

# 09 - SPEKTRALNE METODE ZA ZAČETNE PROBLEME PDE

Matematično-fizikalni praktikum, avgust 2024

Luka Skeledžija, 28201079

---

## 1. Uvod

Za reševanje začetnih problemov s parcialnimi diferencialnimi enačbami (PDE) imamo na voljo dva obsežna razreda metod. Pri *diferenčnih metodah* na neki način aproksimiramo časovne in krajevne parcialne odvode s končnimi diferencami.

Reševanje PDE nato prevedemo na reševanje algebrajskih enačb ali sistemov enačb za približne vrednosti funkcij, ki v teh differencah nastopajo. Diferenčne metode spoznamo pri naslednji vaji. Pri tej vaji obravnavamno *spektralne metode*: pri njih iskano rešitev formalno izrazimo z nekim naborom funkcij, nato pa v času spremlijamo koeficiente v takem razvoju. Kako se selimo med krajevno in časovno sliko problema, je odvisno od posamezne metode. Osnovne prijeme spoznamo ob Fourierovi metodi in metodi končnih elementov s kubičnimi  $B$ -zlepki ( $B$ -splines).

Fizikalno ozadje naj bo enorazsežna difuzijska enačba, ki opisuje na primer temperaturno polje  $T(x, t)$  v homogeni neskončni plasti s končno debelino  $a$  brez izvorov topote:

$$\frac{\partial T}{\partial t} = D \frac{\partial^2 T}{\partial x^2}, \quad 0 \leq x \leq a, \quad D = \frac{\lambda}{\rho c}.$$

Temperaturo v poljubni točki  $x$  ob času  $t$  izrazimo s Fourierovo vrsto

$$T(x, t) \simeq \sum_{k=0}^{N-1} \tilde{T}_k(t) e^{-2\pi i f_k x},$$

kjer je  $f_k = k/a$ , torej

$$\sum_k \frac{\partial \tilde{T}_k(t)}{\partial t} e^{-2\pi i f_k x} = D \sum_k (-4\pi^2 f_k^2) \tilde{T}_k(t) e^{-2\pi i f_k x}.$$

Od tod sledi evolucijska enačba za Fourierove koeficiente

$$\frac{\partial \tilde{T}_k(t)}{\partial t} = D(-4\pi^2 f_k^2) \tilde{T}_k(t). \quad (1)$$

Pogosto uporabimo spekralno reprezentacijo za krajevni odvod, medtem ko časovni korak naredimo z neko eksplisitno integracijsko shemo, na primer kar po Eulerju

$$\tilde{T}_k(t+h) = \tilde{T}_k(t) + hD(-4\pi^2 f_k^2) \tilde{T}_k(t). \quad (2)$$

Reprezentacijo  $T(x, t)$  v običajnem prostoru nato dobimo z obratno Fourierovo transformacijo.

Tu lahko v splošnem časovni korak izvedeš po Eulerju, v tem konkretnem primeru pa obstaja tudi enostavna analitična rešitev enačbe ???, ki jo lahko uporabiš za primerjavo. V numerični metodi tako najprej izračunaj Fourierovo reprezentacijo  $\tilde{T}_k(0)$  začetnega pogoja, nato pa jo po Eulerju evolviraj v času. Pri tem moraš paziti na stabilnost Eulerjeve diferenčne sheme: pri katerem koli koraku mora veljati

$$\left| \frac{\tilde{T}_k(t+h)}{\tilde{T}_k(t)} \right| = \left| 1 + hD(-4\pi^2 f_k^2) \right| < 1.$$

Nekaj pozornosti zahteva tudi diskretizacija: za vsak  $k$  seveda velja  $-f_{\text{Nyquist}} < f_k < f_{\text{Nyquist}}$  in s tem povezan možen pojav \emph{aliasinga} (Kaj je že to?). Ta pojav lahko \v studira\v s, \v ce se spomni\v s, da obstaja analiti\v cna re\v sitev FT Gaussove funkcije (je spet Gaussova funkcija) - kaj se z le-to dogaja znotraj dovoljenega frekvenčnega intervala? \v Ce izbereš veliko število točk, je seveda smiselno uporabiti kar algoritem FFT. Temperaturni profil  $T_j(t) \equiv T(x, t)$  ob poljubnem času nato dobiš z inverzno FFT.

Pri razvoju  $T(x, t)$  nismo omejeni na trigonometrične funkcije. Rešitev PDE na  $0 \leq x \leq a$  lahko aproksimiramo tudi z drugačno vrsto funkcij, na primer kubičnimi  $B$ -zlepki,

$$T(x, t) = \sum_{k=-1}^{N+1} c_k(t) B_k(x), \quad (3)$$

kjer je  $B_k(x)$  kubični zlepek s središčem okrog  $x = x_k$ . Aastnosti  $B$ -zlepkov so navedene v dodatku. Tako zasnujemo \sl metodo končnih elementov, s \emph{kolokacijskim pogojem}, da naj se zlepek ujema z rešitvijo v določenih izbranih točkah. Podobno kot pri Fourierovi metodi tudi pri tej metodi zahtevamo, da razvoj zadošča osnovni PDE in robnim pogojem. Razvoj vstavimo v PDE in izvrednotimo rezultat pri  $x = x_j$ . (Interval  $[0, a]$  diskretiziramo na  $N$  podintervalov širine  $\Delta x$  s točkami  $x_j = j\Delta x$ , kjer je  $j = 0, 1, \dots, N$ . Za kolokacijo je smiselno izbrati enake točke kot za diskretno mrežo.) Tako dobimo

$$\sum_{k=-1}^{N+1} \dot{c}_k(t) B_k(x_j) = D \sum_{k=-1}^{N+1} c_k(t) B''_k(x_j), \quad j = 0, 1, \dots, N.$$

Upoštevamo lastnosti  $B$ -zlepkov in dobimo sistem diferencialnih enačb za koeficiente  $c_j(t)$ :

$$\dot{c}_{j-1}(t) + 4\dot{c}_j(t) + \dot{c}_{j+1}(t) = \frac{6D}{\Delta x^2} (c_{j-1}(t) - 2c_j(t) + c_{j+1}(t)),$$

kjer je  $j = 0, 1, \dots, N$ . Iz robnega pogoja pri  $x = 0$  ugotovimo  $c_{-1} = -4c_0 - c_1$ . Če dodamo še zahtevo za 'naravni' kubični zlepki, da je na robu

$\sum_{k=-1}^{N+1} c_k(t) B''_k(x = (0, a)) = 0$ , sledi  $c_0 = c_N = 0$  in  $c_{-1} = -c_1$  ter  $c_{N-1} = -c_{N+1}$ . Reševanje prejšnje enačbe smo torej prevedli na reševanje matričnega sistema

$$\mathbf{A} \frac{d\vec{c}}{dt} = \mathbf{B}\vec{c},$$

kjer je

$$\mathbf{A} = \begin{pmatrix} 4 & 1 & & & \\ 1 & 4 & 1 & & \\ 1 & 4 & 1 & & \\ & & \vdots & & \\ & & 1 & 4 & 1 & \\ & & & 1 & 4 & 1 & \\ & & & & 1 & 4 & \\ & & & & & 1 & 4 \end{pmatrix},$$

$$\mathbf{B} = \frac{6D}{\Delta x^2} \begin{pmatrix} -2 & 1 & & & \\ 1 & -2 & 1 & & \\ 1 & -2 & 1 & & \\ & & \vdots & & \\ & & 1 & -2 & 1 & \\ & & & 1 & -2 & 1 & \\ & & & & 1 & -2 \end{pmatrix}$$

in  $\vec{c} = (c_1(t), c_2(t), \dots, c_{N-1}(t))^T$ . Začetni pogoj za PDE je  $T(x_j, 0) = f(x_j)$ , torej je začetni približek za kolokacijsko aproksimacijo

$$\mathbf{A}\vec{c}^0 = \vec{f},$$

kjer je  $\vec{f} = (f(x_1), f(x_2), \dots, f(x_{N-1}))^T$ . To zdaj rešujemo s kako metodo, ki jo poznamo iz prejšnjih nalog, recimo z eksplicitno Eulerjevo metodo: ob zaporednih časih  $n\Delta t$  dobimo

$$\vec{c}^{n+1} = \vec{c}^n + \Delta t A^{-1} B \vec{c}^n = (1 + \Delta t A^{-1} B) \vec{c}^n.$$

Ob poljubnem času nato dobimo temperaturni profil tako, da znova izračunamo vsoto.

Ker nam je že znano, da je Eulerjeva ob predolgih časovnih korakih lahko nestabilna, lahko uporabimo stabilno implicitno metodo, kjer v vsakem časovnem koraku rešujemo

$$\left( A - \frac{\Delta t}{2} B \right) \vec{c}^{n+1} = \left( A + \frac{\Delta t}{2} B \right) \vec{c}^n.$$

## 2. Naloga

Reši difuzijsko enačbo v eni razsežnosti  $x \in [0, a]$

z začetnim pogojem po plasti gaussovsko porazdeljene temperature

$$T(x, 0) \propto e^{-(x-a/2)^2/\sigma^2}$$

(izberi razumne vrednosti za  $D$ ,  $a$  in  $\sigma$ ) in

1. periodičnim robnim pogojem  $T(0, t) = T(a, t)$ .
2. homogenim Dirichletovim robnim pogojem  $T(0, t) = T(a, t) = 0$ .

po Fourierovi metodi in homogenih Dirichletovih robnih pogojih  $T(0, t) = T(a, t) = 0$  ter primerjaj obe metodi. Kolokacijsko metodo uporabi ob Gaussovem začetnem pogoju

**Dodatna naloga:** Izberi si še kakšen primer začetnih pogojev, recimo:

$$T(x, 0) = f(x) = \sin(\pi x/a)$$

in preizkusi obe metodi.

## 3. Reševanje PDE

Najprej si poglejmo ustrezne začetne pogoje. Periodični robeni pogoji  $T(0, t) = T(a, t)$  si lahko fizikalno interpretiramo kot da je palica 1 telo in je toplotno izolirana od okolice – toplota nikamor ne odteka. V ozadju homogenih (Dirichletovih) robnih pogojev  $T(0, t) = T(a, t) = 0$  pa stoji predpostavka, da iz robov palice ves čas črpamo toploto in tako vzdržujemo konstantno temperaturo. Začetni pogoji je v obeh primerih Gaussova porazdelitev temperature s središčem v  $a/2$ .

### 3.1. Fourierova metoda

Najprej analizirajmo enačbo s homogenimi robnimi pogoji. Difuzijsko lahko poenostavimo z uvedbo spremenljivk  $\tau$  in  $\xi$ , tako da jo prepišemo v obliko

$$\frac{\partial T}{\partial \tau} = \frac{\partial^2 T}{\partial \xi^2},$$

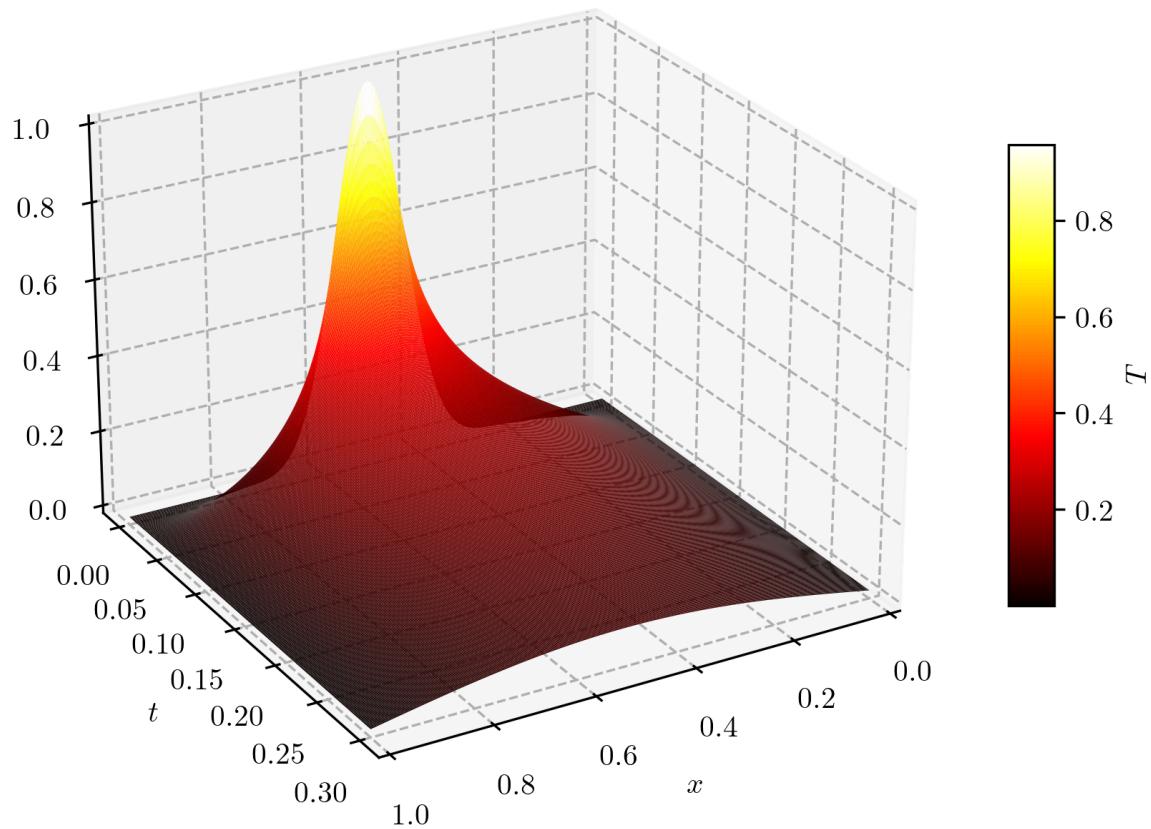
kjer je  $\tau = Dt/a^2$  in  $\xi = x/a$ . Arbitrarno izberemo vrednosti  $D = 0.5$  in  $a = 1$ .

Temperatura je porazdeljena Gaussovsko in sicer kot

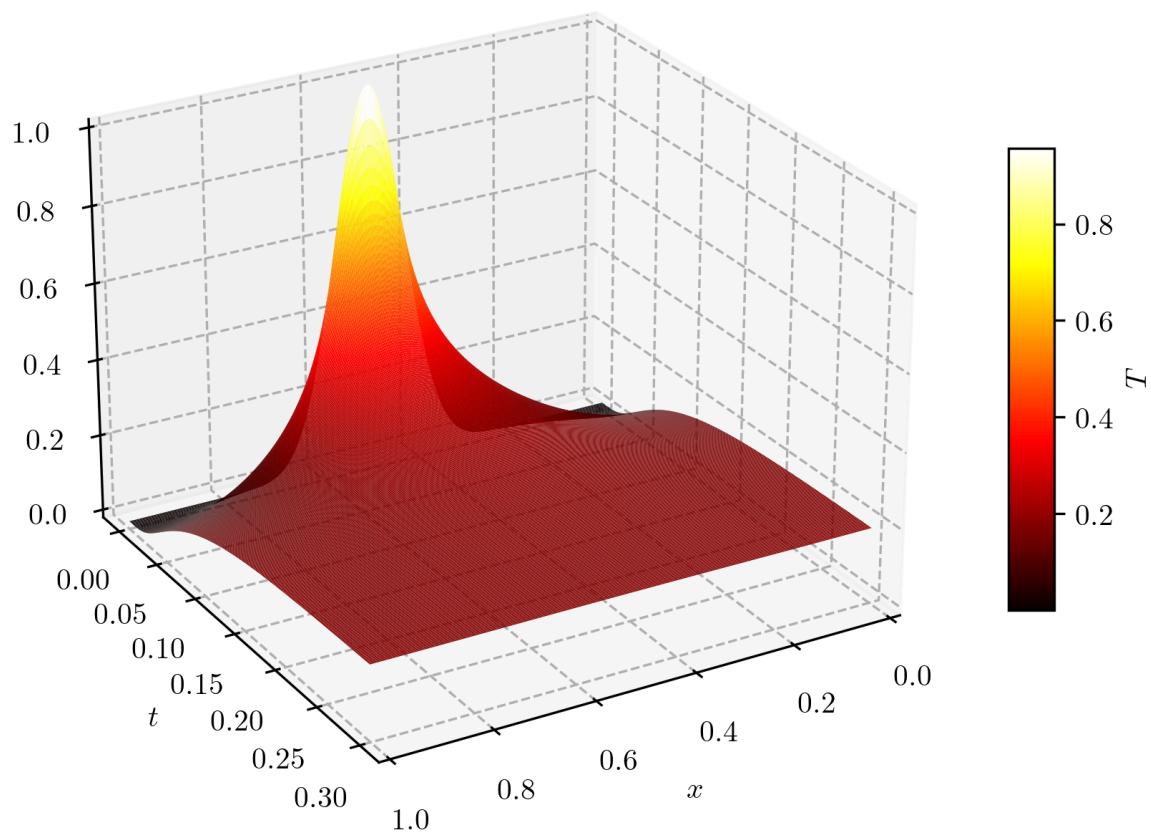
$$T(x, 0) = e^{-(x-x_0)^2/\sigma^2}$$

za vrednosti  $\sigma = a/10$  in  $x_0 = a/2$ . Da zadostimo homogenim robnim pogojem uporabimo lastnost, da gre liha funkcija lahko simetrično skozi izhodišče in naredimo liho razširitev začetnega pogoja.

### Homogeni robni pogoji, Fourierova metoda



## Periodični robni pogoji, Fourierova metoda

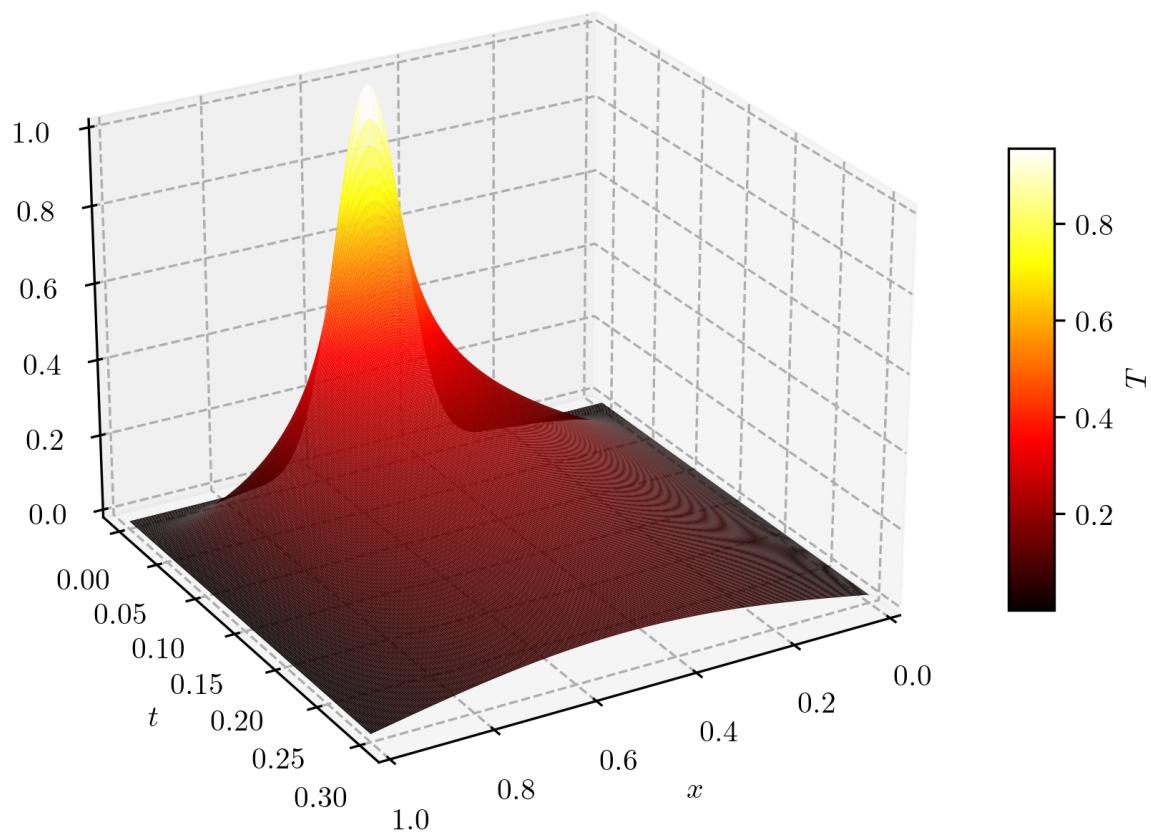


Največjo razliko opazimo po daljšem času. Pri homogenih robnih pogojih se oblika Gaussovke ohranja, energija iz sistema odteka. V primeru periodičnih robnih pogojev pa dosežemo temperaturno ravnovesje.

### 3.1. Metoda končnih elementov

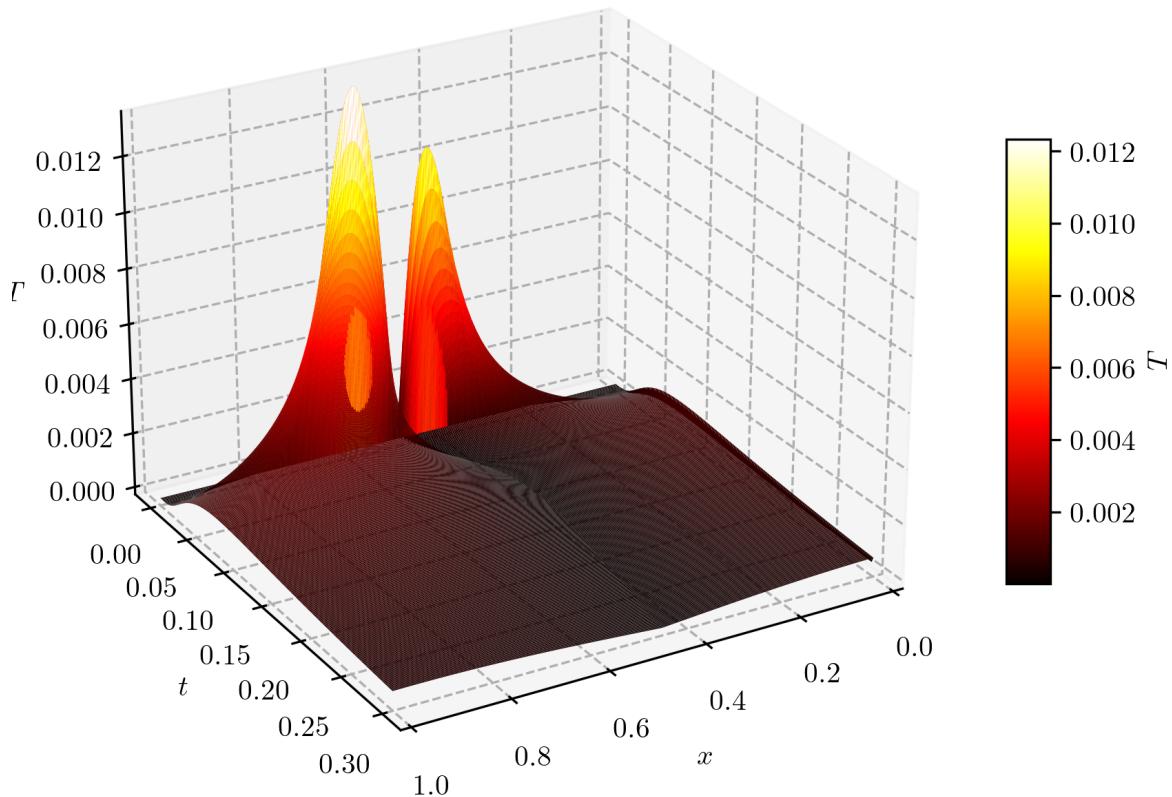
Rešimo PDE se z metodo končnih elementov. Kvalitativno gledano je rezultat enak kot zgoraj.

## Homogen robni pogoj, Metoda končnih elementov

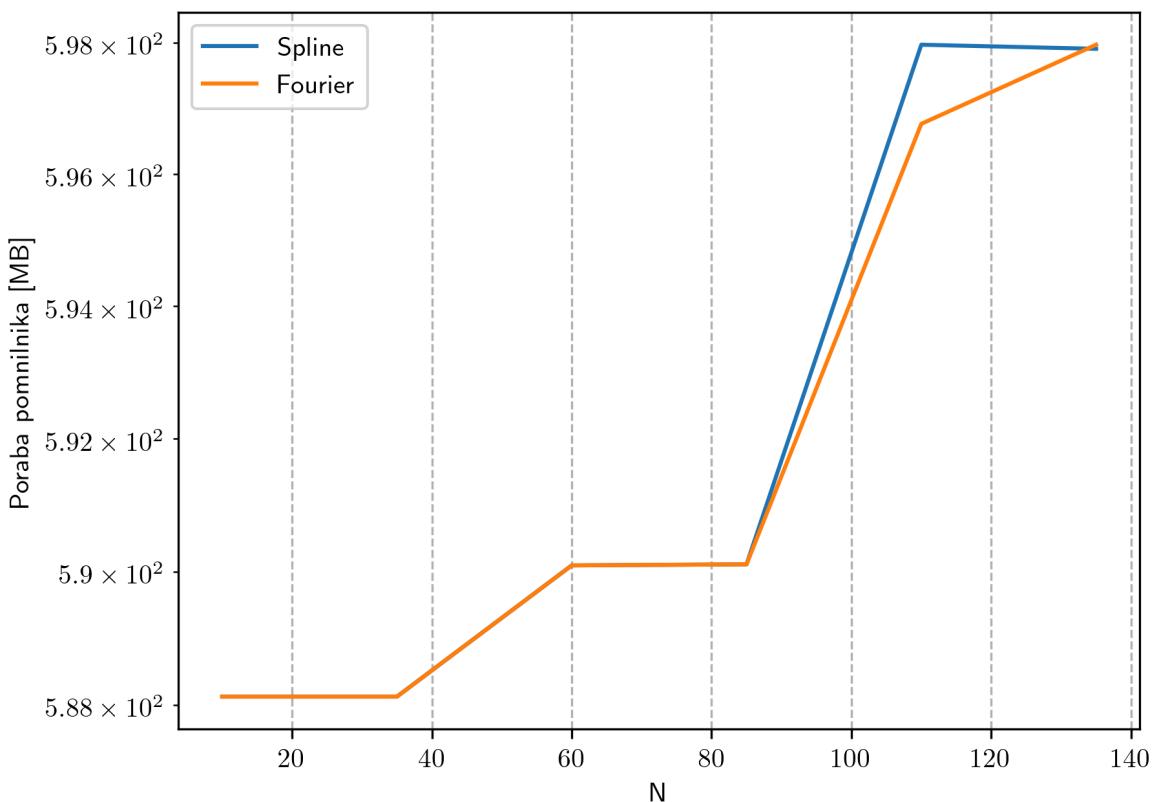
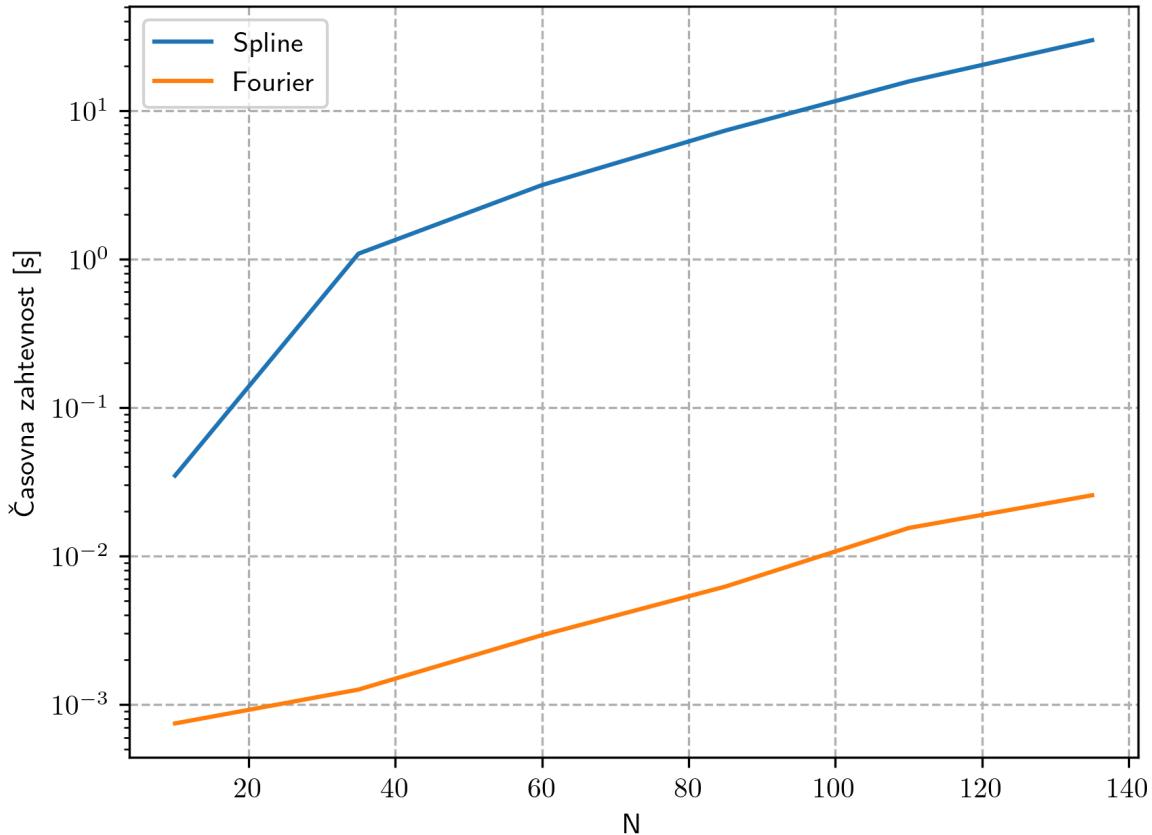


Izračunajmo še abs. razliko obeh metod pri  $N = 100$ . Odstopanja so manjša od 0.012.

## Abs. razlika med Fourierovo metodo in metodo končnih elementov



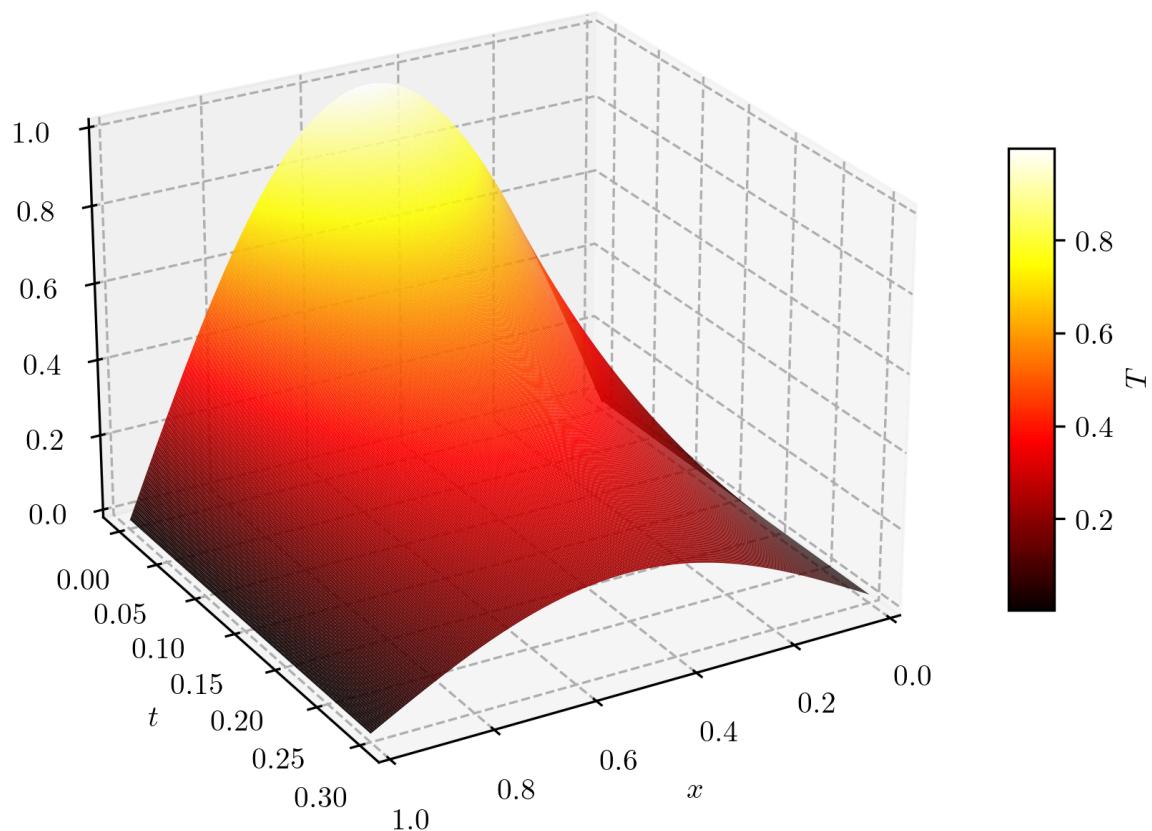
Če primerjamo hitrost obeh metod, vidimo, da je Fourierova metoda nekaj velikostnih redov hitrejša. Kot dodatek smo želeli primerjati tudi porabo pomnilnika, vendar se nam to ni najbolj izšlo - nisem povsem prepričan v interpretacijo rezultatov. Verjetno bi morali programe poganjati prek zunanjih skript, ne pa samo preverjati porabe znotraj Pythona samega. Ali morda izbrati kakšno implementacijo v jeziku C++ - tam bi se dalo to veliko bolje oceniti, ker se dejansko izvaja samo napisan program in ne celotna interpretacija jezika Python.



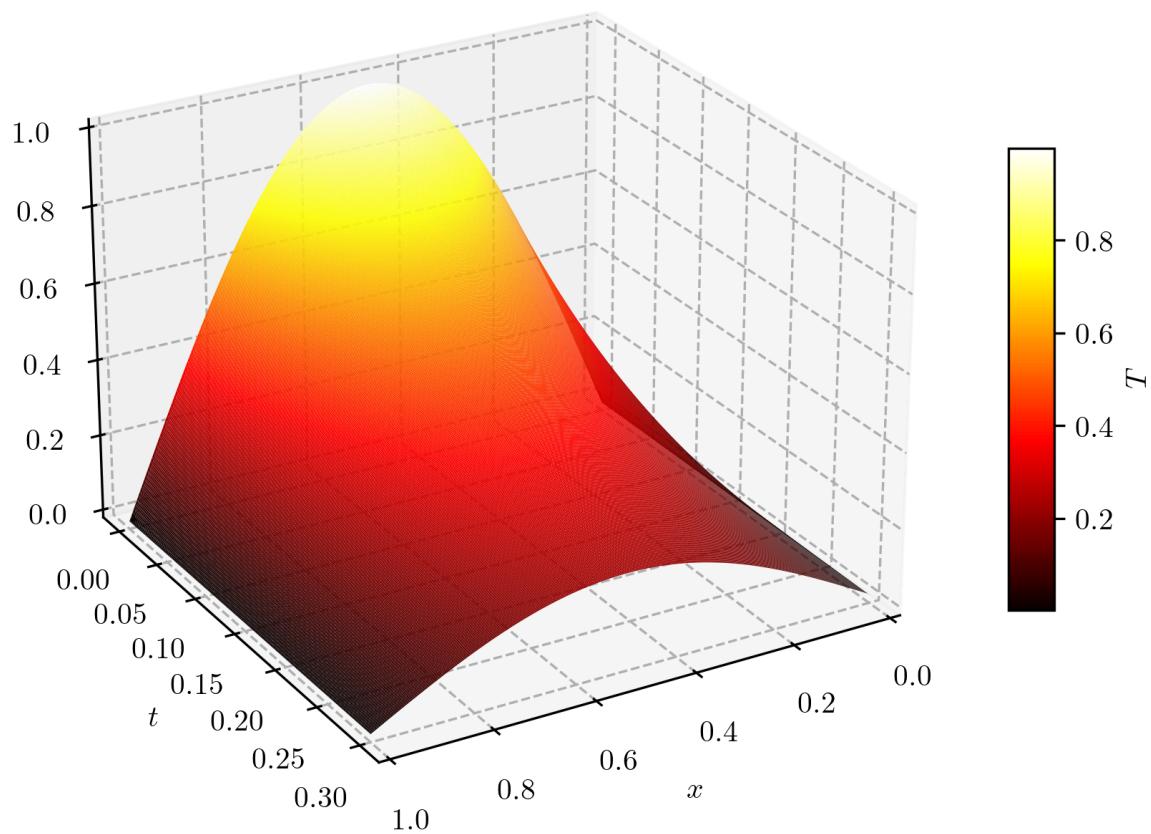
## 4. Dodatek: Sinusni začetni pogoj

Analiziramo celotno stvar še enkrat, le da si za začetni pogoj izberemo sinusno krivuljo  $T(x, 0) = f(x) = \sin(\pi x/a)$ .

## Homogeni robni pogoji, Fourierova metoda

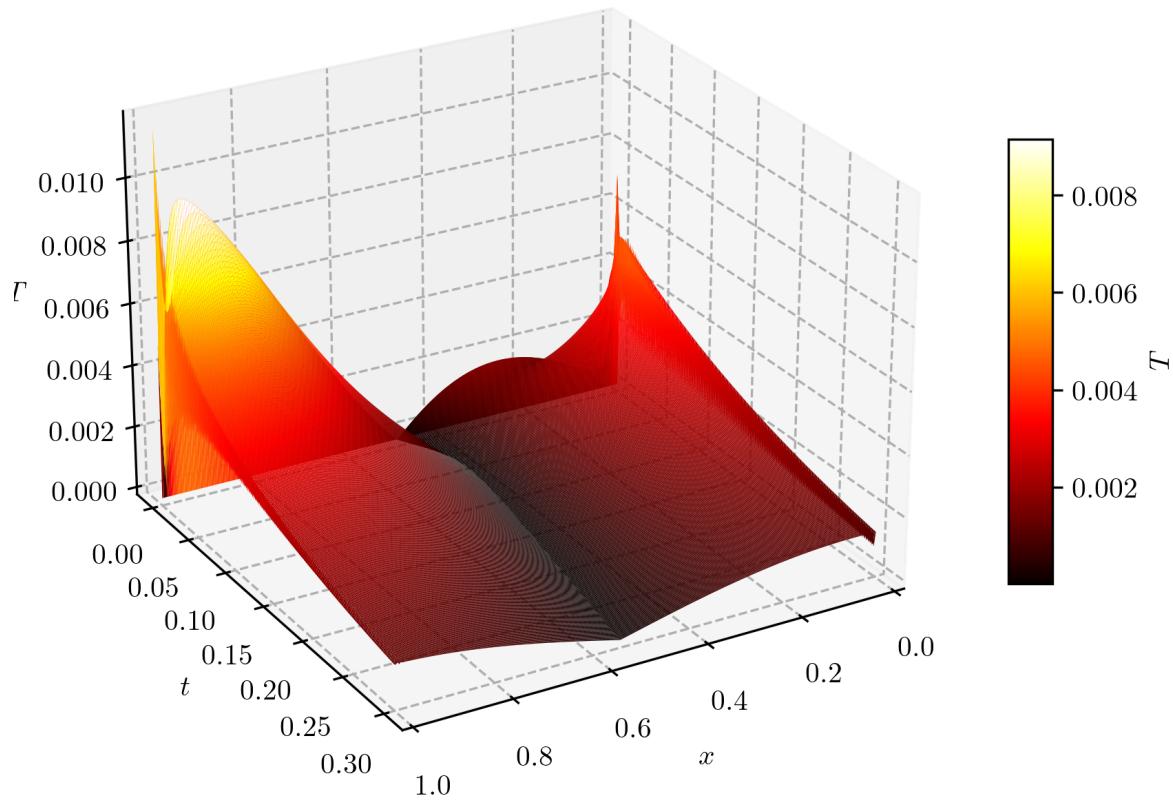


## Homogeni robni pogoj, Metoda končnih elementov

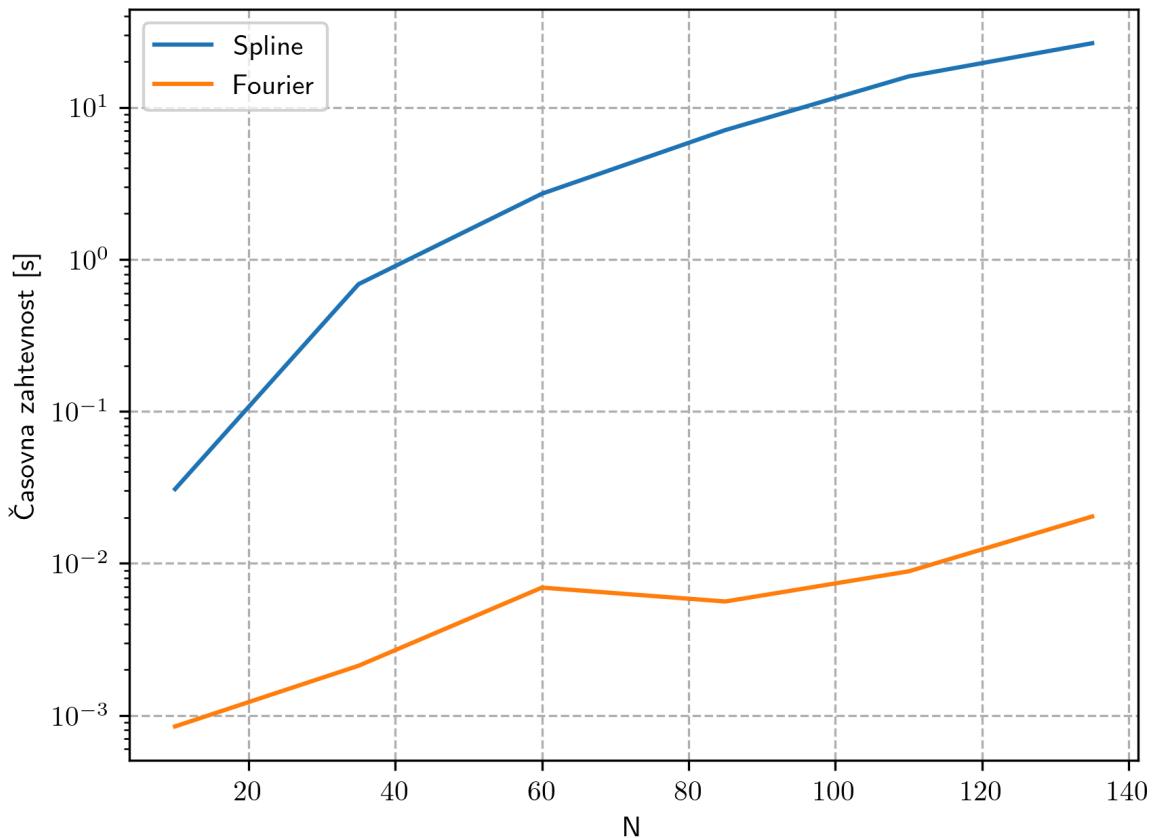


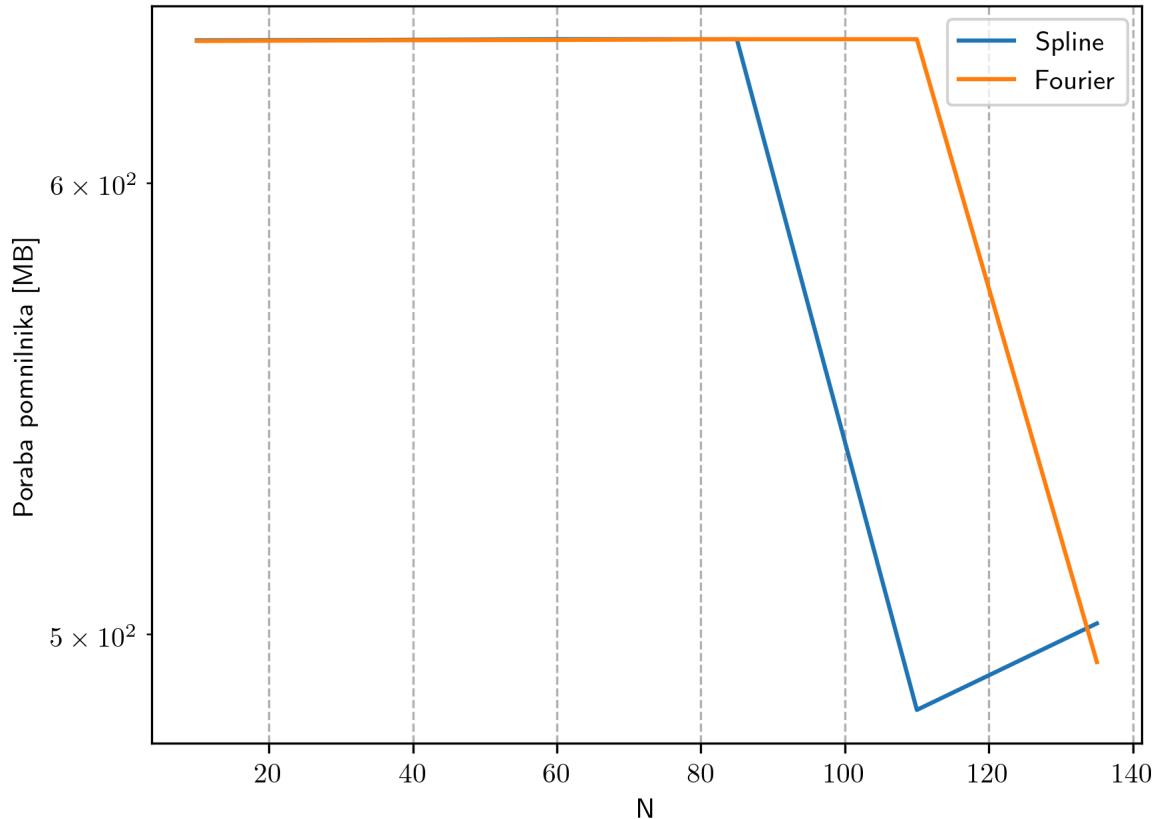
Rezultati so podobni kot prej, vendar malo drugačne oblike. Napaka med metodama je malce manjša, ampak brez bistvenih razlik.

## Abs. razlika med Fourierovo metodo in metodo končnih elementov



Kot prikazano spodaj, je Fourierova metoda bistveno hitrejša. Meritev porabe pomnilnika pa še vedno izgleda relativno slabo in nezanesljivo.





## 5. Zaključek

Pri tej vaji smo primerjali dve spektralni metodi za reševanje enorazsežne difuzijske enačbe. Fourierova metoda se je izkazala za bistveno hitrejšo od metode končnih elementov, zlasti pri večjem številu točk. Razlike natančnosti med metodama so pri  $N = 100$  točkah manjše od 0.012. Izračuni porabe pomnilnika so bili manj zanesljivi; verjetno bi bilo smiselno preučiti alternativne pristope ali implementacijo v jeziku C++. Zdi se mi, da je Fourierova metoda boljša izbira glede na hitrost in enostavnost implementacije.

---

Luka Skeledžija, [Github source](#) , 2024