UNIVERZA V LJUBLJANI | FMF Fakulteta za matematiko in fiziko

Matematično-fizikalni praktikum

Enajsta naloga: *Galerkinova metoda*

Simon Bukovšek, 28211067

Mainz, 11. januar 2024

Profesor: prof. dr. Borut Paul Kerševan

Naloga: Galerkinova metoda

Izračunaj koeficient C. V ta namen moraš dobiti matriko A in vektor b; preuči, kako je natančnost rezultata (vsote za koeficient C) odvisna od števila členov v indeksih m in n. Zaradi ortogonalnosti po m lahko oba učinka preučuješ neodvisno.

1 Uvod

Pri opisu enakomernega laminarnega toka viskozne in nestisljive tekočine po dolgi ravni cevi pod vplivom stalnega tlačnega gradienta p' se Navier-Stokesova enačba poenostavi v Poissonovo enačbo

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

kjer je v vzdolžna komponenta hitrosti, odvisna samo od koordinat preseka cevi, η pa je viskoznost tekočine. Enačbo rešujemo v notranjosti preseka cevi, medtem ko je ob stenah hitrost tekočina enaka nič. Za pretok velja Poiseuillov zakon

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

kjer je koeficient C odvisen samo od oblike preseka cevi ($C = 1$ za okroglo cev). Določili bomo koeficient za polkrožno cev z radijem R. V novih spremenljivkah $\xi = r/R$ in $u = v\eta/(p'R^2)$ se problem glasi

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

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

Če poznamo lastne funkcije diferencialnega operatorja za določeno geometrijo[1] se reševanje parcialnih diferencialnih enačb včasih lahko prevede na razvoj po lastnih funkcijah. Da bi se izognili računanju lastnih (za ta primer Besselovih) funkcij in njihovih ničel, ki jih potrebujemo v razvoju, lahko zapišemo aproksimativno rešitev kot linearno kombinacijo nekih poskusnih (*trial*) funkcij

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

za katere ni nujno, da so ortogonalne, pač pa naj zadoščajo robnim pogojem, tako da jim bo avtomatično zadoščala tudi vsota (1). Ta pristop nam pride prav v kompleksnejših geometrijah, ko je uporabnost lastnih funkcij izključena in potrebujemo robustnejši pristop. Približna funkcija $\tilde{u}$ seveda ne zadosti Poissonovi enačbi: preostane majhna napaka ε

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

Pri metodi Galerkina zahtevamo, da je napaka ortogonalna na vse poskusne funkcije Ψ_i,

$$(\varepsilon, \Psi_i) = 0\,, \qquad i = 1, 2, \ldots, N\,.$$

V splošnem bi lahko zahtevali tudi ortogonalnost ε na nek drug sistem utežnih (*weight*) oziroma testnih (*test*) funkcij Ψ_i. Metoda Galerkina je poseben primer takih metod (*Methods of Weighted Residuals*) z izbiro $\Psi_i = \Psi_i$. Omenjena izbira vodi do sistema enačb za koeficiente a_i

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

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

tako da je koeficient za pretok enak

$$C = -\frac{32}{\pi} \sum_{ij} b_i A_{ij}^{-1} b_j\,.$$

Za kotni del poskusne funkcije obdržimo eksaktne funkcije $\sin((2m+1)\phi)$, Besselove funkcije za radialni del pa nadomestimo s preprostejšimi funkcijami $\xi^{2m+1}(1-\xi)^n$. Pozor: indeks i pomeni seveda dvojni indeks (šteje obenem m in n)[2]. Zaradi ortogonalnosti po m razpade matrika A v bloke, obrneš pa jo lahko s kako pripravljeno rutino, npr. s spodnjim in zgornjim trikotnim razcepom `ludcmp` in `lubksb` iz NRC.

[1] Spomni se na primer na vodikov atom v sferični geometriji, kjer smo imeli $\widehat{L}^2 Y_{lm}(\theta,\phi) = \hbar^2 l(l+1) Y_{lm}(\theta,\phi)$ in $\widehat{L}_z Y_{lm}(\theta,\phi) = m\hbar Y_{lm}(\theta,\phi)$.

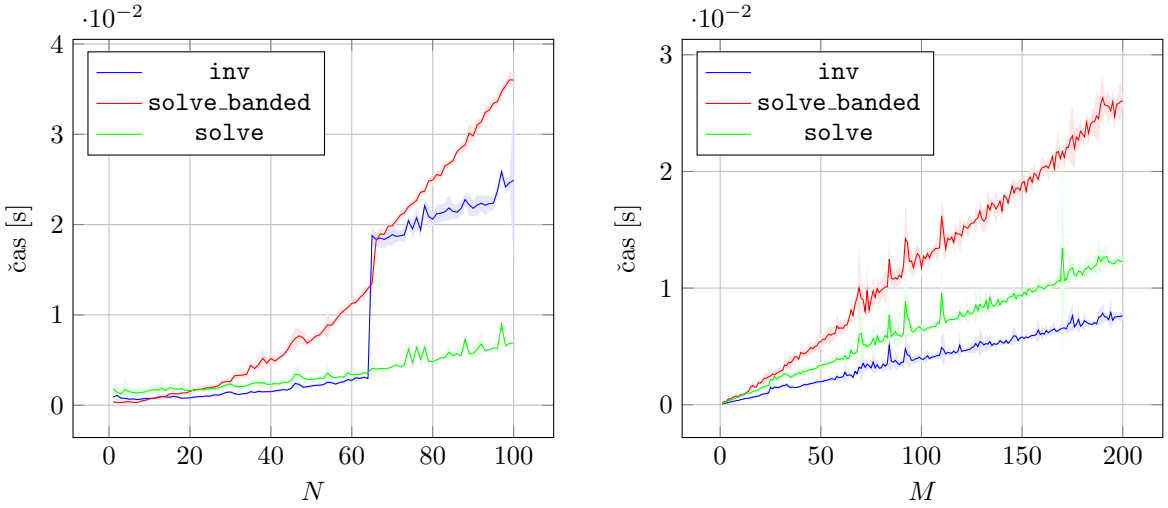
[2] Glej tudi prilogo na spletni učilnici.

2 Izbira metode

Tokratna naloga je v osnovi zelo preprosta in kratka. Izračunati moramo število $C = \mathbf{b}^T \mathsf{A}^{-1}\mathbf{b}$, pri čemer sta matrika A in vektor $\mathbf{b}$ eksplicitno podana. Vektor $\mathbf{b}$ je velikosti $M \times N$, matrika A pa je bločno diagonalna, sestavljena iz M simetričnih blokov velikosti $N \times N$. Izračuna se lahko lotimo malo bolj ali malo manj premišljeno. Tukaj je naštetih nekaj metod po naraščajoči učinkovitosti.

- Lahko preprosto sestavimo matriko A, jo obrnemo s pomočjo funkcije `np.linalg.inv` ter matrično množimo z vektorjema $\mathbf{b}$ z leve in desne. Ta metoda nam vzame $\mathcal{O}(N^3M^3)$ časa in $\mathcal{M}^{\in}\mathcal{N}^{\in}$ prostora.
- Namesto da bi obračali matriko A in jo množili z vektorjem $\mathbf{b}$, lahko uporabimo metodo `scipy.linalg.solve` za izračun $\mathsf{A}^{-1}\mathbf{b}$. Ta metoda direktno reši sistem enačb $\mathsf{A}\mathbf{x} = \mathbf{b}$, kar je v našem primeru ekvivalentno iskanju inverza matrike A. Ta metoda še vedno vzame $\mathcal{O}(N^3M^3)$ časa in $\mathcal{O}(M^2N^2)$ prostora, vendar je številski faktor v obeh primerih manjši (shraniti rabimo samo eno matriko velikosti $MN \times MN$).
- Lahko opazimo, da je matrika A pasovna, saj je v primeru $M \gg 1$ skoraj povsod enaka nič. V ta namen lahko izkoristimo algoritem `scipy.linalg.solve_banded`, ki hitro rešuje sisteme enačb z pasovnimi matrikami. Ta metoda nam vzame $\mathcal{O}(N^3M)$ časa in $\mathcal{O}(N^2M^2)$ prostora (metoda še vedno zahteva, da ji podamo celo matriko).
- Bločno diagonalna oblika matrike nam ponuja še hitrejšo možnost za izračun. Matriko A lahko razcepimo na M blokov velikosti $N \times N$ in obrnemo vsakega posebej (spet s funkcijo `scipy.linalg.inv`). To nam vzame $\mathcal{O}(N^3M)$ časa, vendar pa ob previdni uporabi zahteva le $\mathcal{O}(N^2)$ prostora (v pomnilniku ne rabimo imeti hkrati več kot enega bloka).
- Iskanje inverza ponovno lahko nadomestimo s funkcijo `scipy.linalg.solve`, ki direktno reši sistem enačb $\mathsf{A}\mathbf{x} = \mathbf{b}$. Ta metoda nam vzame $\mathcal{O}(N^3M)$ časa in $\mathcal{O}(N^2)$ prostora.

Prvi dve metodi sta se mi zdeli preveč potratni (časovno in prostorsko), zato sem primerjal le zadnje tri možnosti. Pri konstantnem M sem spreminjal N ter pri konstantnem N sem spreminjal N in opazoval hitrost reševanja celotnega problema za zadnje tri naštete metode. Rezultati so prikazani na sliki 1a in 1b. Zanimivo je metoda `inv` pri $N = 30$ hitrejša od `solve` za vse preizkušene M, pri konstantnem M pa to velja samo za $N < 65$. Takrat metoda `inv` naglo zraste v času izvajana. V vsakem primeru lahko odpišemo metodo `solve_banded`, saj je časovno najpočasnejša, po porabi RAM-a pa je prav katastrofalna. Moj pomnilnik lahko drži matriko največ $10\,000 \times 10\,000$, torej sem s to metodo omejen na $M \times N < 10\,000$. Pri ostalih dveh metodah sem omejen samo na $N < 10\,000$, pri čemer me seveda časovne omejitve dohitijo bistveno prej.

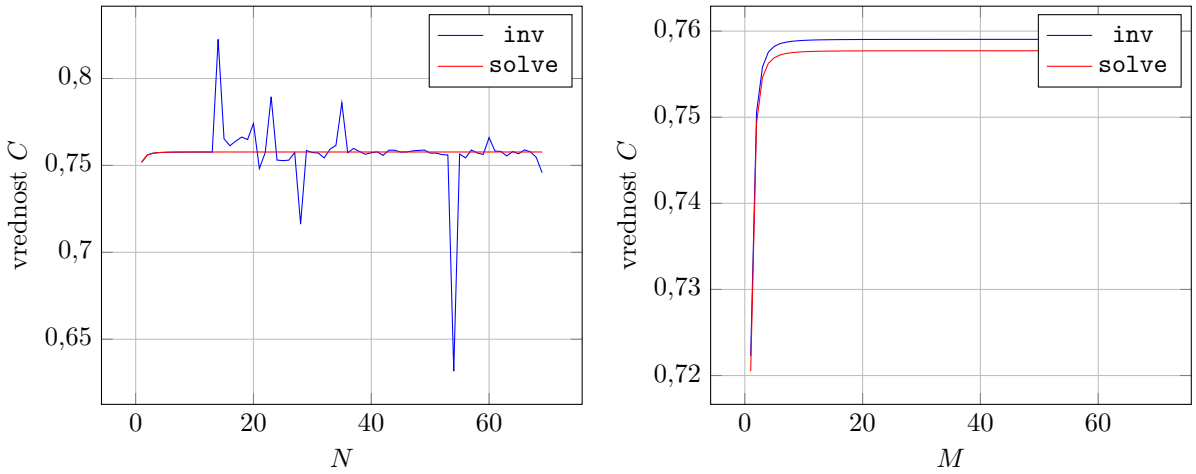


(a) Čas izračuna pri $M = 30$. **(b)** Čas izračuna pri $N = 30$.

Slika 1: Čas izračuna koeficienta pri različnih velikostih matrike in različnih metodah. V svetlem je prikazan standardni odklon na vzorcu desetih ponovitev.

3 Natančnost

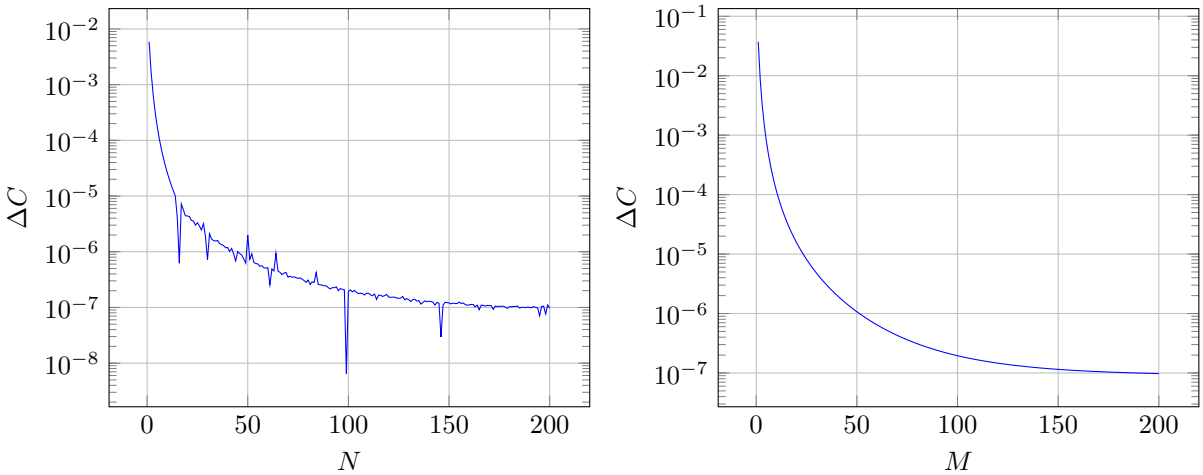
Za izračun števila C sem uporabil vse tri metode iz grafov 1. Hitro sem ugotovil, da `solve` in `solve_banded` vrneta identične rezultate, `inv` pa ne. Zaradi tega opažanja sem se odločil, da primerjam metodi `inv` in `solve` še računsko. Vzel sem $M = 100$ ter spreminjal N med 1 in 70 (Slika 2a). Obe metodi sta se zelo hitro približala neki konstantni vrednosti. Metoda `solve` je pri tej vrednosti ostala pri poljubno velikem N, medtem ko je metoda `inv` že pri $N = 15$ začela naključno odstopati. Pri konstantnem $N = 100$ in spreminjajočem M pa sta obe metodi lepo konvergirali, vendar ne proti isti vrednosti (Slika 2b). Ena od metod očitno kaže narobe. Preveril sem z množenjem matrike A z vektorjem $\mathbf{x}$ (rešitev sistema enačb $\mathsf{A}\mathbf{x} = \mathbf{b}$). Metoda `solve` je dala rezultate s strojno natančnostjo (in zanesljivostjo), metoda `inv` pa je dala mnogo manj točne vrednosti. Sumim, da so razlog zelo majhne vrednosti v elementov v matrikah in večja količina operacij pri `inv`. Pri $M = 190$ in $N = 190$ še `solve` ni uspel rešiti sistema, saj so bili elementi matrike tako majhni, da je izračunal ničelno determinanto.



(a) Vrednost C pri $M = 100$.

(b) Vrednost C pri $N = 100$.

Slika 2: Vrednost koeficienta C pri različnih velikostih matrike in različnih metodah. Čeprav je po hitrosti izračuna včasih hitrejša metoda `inv`, pa je mnogo manj konsistentna.



(a) Odstopanje C pri $M = 100$.

(b) Odstopanje C pri $N = 100$.

Slika 3: Absolutno odstopanje koeficienta C od prave vrednosti pri različnih velikostih matrike. Prava vrednost je bila računana pri $N = 180$ in $M = 180$.

Za izračun točnega rezultata sem vzel $N = 180$ in $M = 180$ in dobil

$$C = 0{,}757\,722\,068.$$

Napako sem ocenil, tako da sem dodal še en blok (torej povečal M za ena) in preveril, koliko ta blok doprinese. Prispevek dodatnega bloka je bil reda 10^{-11}, zato lahko napako ocenim na 10^{-10}. Lahko bi se zgodilo, da `solve` ne reši sistemov dovolj natančno (program je javljal opozorila), morda se je večja napaka akumulirala preko mnogo izvedenih seštevanj, vendar do devete decimalke sem dokaj prepričan v natančnost rezultata.

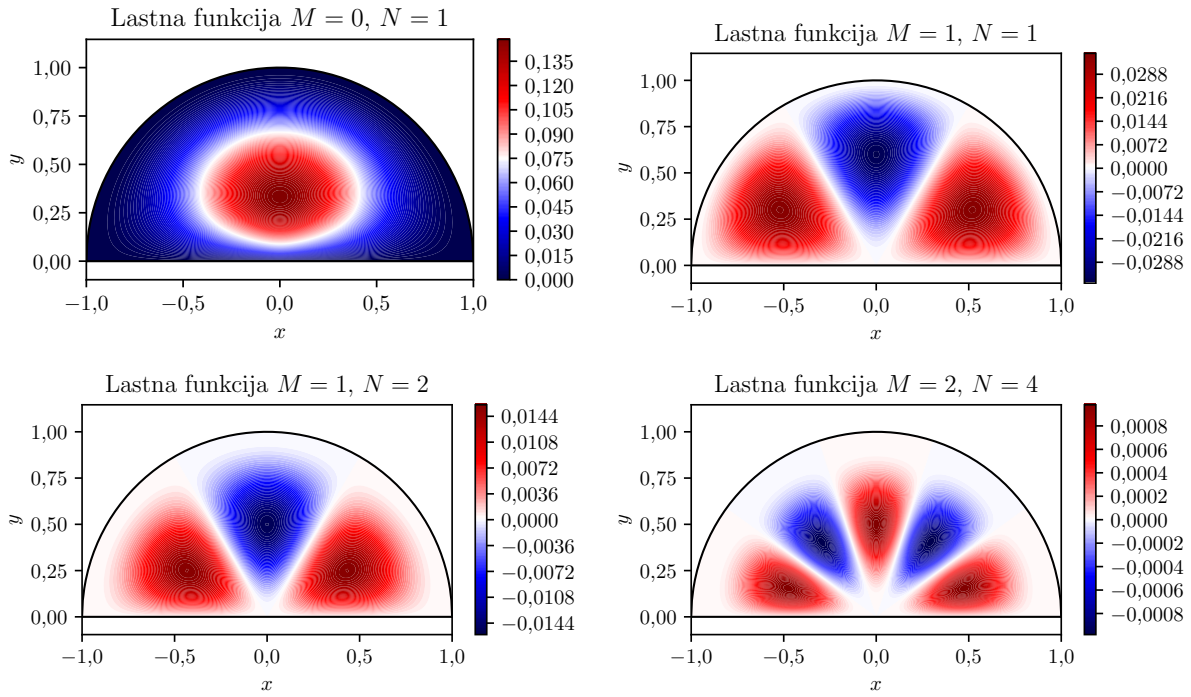
Ostane še vprašanje, kako se napaka zmanjšuje z večanjem velikosti M in N. Odgovor je prikazan na Grafih 3a in 3b. V obeh primerih je bila fiksna dimenzija enaka 100. V obeh primerih se napaka najprej zmanjšuje hitreje kot eksponentno, pri večjih vrednostih dimenzije pa se ustali pri 10^{-7}.

4 Pretok skozi polkrožno cev

Pretok skozi cev lahko približno ocenimo iz naših poskusnih funkcij:

$$u(\xi, t) = \sum_{m,n} a_{mn} \Psi_{mn}(\xi, \phi).$$

Poglejmo si najprej nekaj osnovnih (poenostavljanih) poskusnih funkcij (Slika 4).

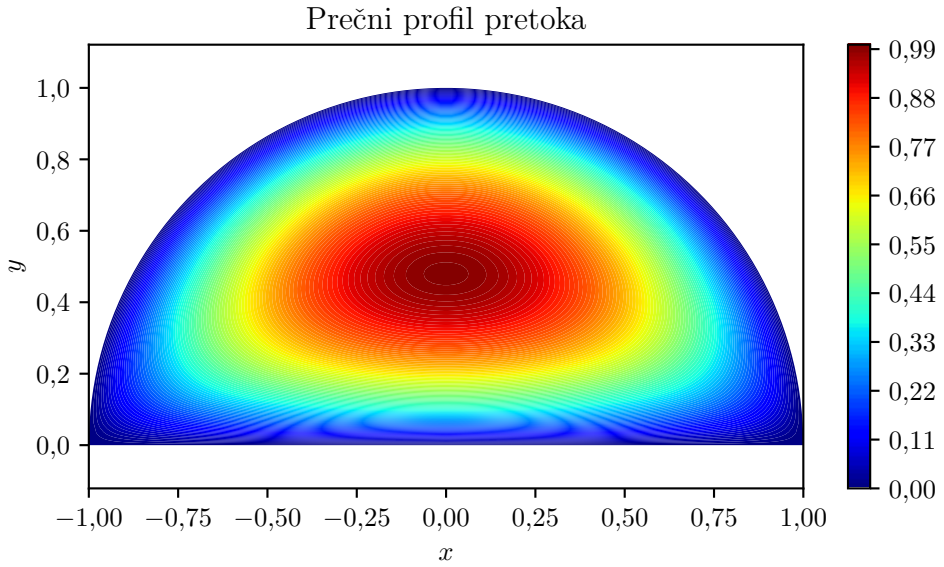


Slika 4: Prikaz nekaterih poskusnih funkcij, iz katerih sestavimo rešitev.

Če za koeficiente vzamemo ustrezne vrednosti, ki jih dobimo iz vektorja $\mathbf{a}$, lahko izračunamo pretok skozi cev. Pri $N = M = 3$ je na pogled rešitev že praktično enaka kot pri kateri koli višji dimenziji. Pri $N = 100$ in $M = 100$ sem dobil pretok, prikazan na Sliki 5. Opazimo, da je pretok lepo normiran na $\max u = 1$.

5 Zaključek

Tokrat nisem imel časa za dodatno raziskovanje in sem dodatno nalogo preskočil. Veliko časa sem namenil razmišljanju o različnih metodah in časovnih zahtevnostih. Po svoje sem kar vesel, da smo končali z diferencialnimi enačbami.



Slika 5: Profil pretoka skozi polkrožno cev.