desnem kotu barvni gradient odrezan in je kot popolnoma bel; tam bi namreč C primerjali s samim sabo.

Kakor koli, glavna ugotovitev tu je, da je natančnost rešitve odvisna tako od števila funkcij po naraščajočem n kot tudi po m, najbolj smiselno pa se je premikati približno po diagonali z enakomernim povečevanjem po obeh parametrih.

2.4 Časovna zahtevnost

Za konec si poglejmo še časovno zahtevnost računanja



Slika 5: Čas, potreben za izračun, v odvisnosti od števila testnih funkcij po m in n.

Tu ni kaj dosti za komentirati, razen morda tega, da je bila naloga v splošnem bolj časovno zahtevna od večine prejšnjih domačih nalog, predvsem zato, ker tu obravnavam dva prosta parametra naenkrat, koda ima več zank in je na splošno več računanja.

3 Zaključek

Naloga konceptualno ni bila pretežka, v veliko pomoč je bilo pri reševanju je bilo, da je bilo precej teoretičnih izračunov že narejenih na powerpoint prosojnicah; brez tega bi se mi najbrž zataknilo že pri samem začetku oziroma bi neizbežno naredil napake pri izpeljavah in prišel do napačnih rezultatov.

Zanimivo bi bilo tudi preveriti, kako uporaben je tak pristop napram drugim že obravnavanim metodam za PDE. Glede na počasnost izvajanja kode v tej nalogi slutim, da bi se kaj drugega obneslo precej bolje. Vendar pa je bila ta naloga reševana v prazničnem času, v luči česar sem se odločil, da je mojo pozornost bolj zdravo usmeriti v druge reči ...