Univerza v Ljubljani
Fakulteta za strojništvo

Študijsko leto 2020/2021

Seminarska naloga:

Reševanje toplotne enačbe z uporabo Fourierove transformacije

Matematika 4

Avtor: Žiga Perne

Mentor: prof. dr. Aljoša Peperko

Ljubljana, februar 2021

Kazalo

1. Uvod

1.1. Fourierova transformacija

Fourierova transformacija je matematična transformacija, ki transformira funkcije časa $f(t)$ ali prostora $f(x)$ v frekvenčno funkcijo $\hat{f}(w)$. Vir [1] definira transformacijo kot:

$$\hat{f}(w) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} f(x) e^{-iwx} dx \quad (1)$$

Poznamo tudi inverzno Furierevo transformacijo:

$$f(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \hat{f}(w) e^{iwx} dw \quad (2)$$

Furierova transformacija funkcije $f(x)$ obstaja, če je funkcija integrabilna na osi x in odsekoma zvezna na vsakem končnem intervalu.

Transformacijo lahko označimo tudi kot:

$$\hat{f} = \mathcal{F}(f) \quad (3)$$

1.2. Pravilo odvoda

Pomembna lastnost Fourierove transformacije, ki jo bomo uporabili pri reševanju problema, ki smo si ga zadali, je pravilo odvoda:

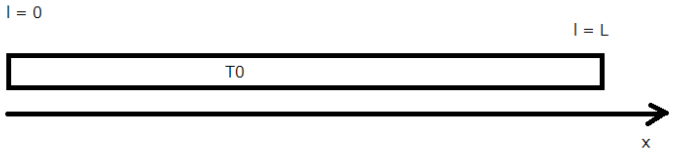
$$\mathcal{F}\{f'(x)\} = iw\mathcal{F}\{f(x)\} \quad (4)$$

$$\mathcal{F}\{f''(x)\} = i^2 w^2 \mathcal{F}\{f(x)\} = -w^2 \mathcal{F}\{f(x)\} \quad (5)$$

2. Definicija naloge

Obravnavamo ravno palico dolžine L = 1 iz homogenega materiala in konstantnega prečnega preseka v eni dimenziji. Pri času $t = 0$ predpostavimo popolnoma enakomerno temperaturo T_0 na celotni palici. Predpostavimo, da toplotni tok teče samo v x-smeri. Na palici ni izvora ali ponora toplote. Koeficient prevoda toplote je 0,025.

Določiti želimo, kako se temperatura na palici spreminja s časom.



Slika 1: Shematski prikaz palice

Robni pogoji:

$$T(0,t) = 0, \qquad T(L,t) = 0 \quad za\ vse\ t \geq 0$$

Začetna porazdelitev temperature:

$$T(x,0) = \begin{cases} 1; & \frac{L}{2} - \frac{L}{10} \leq x \leq \frac{L}{2} + \frac{L}{10} \\ 0; & Povsod\ drugod \end{cases}$$

3. Nastavitev enačb

Izhajamo iz toplotne enačbe za enodimenzijsko telo, ki nam poda temperaturo T v odvisnosti od pozicije x in časa t:

$$T = f(x,t) \tag{6}$$

Enačba se glasi:

$$\frac{\partial T}{\partial t} = c^2 \frac{\partial^2 T}{\partial x^2} \tag{7}$$

Opravili bomo Fourierovo transformacijo prostorske spremenljivke x, torej:

$$T(x,t) \overset{\mathcal{F}}{\Rightarrow} \hat{T}(w,t) \tag{8}$$

Opazimo, da transformirana oblika funkcije ostane v odvisnosti od časa, prostorsko spremenljiko pa transformiramo v prostorsko frekvenco w.

Če uporabimo pravilo odvodov, vidimo da je:

$$T_x \overset{\mathcal{F}}{\Rightarrow} iw\hat{T} \tag{9}$$

$$T_{xx} \overset{\mathcal{F}}{\Rightarrow} -w^2\hat{T} \tag{10}$$

Ker je $\hat{T}$ funkcija ćasa, lahko zapišemo diferencialno enačbo:

$$\hat{T}_t = -c^2w^2\hat{T} \tag{11}$$

4. Numerično reševanje v programskem jeziku Python

Enačbo bomo rešili numerično v programskem jeziku Python z uporabo algoritma FFT (*Fast Fourier Transform*, "Hitra Fourierova transformacija"), ki zračuna diskretne Fourierove transformacije v vsaki točki dane sekvence. Uporabili bomo FFT, ki je na voljo v knjižnjici NumPy. V viru [2] lahko dostopamo do dokumentacije za izbran algoritem. Definiran je kot:

$$y[k] = \sum_{n=0}^{N-1} e^{-2\pi i \frac{kn}{N}} x[n] \tag{12}$$

Inverz je definiran kot:

$$x[n] = \frac{1}{n}\sum_{k=0}^{N-1} e^{2\pi i \frac{kn}{N}} y[k] \tag{13}$$

V našem primeru bomo torej v algoritem FFT poslali funkcijo $\hat{T}(t)$, začetne pogoje, diskretno tabelo časov z določenim časovnim korakom, vektor frekvenc w ter konstanto c.

Ker imamo opravka z diskretnimi vrednostmi, lahko obravnavamo enačbo (10) kot produkt vektorjev frekvenc in Forierovih koeficientov na vsaki diskretni točki:

$$-w^2\hat{T} = -\begin{bmatrix} w_1^2\widehat{T_1} \\ w_2^2\widehat{T_2} \\ \dots \\ w_n^2\widehat{T_n} \end{bmatrix} \tag{14}$$

Iz enačbe (11) lahko tako zapišemo sistem n neodvisnih navadnih diferencialnih enačb, ki jih bomo rešili z ScyPy funkcijo *odeint.*

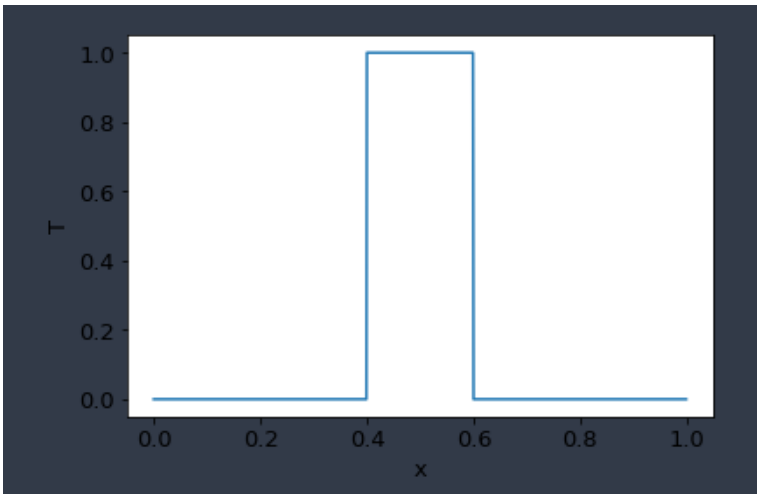
S tem lahko pričnemo z reševanjem problema v Python-u. Najprej definiramo domeno, čez katero rešujemo diferencialno enačbo (11). Prikaz definicije je na sliki 2:

```
c = 0.025 #difuzivna konstanta
l = 1 #dolžina
n = 1000 #št. diskretizacijskih točk
dx = l/n
x = np.arange(0, l, dx)

w = 2*np.pi*np.fft.fftfreq(n, d = dx)

T0 = np.zeros(len(x))
for i in range(int(n/2 - n/10), int(n/2 + n/10)):
    T0[i] = 1
```

Slika 2: definicija domene

Spremenljivka c predstavlja koeficient prevoda toplote, l pa dolžino domene. Spremenljivka n nam predstavlja število diskretizacijskih točk, v našem primeru smo jih izbrali 1000. V tabelo x spravimo seznam pozicij, na katerih mestih bomo reševali enačbe. Definirati moramo še vektor frekvenc, ki ga dobimo z funkcijo *np.fft.fftfreq()*, ki je del knjižnjice NumPy.

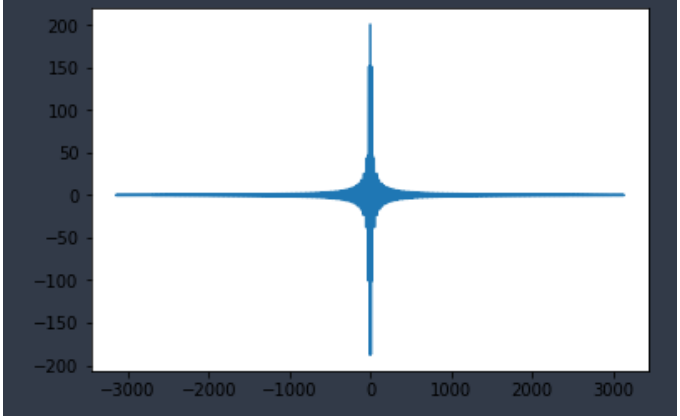
Ostane nam še definicija začetne porazdelitve temperature v tabeli T0. Da se prepričamo, da smo začetne pogoje pravilno popisali, lahko izrišemo T0 v odvisnosti od x, prikazano na sliki 3:



Slika 3: Začetni pogoji

Tudi vizualno lahko potrdimo, da smo T0 pravilno definirali. Sledi Fourierjeva transformacija T0, kar naredimo enostavno z funkcijo *np.fft.fft()*. Na sliki 4 je prikazana definicija ter graf frekvenčne domene za T0:

```
T0_ = np.fft.fft(T0)
plt.plot(w, T0_);
```



Slika 4: prikaz transformirane funkcije začetnih pogojev

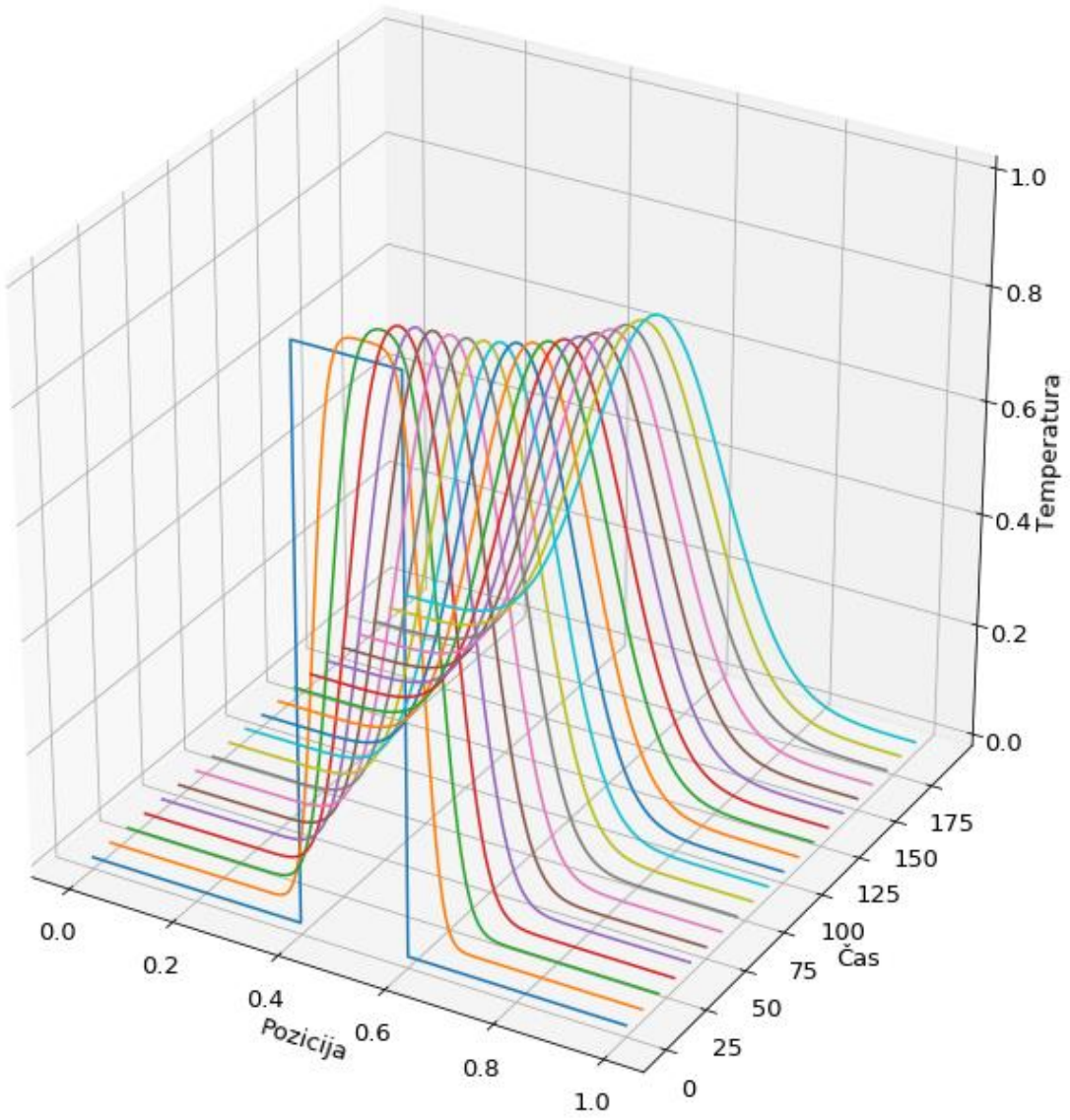
Sledi samo reševanje diferencialne enačbe. Prikazano je na sliki 5:

```
def toplotna_enacba(T_locena, t, w, c):
    T_ = zdruzi_kompleksna(T_locena)
    dT_ = - np.power(c, 2) * np.power(w, 2) * T_
    return(loci_komlpeksna(dT_).astype("float64"))

T_locena = odeint(toplotna_enacba, T0_locena, t, args = (w, c))
T_ = np.array([])
for i in range(0, len(T_locena)):
    T_ = np.append(T_, zdruzi_kompleksna(T_locena[i]))
T_ = T_.reshape(int(10 / dt), 1000)

T = np.fft.ifft(T_).real
```

Slika 5: Potek reševanja diferencialne enačbe

Funkcija *toplotna_enacba* predstavlja desno stran enačbe (11). Na levi strani enačbe pa imamo prvi odvod transformirane funkcije temperature po času $\hat{T}_t$. Da dobimo dejansko vrednost $\hat{T}$, integriramo $\hat{T}_t$ glede na čas z uporabo SciPy funkcije *odeint* (ki ne sprejema kompleksnih števil, zato jih je potrebno ločiti na realno in imaginarno komponento), kot je vidno na sliki 5. Ko imamo enkrat vrednosti $\hat{T}$, lahko z uporabo inverzne Fourierove transformacije *np.fft.ifft* pridobimo iskano vrednost T.

S tem smo zaključili reševanje danega problema. Preostane nam še izris grafa časovnega poteka temperature v palici, ki je prikazan na sliki 6:



Slika 6: Graf temperature v odvisnosti od pozicije in časa

5. Zaključek

V seminarski nalogi je bil prikazan postopek numeričnega reševanja diferencialnih enačb z uporabo fourierove transformacije. Za reševanje je bil uporabljen algoritem FFT. Rešitev enačbe lahko ocenimo kot ustrezno, saj je izračunan potek temperature v skladu s pričakovanji. Zraven te seminarske naloge je oddana tudi datoteka z Python kodo, kjer so prikazane tudi nekatere pomožne funkcije, s katerim je bila rešena enačba.

Literatura

[1] Erwin Kreyszig: Advanced Engineering Mathematics. Deseta izdaja (str. 522-531).

[2] Dokumentacija NumPy: fft (https://numpy.org/doc/stable/reference/routines.fft.html). Dostop 20.2.2021

[3] Wikipedia: Fourier Transform (https://en.wikipedia.org/wiki/Fourier_transform). Dostop 20.2.2021