

# Direktno reševanje Poissonove enačbe

Miha Čančula

17. maj 2012

## 1 Kvadratna opna

Najprej sem na primeru kvadratne, enakomerno obtežene opne preverjal učinkovitost treh metod:

**SOR** Pospešeno relaksacijo smo uporabljali že pri eni izmed prejšnjih nalog, zato sem lahko uporabil kar isti algoritem.

**1D FFT** Opno sem preslikal v frekvenčni prostor le po eni dimenziji, v drugi dimenziji pa sem nastavil in rešil tridiagonalni matrični sistem v vsaki vrstici.

**2D FFT** Najenostavnejša med uporabljenimi metodami pa je bila dvodimenzionalna Fourierova transformacija. Tu sem uporabil FFT v obeh dimenzijah, koeficiente delil s primernim faktorjem in tako rešil enačbo, nato pa spet uporabil dvodimenzionalno inverzno transformacijo.

Za čimbolj nepristransko primerjavo sem vse tri algoritme implementiral v jeziku Octave, za računsko zahtevne dele (FFT, matrične operacije) pa sem uporabil

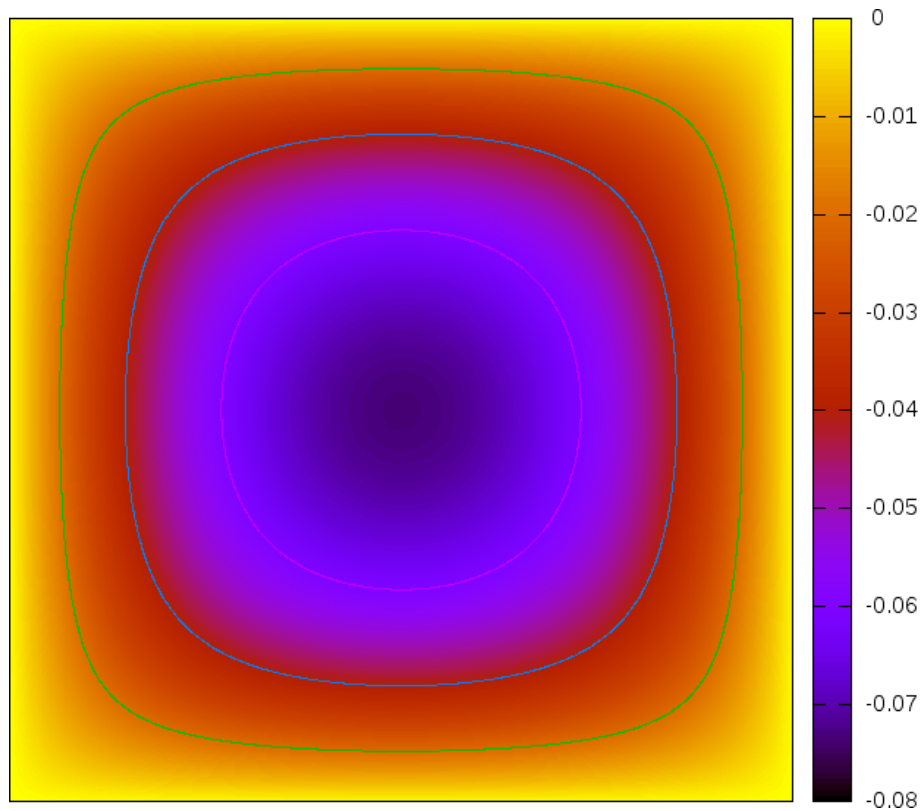
Relaksacijo sem implementiral v matrični obliki. Poves opne v vsaki točke na mreži sem predstavljal z vektorjem  $x$  z  $n \times n$  komponentami, popravek v vsak

$$\mathbf{p}_i = \mathbf{A}\mathbf{x}_i + \mathbf{g} \quad (1)$$

$$\mathbf{x}_{i+1} = \mathbf{x}_i + \frac{1}{4}\omega\mathbf{p}_i \quad (2)$$

kjer je  $\mathbf{A}$  matrika diskretiziranega operatorja  $\nabla^2$ ,  $\mathbf{g}$  vektor desne strani Poissonove enačbe,  $\mathbf{p}_i$  pa popravek v  $i$ -tem koraku. Relaksacijo sem ustavil, ko je velikost popravka  $\|\mathbf{p}_i\|$  padla pod določeno mejo.

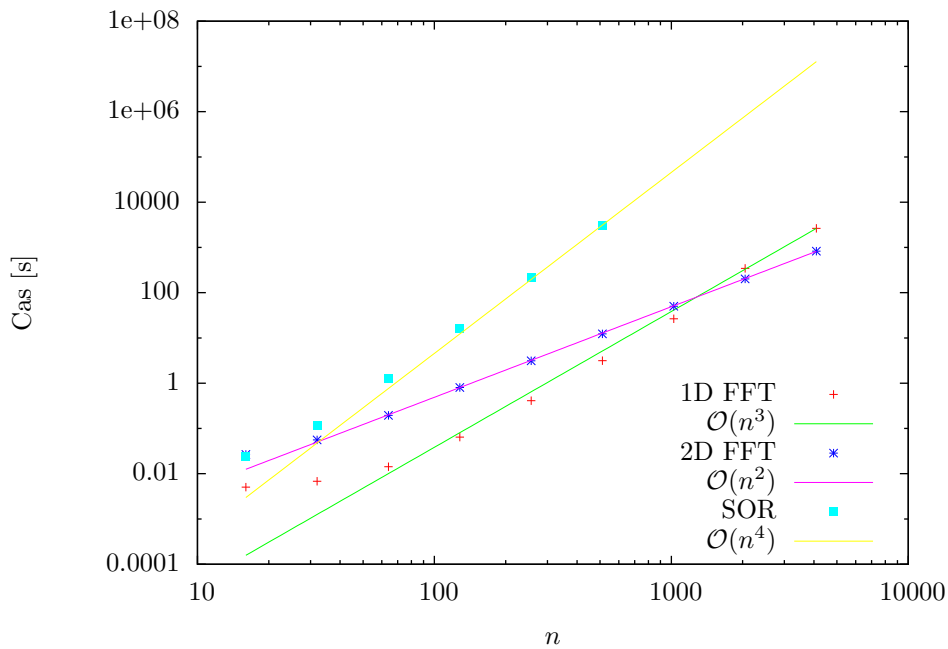
Ker imamo v nalogi robne pogoje 1. vrste, v Fourierovem razvoju nastopajo le sinusi. Zato sem namesto FFT vedno uporabil hitro sinusno transformacijo, ki jo ponuja paket `octave-control`.



Slika 1: Oblika enakomerno obtežene opne

## 1.1 Hitrost

Najprej sem primerjal hitrost različnih algoritmov. FFT in reševanje 1D diferencialne enačbe sta direktna postopka, kjer je natančnost odvisna le od diskretizacije oz. od števila točk. Pri relaksaciji pa sem moral določiti še pogoj, kdaj ustaviti računanje. Izbral sem enakega kot pri nalogi 205, torej sem račun ustavil, ko je vsota kvadratov popravkov v enem koraku padla pod fiksno mejo  $10^{-8}$ .



Slika 2: Časovna zahtevnost uporabljenih algoritmov

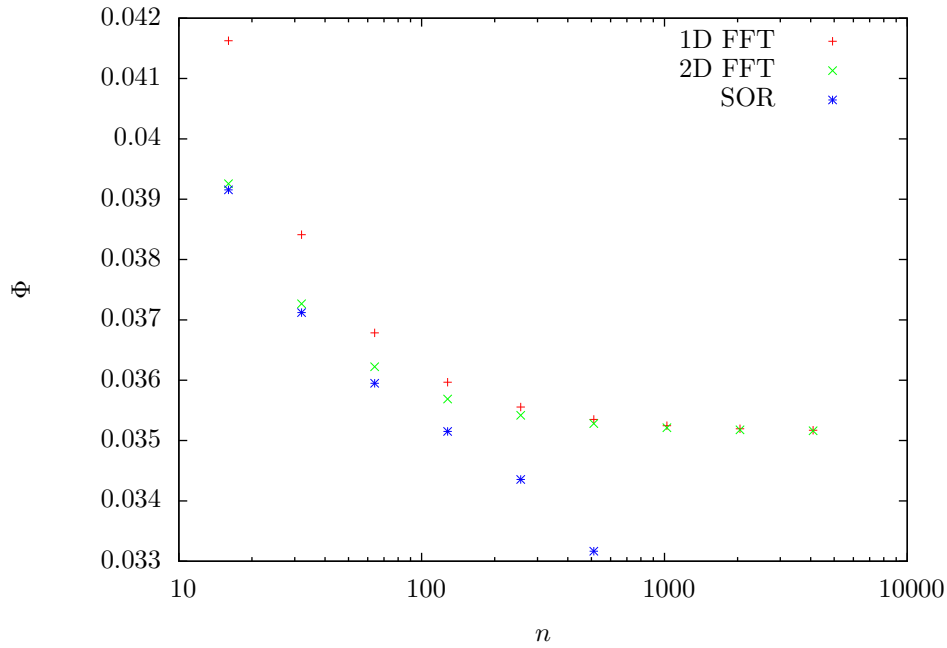
Pri diskretizacijah z majhnim številom točk (do  $1000 \times 1000$ ) je najhitrejša metoda s Fourierovo transformacijo v eni smeri in reševanju sistema v drugi. Z večanjem števila točk pa časovna zahtevnost hitro narašča in postane najugodnejša dvodimenzionalna FFT. Dvodimenzionalna metoda ima tudi najbolj jasno časovno odvisnost  $\mathcal{O}(n^2)$ , torej kar linearno s številom točk. Pri ostalih dveh ne vidimo tako lepe potenčne odvisnosti.

Relaksacija pri zgoraj opisanem pogoju za ustavitev je najbolj računsko zahtevna metoda. Njeno časovno zahtevnost bi lahko spremenil, če bi mejo za ustavitev spreminjal s številom točk. S tem pa bi se spremenila tudi natančnost metode, ki sem jo opazoval na naslednjem grafu.

## 1.2 Natančnost

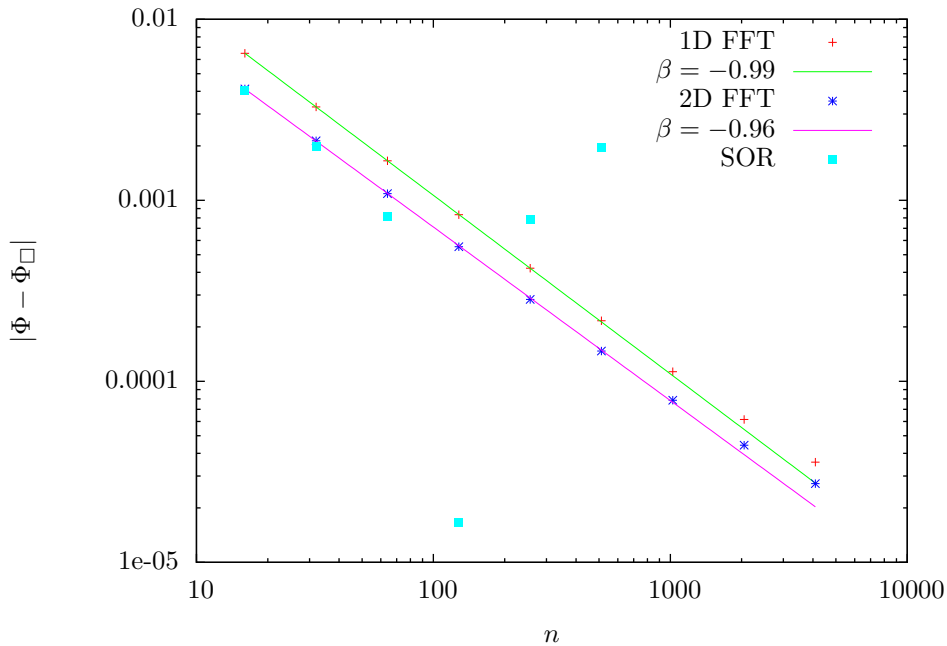
Časovna odvisnost od števila točk nam nič ne pove, če ne vemo kakšno natančnost lahko pričakujemo. Zato sem opazoval tudi hitrost konvergence posameznih metod.

Problem enakomerno obtežene opne ima rezultat, ki je skalarna spremenljivka in je enak povprečnemu povesu opne. Enaka Poissonova enačba opisuje tudi tok tekočine po cevi, kjer ima ta rezultat bolj nazoren fizikalen pomen skupnega pretoka po cevi. Na grafu 3 sem prikazal, kako se ta pretok približuje končni vrednosti, ko povečujemo število točk.



Slika 3: Konvergenca skupnega pretoka po cevi

Za kvadratno cev poznamo tudi točno rešitev, ki znaša  $\Phi_{\square} \approx 0.0351342$ . Natančnost metod torej bolj vidimo na logaritmskem grafu napake 4.



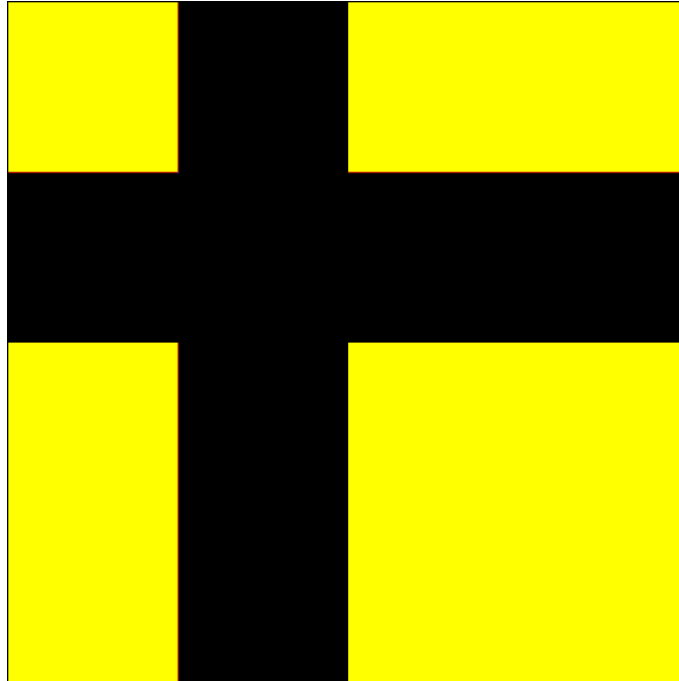
Slika 4: Napaka skupnega pretoka po cevi

1D in 2D FFT imata obe približno linearno konvergenco, 2D FFT pa je vseskozi malo natančnejša. Opazimo pa lahko tudi, da relaksacijska metoda odpove že pri  $100 \times 100$  točkah. Razlog za to je verjetno v slabo izbranem pogoju za ustavitev. Žal pa zaradi počasne konvergence nisem mogel natančneje rešiti problema za večje število točk. Če bi mejo za ustavitev zniževal, bi se natančnost seveda izboljšala, ampak bi račun postal še bolj dolgotrajen. V nobenem primeru pa

ne bi dobil boljšega rezultata kot z direktnim reševanjem.

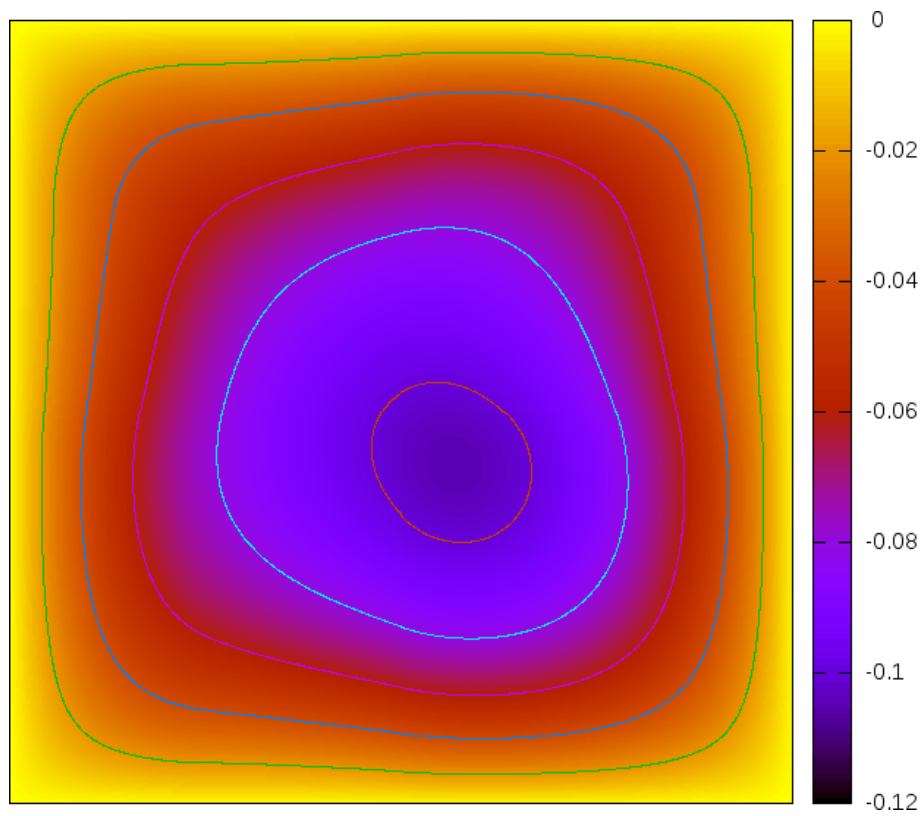
## 2 Obtežena opna

Po preizkušanju metod sem zaključil, da je najboljša 2D FFT, ki je tudi najenostavnejša za implementacijo. Z njo sem izračunal obliko neenakomerno obtežene opne. Oblike opne je na sliki 5

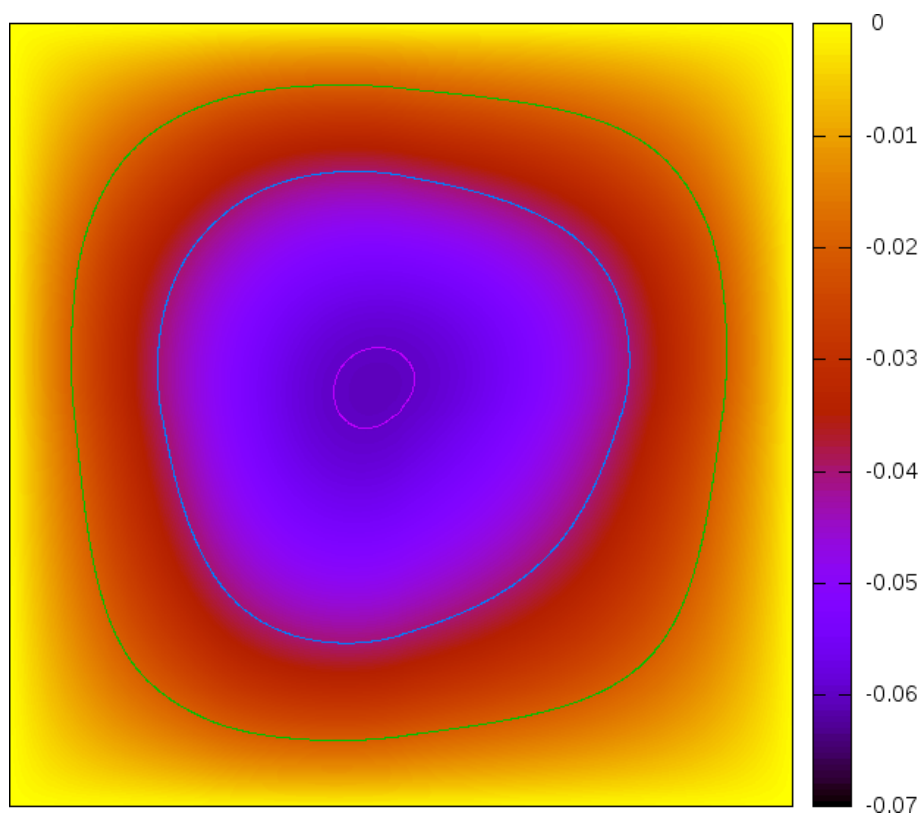


Slika 5: Oblika opne s prikazanimi območji večje in manjše obtežitve

Na slikah 6 in 7 sta ravnovesni obliki opne z različnima razmerjima gostote, izračunane z mrežo  $1024 \times 1024$  točk.



Slika 6: Oblika opne, ki je na robovih dvojno obtežena



Slika 7: Oblika opne, ki je na robovih polovično obtežena

### 3 Prevajanje toplote po valju

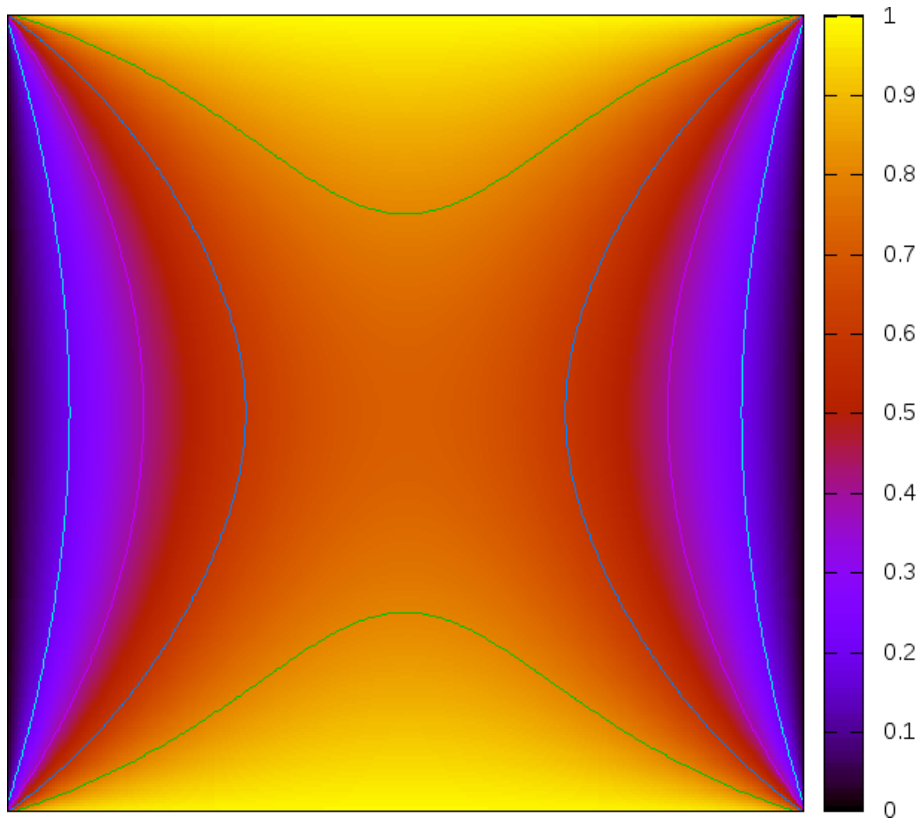
Za reševanje enačb v valjni geometriji ne moremo uporabiti Fourierove v obeh smereh, saj operator  $\nabla^2$  v radialni smeri ne vsebuje le drugega odvoda po  $r$ . Zato sem FFT izvedel samo v smeri  $z$ , v radialni smeri pa rešil matrični sistem.

Zaradi linearnosti enačbe sem problem lahko poenostavil, tako da sem eno izmed temperatur postavil na 0, drugo pa na 1. Nehomogeni robni pogoj bi lahko dal na vrh in dno valja (in s tem otežil FFT) ali na plašč (in otežil reševanje matričnega sistema). Odločil sem se, da nehomogen robni pogoj postavim na stran Fourierove transformacije in ga pridružim gostoti izvirov, kot je opisano v navodilih. Povsod razen na zgornji in spodnji ploskvi je gostota izvirov enaka nič.

Diferencialno enačbo, ki jo rešujemo v transformiranem prostoru sem popravil, tako da sem dodal člen s prvim odvodom po  $r$ .

$$\left(1 - \frac{h}{2r}\right)U_{l-1}^m - (4 - 2\cos(m\pi/J))U_l^m + \left(1 + \frac{h}{2r}\right)U_{l+1}^m = h^2 G_l^m \quad (3)$$

Računal sem s celotnim valjem, tako da je  $r$  šel od  $-1/2$  do  $+1/2$ .



Slika 8: Temperaturni profil po preseku valja