# Izračun porazdelitve temperature v 2D prerezih

Luka Berlec, Nejc Facija

23. januar 2024

# Kazalo



# Slike

# 1 Teorija

## 1.1 Teorija prenosa toplote v 2D prerezu

V tem projektu smo se ukvarjali s časovno neodvisnim prenosom toplote v 2D prerezu. Naš cilj je bil izračunati porazdelitev temperature v materialu ob danih robnih pogojih, kjer smo predpostavili temperaturno neodvisno toplotno prevodnost in konstantne robne pogoje, hkrati pa smo zanemarili tudi notranjo generacijo toplote.

Pri teh pogojih in predpostavki, je naša glavna diferencialna enačba za prenos toplote po trdnini, definirana kot:

$$\frac{\partial}{\partial x}\left(k\frac{\partial T}{\partial x}\right)+\frac{\partial}{\partial y}\left(k\frac{\partial T}{\partial y}\right)+q=0 \quad (1) \tag{1}$$

Kjer $k$ predstavlja temperaturno neodvisno toplotno prevodnost.

Za rešitev te diferencialne enačbe smo morali na mejah našega definiranega območja (2D prerez) določiti robne pogoje, ki so lahko vključevali temperaturo $T$ v stopinjah Celzija, toplotni tok v $W/m^2$, in prestop toplote s toplotno prestopnostjo $h$ v $W/m^2K$ ter temperaturo okoljske tekočine $T_{ext}$ v stopinjah Celzija.

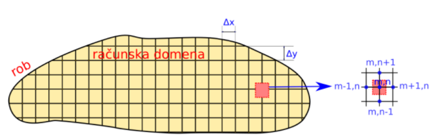
Pri stacionarnem dvodimenzionalnem prevodu toplote z zgoraj navedenimi pogoji smo dobili okrajšano enačbo:

$$\frac{\partial^2 T}{\partial x^2}+\frac{\partial^2 T}{\partial y^2}=0 \tag{2}$$

Pri reševanju našega problema, smo si pomagali z metodo končnih razlik.

## 1.2 Metoda končnih razlik

V inženirskih aplikacijah se pogosto srečujemo s kompleksnimi prerezi, kjer analitična rešitev diferencialne enačbe ni mogoča. V takih primerih se zatekamo k numeričnim metodam, kot so metoda končnih razlik (MKR), metoda končnih elementov (MKE), metoda končnih volumnov (MKV), itd. Pri tej projektni nalogi smo uporabili metodo končnih razlik.




Slika 1: Prikaz mreže

Metoda končnih razlik je numerična metoda, ki se uporablja za reševanje diferencialnih enačb. Diferencialno enačbo iskane funkcije v danem prostoru numerično rešujemo tako, da odvode funkcije aproksimiramo s kvocientom razlik. Pri izbrani mreži točk oziroma vozlišč v prostoru nas omenjen način privede do sistema (diferenčnih) enačb za funkcijske vrednosti v teh vozliščih.

Ker je rešitev po naši metodi dana v diskretnih točkah (vozliščih), moramo obravnavano območje (računsko domeno) popisati s strukturirano mrežo pravokotnikov (razvidnih v sliki 1). V robnih točkah mreže pa definiramo v naprej določene robne pogoje.

Iskane vrednosti v točkah na naši mreži se lahko iračuna po naslednjih enačbah, za drugi odvod v diskretnih točkah.

Za koordinato x:

$$\partial^2 T/\partial {x^2}_{m,n} \approx \frac{T_{m+1,n} - 2T_{m,n} + T_{m-1,n}}{\Delta x^2} \tag{3}$$

In za koordinato y:

$$\partial^2 T/\partial {y^2}_{m,n} \approx \frac{T_{m,n+1} - 2T_{m,n} + T_{m,n-1}}{\Delta y^2} \tag{4}$$

## 1.3 Robni pogoji

Vozlišča na robovih moramo definirati preko sledečih enačb.

- Prestop toplote na notranjem kotu:

$$2(T_{m-1,n}+T_{m,n+1})+(T_{m+1,n}+T_{m,n-1})+2h\Delta x\frac{T_{\text{ext}} - 2(3+h\Delta x)}{k}T_{m,n} = 0 \tag{5}$$

- Prestop toplote na robu

$$2(T_{m-1,n} + T_{m,n+1} + T_{m,n-1}) + \frac{2h\Delta x}{k}(T_{\text{ext}} - 2(h\Delta x + 2))T_{m,n} = 0 \tag{6}$$

- Prestop toplote na zunanjem kotu

$$(T_{m,n-1} + T_{m-1,n}) + \frac{2h\Delta x}{k}(T_{\text{ext}} - 2(h\Delta x + 1))T_{m,n} = 0 \tag{7}$$

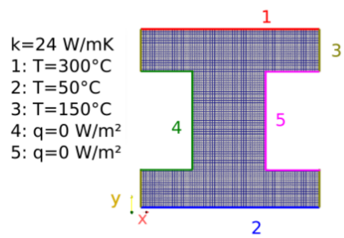
- Toplotni tok na robu

$$(2T_{m-1,n} + T_{m,n-1} + T_{m,n+1}) + \frac{2q\Delta x}{k} - 4T_{m,n} = 0 \tag{8}$$

- Vozlišče v notranjosti

$$T_{m-1,n} + T_{m+1,n} + T_{m,n-1} + T_{m,n+1} - 4T_{m,n} = 0 \tag{9}$$

Vsak par je dobil različne robne pogoje. Midva sva dobila 4. tip. Na treh robovih sva dobila definirano temperaturo, na dveh pa definiran toplotni tok.




Slika 2: Robni pogoji

### 1.4 Rešitev sistema enačb

Enačb dobimo toliko, kot imamo vozlišč. Ko sestavimo enačbe za vsako vozlišče dobimo sistem linearno neodvisnih enačb:

$$\begin{aligned}
a_{11}T_1 + a_{12}T_2 + a_{13}T_3 + \ldots + a_{1N}T_N &= C_1 \\
a_{21}T_1 + a_{22}T_2 + a_{23}T_3 + \ldots + a_{2N}T_N &= C_2 \\
a_{31}T_1 + a_{32}T_2 + a_{33}T_3 + \ldots + a_{3N}T_N &= C_3 \\
&\vdots \\
a_{N1}T_1 + a_{N2}T_2 + a_{N3}T_3 + \ldots + a_{NN}T_N &= C_N
\end{aligned}$$

Oziroma v matrični obliki: [A][T] = [b],

kjer A predstavlja koeficientno matriko sistema, T vektor neznank in b vektor konstant.

## 2 Računalniški program

Ko sva razumela fizikalni in matematični model najnega problema, sva se lotila pisati računalniškega programa, s katerim sva kasneje prišla do rešitve. Računalniški program sva pisala v jeziku C++. Najprej sva v program vključila vse knjižnjice, ki sva jih potrebivala med pisanjem programa. Knjižnjice so služile za branje podatkov iz datotek, za obdelavo vektorjev, nizanje string-ov, uporabo že definiranih algoritmov, pretvorbo podatkov za uporabu v drugih računalniških okoljih, merjenje časa izvajanaja programa, itd.

### 2.1 Branje podatkov iz datoteke

V temu dele kode sva najprej inicializirala sezname za shranjevanje geometrijskih in robnih podatkov. Te seznami vklučujejo koordinate točk (X in Y), vozlišča celic, tipe robnih pogojev in njihove vrednosti ter seznam vozlišč, kjer veljajo robni pogoji.

Nadalje program bere geometrijske podatke iz datoteke ”primer4mreza.txt”, vključno s koordinatami točk in vozlišči celic. Nato program prebere robne pogoje, kakšen tip je robni pogoj (temperatura, toplotni tok ali prestop toplote), in na koncu še vrednosti robnih pogojev.

Celoten postopek se ponavlja za celotno datoteko ”primer4mreza.txt”, na koncu pa datoteko zapre.

Nato sva za vsako vozlišče v mreži, ustvarila vektor sosednjih vozlišč označen sosedje-i. Ki se iterira skozi celice in preveri, ali je vozlišče (i) eno od vozlišč v trenutni celici. Če je, se za vsako sosednje vozlišče v tej celici določi pozicija glede na obravnavano vozlišče (i) ter se nato to sosednje vozlišče shrani v ustrezen del vektorja sosedje-i.

Kot končni rezultat sva dobila seznam sosednjih vozlišč za vsako vozlišče v mreži, ki sva ga shranila v vektorju sosednja-vozlisca.

## 2.2 Sestavljanje matrike A in vektorja b

V temu delu računalniškega programa sva se ukvarjala s konstrukcijo sistema linearnih enačb, ki predstavlja diskretizacijo našega problema prenosa toplote. To sva naredila z matriko koeficientov A in vektorjem konstant b.

Začela sva z inicializacijo spremenljivk. Ustvarila sva ničelno matriko A in ničelni vektor b. Število elementov v A in b, pa sva definirala z korakoma diskretizacije prostota ”deltaX”(korak v X osi) in ”deltaY”(korak v Y osi). Kot konstanto v enačbah, pa sva določila ”k”.

Nato sva izvedla interacijo skozi vsa vozlišča mreže. Tukaj program za vsako vozlišče pridobi indeks njegovih sosednjih vozlišč. Nato za ta vozlišča preveri ali so notranja ali robna (na robu mreže vozlišč).

Če je vozlišče bilo notranje, se je v matriko A vstavil koeficient glede na diskritizacijsko shemo za prenos toplote. Če pa je vozlišče bilo na robu, je program preveril različne tipe robnih pogojev (temperaturo, toplotni tok, prevod toplote). Če je bil robni pogoj znana temperatura, se je v matriko A zapisala vrednost ena na diagonalnem mestu, pripadajoči element v vektorju b pa je dobil vrednost temperature. Če je bil robni pogoj znan toplotni tok, se je preverilo število sosednjih vozlišč. Če je bilo število sosedov enako tri, sta se matrika A in vektor b dodatno obdelala. Podobno je veljalo za robni pogoj prestop toplote, kjer se je prav tako preverilo število sosednjih vozlišč in ustrezno obdelalo matriko A in vektor b. Tako sva dobila matriko A in vektor b in s tem sistem linearnih enačb, ki ga je mogoče rešiti.

## 2.3 Reševanje sistema linearnih enačb

Sedaj ko sva dobila sistem linearno neodvisnih linearnih enačb, sva lahko začela z numeričnim računanjem rešitve. Program je uporabil iterativno reševanje sistema linearnih enačb s interacijo. Temu postopku pravimo Jacobijeva metoda.

Metoda deluje na sledeč način: Začnemo z začetnimi vrednostmi temperature (T) na vsakem vozlišču mreže. Te vrednosti so se v našem primeru inicializirale na 100,0. Nato za vsako vozlišče na mreži in v vsaki interaciji izračunamo novo vrednost temperature (T[jj]) na trenutnem vozlišču, pri čemer upoštevamo vrednosti sosednjih vozlišč, za kar uporabimo Jacobijevo iterativno formulo:

$$T_{jj}^{(k+1)} = \frac{1}{A_{jj}} \left( b_{jj} - \sum_{i \neq j} A_{jj} \cdot T_i^{(k)} \right) \tag{10}$$

Postopek se ponavlja za vsa vozlišča in se izvede večkrat (v našem primeru se je 1000krat).

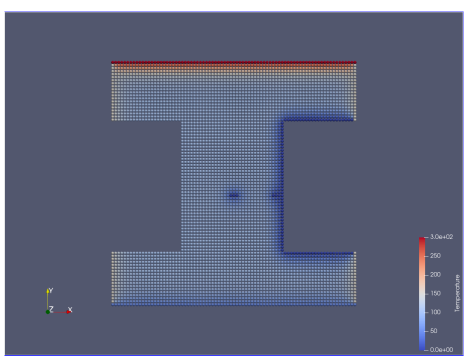
Po končani interaciji je program izpisal končne vrednosti temperature na vsakem vozlišču v naši mreži. Končne vrednosti temperature in informacije o mreži so se izvozile v VTK format ("Visualization Toolkit"), ki se je shranjeval v datoteki "resitev4mreza.vtk". To nama je omogočilo vizualizacijo rezulataov.

# 3 Rezultati

Matriko A in vektor b, sva prenesla še v računalniški jezik Mathematica, z funkcijo "JSON". V Mathematici pa sva sistem linearni enačb še enkrat rešila z funkcijo "linsolve", da sva lahko primerjala rešitve.

V C++ je program potreboval približno 612 sekund za izračun matrike temperatur po najnem prerezu, v Mathematici pa je program potreboval le 64 sekund. Ugotovila sva da je Program v Mathematici potreboval za izračun skoraj desetkrat manj časa, zaradi manjšega števila interacij, kot v programu C++. K hitrejšem izračunu, pa je prispevalo tudi to, da sva v Mathematico naložila že ustvarjeno matriko A in vektor b, v C++ pa jih je program moral ustvariti sam.

Za vizualizacijo temperature po prerezu sva uporabila program ParaView. Vanj sva vnesla vmesnik VTK, program pa je izrisal temperaturo, kot je prikazano na sliki 3.



Slika 3: VTK-Vizualizacija rešitve