Napredna računalniška orodja, projektna naloga
Izračun porazdelitve temperature v 2D prerezih

Jure Zupančič, 23211246,

Žiga Kemperle, 23211184

Fakulteta za strojništvo, Univerza v Ljubljani

Ljubljana, januar 2024

1 Teorija

V projektni nalogi je obravnavan primer časovno neodvisnega prenosa toplote v 2D prerezu. Za dane robne pogoje smo imeli za nalogo izračunati porazdelitev temperatur.

Upoštevane so bile naslednje predpostavke: temperaturno neodvisna toplotna prevodnost in robni pogoji, dvodimenzionalni prevod toplote (x, y), predpostavili smo stacionaren prevod toplote in zanemarili smo notranjo generacijo toplote.

Enačba spodaj predstavlja stacionaren dvodimenzionalen prevod toplote.

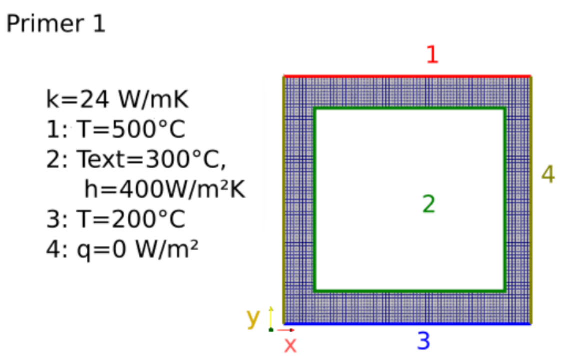
$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} = 0$

Zaradi kompleksnosti primera smo za izračun porazdelitve temperatur uporabili numerično metodo MKR - metodo končnih razlik. MKR je numerična metoda, ki služi numeričnemu reševanju diferencialnih enačb tako, da upošteva za aproksimacijo vozlišča sosednja vozlišča, ki se nahajajo levo, desno, zgoraj in spodaj, ter nato odvode funkcije aproksimira s kvocientom razlik- diferenčna shema . Mreža je sestavljena iz celic, vsaka ima štiri vozlišča. Za izbrani primer za mrežo vozlišč v prostoru dobimo sistem diskrezitiranih diferencialnih (diferenčnih) enačb za funkcijske vrednosti v teh vozliščih.

$$\left.\frac{\partial^2 T}{\partial x^2}\right|_{m,n} \approx \frac{\partial T/\partial x|_{m+1/2,n} - \partial T/\partial x|_{m-1/2,n}}{\Delta x}$$

Slika 1: Drugi odvod, aproksimacija za MKR

Za 2D presek je definirana kvadratna mreža, ki je podana v vhodni datoteki “primer1mreza.txt”. Struktura mreže zajema število celic, vozlišč in podaja 4 robne pogoje.



Slika 2: Naš primer 1 za mrežo in robne pogoje

Za reševanje sistema moramo upoštevati robne pogoje. Lahko so v obliki prestopa toplote na notranjem kotu, na robu ali na zunanjem kotu, in toplotni tok na robu ter temperatura na robu ali v notranjosti vozlišča. V sistemu imamo toliko enačb, kot je vseh vozlišč. Vse pridobljene enačbe lahko zapišemo v matrično obliko. Za reševanje sistema enačb smo uporabili metodo Gauss-Seidel. Rešitev sistema smo dobili po iteracijskem reševanju v obliki vektorja temperaturne porazdelitve.

2 Delovanje programa

V programu je potrebno branje in obdelava informacij iz datoteke, ki opisuje mrežo. Cilj je pridobiti podatke o vozliščih, celicah ter robnih pogojih, ter jih organizirati v ustrezne strukture za nadaljnje numerično reševanje sistema linearnih enačb, ki izhajajo iz problema prenosa toplote.

2.1 Branje podatkov iz datoteke

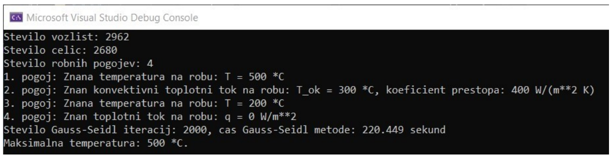
Koda v C++ deluje tako da najprej prebere vhodno datoteko, naloži podatke o vozliščih, celicah in robnih pogojih. Podatki se berejo vrstico po vrstico, in sicer število vozlišč, koordinate vozlišč, sosednost vozlišč število celic, informacije celic, število robnih pogojev, tipi robnih pogojev, vrednosti in povezave vozlišč z robnimi pogoji. Podatki so nato zapisani z aproksimativnimi vrednostmi – MKR.

2.2 Izgradnja sistema enačb

Po končanem branju datoteke so vsi potrebni podatki shranjeni v različne strukture. Na primer, koordinate vozlišč so v vektorjih X in Y, informacije o celicah v matriki celic, podatki o robnih pogojih pa so razdeljeni v več vektorjev in matrik. Na podlagi geometrije mreže in definiranih robnih pogojev sestavi matriko sistema linearnih enačb (A) in vektor prostih členov (b). Slednja sta shranjena v datoteki "A.csv"in "b.csvža nadaljnjo analizo v programu MATLAB.

2.3 Gauss-Seidelova metoda in rešitev

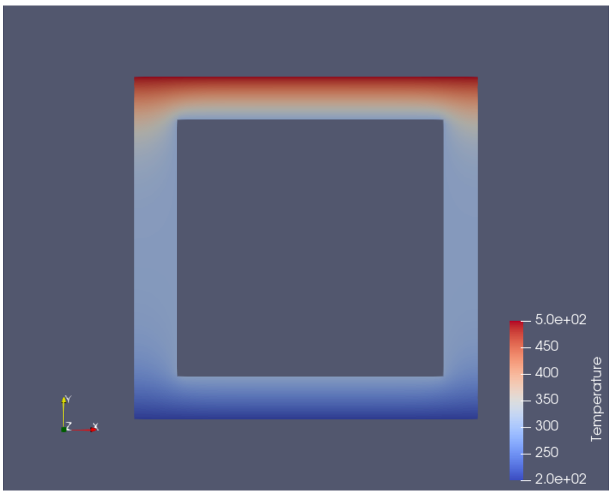
V C++ za relativno enostaven primer sistema enačb, kjer je večina elementov blizu diagonale in imamo redko matriko (sparse matrix) se reševanje izvršuje z Gauss-Seidelovo metodo za iterativno reševanje sistema linearnih enačb. Iteracije potekajo v več nitih (uporaba OpenMP). Najprej inicializiramo vektor rešitev na začetno vrednost . Kot začetno za temperaturne vrednosti je 100 °C za vsako vozlišče. Po končani iterativni metodi program izračuna rešitev temperatur v obliki vektorja temperatur T in lahko določimo maksimalno temperaturo v mreži.



Slika 3: Rezultat reševanja in hitrost kode

2.4 Shranjevanje rezultatov v VTK format in prikaz v okolju Paraview

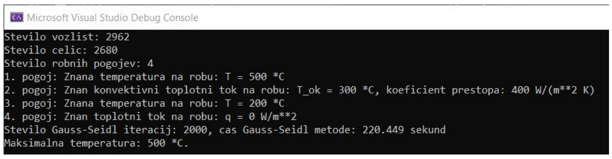
S pomočjo knjižnice, ki avtomatično shrani mreže in rezultate v datoteki VTK. Nato smo datoteko uporabili za vizualizacijo rezultatov v orodju Paraview. Na sliki je prikazana simulacija temperaturne razporeditve za primer številka 1.



Slika 4: Prikaz rezultatov temperaturne razporeditve v orodju Paraview

2.5 Primerjava časov računanja

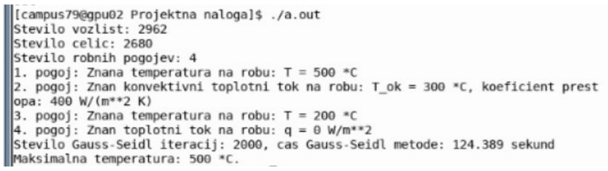
V C++ kodi smo merili čas Gauss-Seidel metode in število iteracij (na osebnem računalniku PC).



```
Microsoft Visual Studio Debug Console
Stevilo vozlist: 2962
Stevilo celic: 2680
Stevilo robnih pogojev: 4
1. pogoj: Znana temperatura na robu: T = 500 *C
2. pogoj: Znan konvektivni toplotni tok na robu: T_ok = 300 *C, koeficient prestopa: 400 W/(m**2 K)
3. pogoj: Znana temperatura na robu: T = 200 *C
4. pogoj: Znan toplotni tok na robu: q = 0 W/m**2
Stevilo Gauss-Seidl iteracij: 2000, cas Gauss-Seidl metode: 220.449 sekund
Maksimalna temperatura: 500 *C.
```

Slika 5: Hitrost kode C++ na PC

Na spodnji sliki je prikazana še primerjava hitrosti reševanja C++ kode s pomočjo HPC s paralelizacijo.

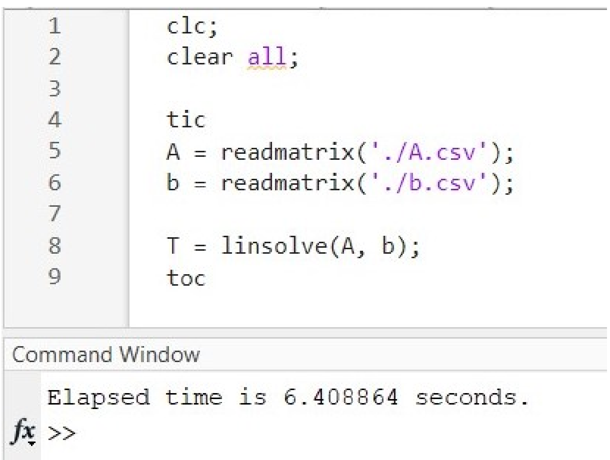


```
[campus79@gpu02 Projektna naloga]$ ./a.out
Stevilo vozlist: 2962
Stevilo celic: 2680
Stevilo robnih pogojev: 4
1. pogoj: Znana temperatura na robu: T = 500 *C
2. pogoj: Znan konvektivni toplotni tok na robu: T_ok = 300 *C, koeficient prest
opa: 400 W/(m**2 K)
3. pogoj: Znana temperatura na robu: T = 200 *C
4. pogoj: Znan toplotni tok na robu: q = 0 W/m**2
Stevilo Gauss-Seidl iteracij: 2000, cas Gauss-Seidl metode: 124.389 sekund
Maksimalna temperatura: 500 *C.
```

Slika 6: Hitrost kode C++ s HPC

Izkaže se, da je izvajanje algoritmov kot pričakovano hitreje s HPC.

V kodi C++ smo shranili matriko A in vektor b v datoteko, ki jo lahko prebere program MATLAB. Nato smo v MATLAB-u pokazali reševanje sistema enačb in primerjali hitrost reševanja. Koda uporablja funkcijo linsolve, ki reši sistem linearnih enačb mnogo hitreje kot naša koda v C++.



```
clc;
clear all;

tic
A = readmatrix('./A.csv');
b = readmatrix('./b.csv');

T = linsolve(A, b);
toc
```

```
Command Window
Elapsed time is 6.408864 seconds.
fx >>
```

Slika 7: Hitrost kode MATLAB

3 Zaključek, ugotovitve

V projektni nalogi smo obravnavali prevod toplote in z numerično metodo v programskem jeziku C++ izračunali temperaturno porazdelitev za naš primer. Simulacijo smo prikazali v okolju Paraview ter vizualno ugotovili, kako se toplota prenaša. Nato smo primerjali hitrosti reševanja sistema enačb s C++ in MATLAB. Ugotovili smo, da je MATLAB veliko hitrejši od C++ (6 sekund proti 200 sekund).