Slika 1: FAKULTETA ZA STROJNIŠTVO

NROR Projektna naloga

Anže Zrimšek Žiger, Alen Fočić

January 2024

TABLE OF CONTENTS

Kazalo

1 Uvod

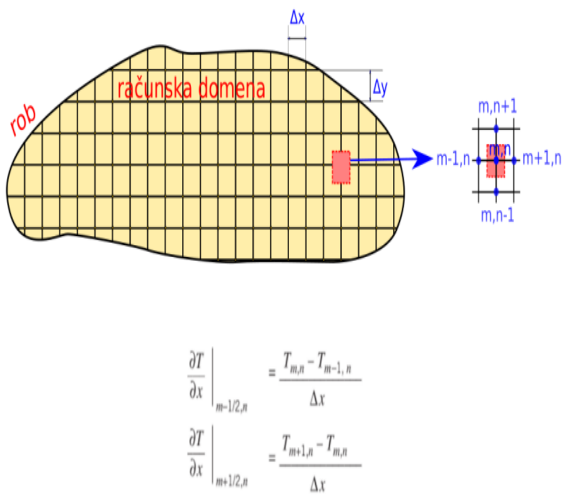
Pri tem projektu, sva s pomočjo programskega jezika c++ in programom paraview prikazala prenos toplote skozi 2D prerez. Splošna diferencialna enačba prenosa toplote je difinirana

$$\frac{\partial}{\partial x}(k\frac{\partial T}{\partial x}) + \frac{\partial}{\partial y}(k\frac{\partial T}{\partial y}) + \frac{\partial}{\partial z}(k\frac{\partial T}{\partial z}) + q = 0.$$

Ker pa vemo, da je prevod stacionaren in le v 2D prostoru, lahko zgornjo enačbo preuredimo in dobimo sledečo eačbo:

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

Tega primera, smo se lotili s pomočjo metode končnih razlik. Metoda sicer ni eksaktna rešitev problema, vendar ob dovolj gosti mreži, pravilno podanih robnih pogojev in pogojev prehoda, poda dovolj dobro rešitev. Slednja se lahko zelo približa eksaktni. Napaka se pojavi pri pretvorbi v diferenčni operator, kjer odvode popišemo s pomočjo sosednih točk. Pri tem, si pa pomagamo s Taylorjevo vrsto.



Slika 2: Prikaz mreže

Kot je napisano že zgoraj, odvode aproksimiramo z diferenčnim operatorjem, kot lahko vidimo v spodnjih enačbah.

$$\frac{\partial^2 T}{\partial x^2} = \frac{\triangle T_{m+1,n} - 2\triangle T_{m,n} + \triangle T_{m-1,n}}{(\triangle x)^2}$$

$$\frac{\partial^2 T}{\partial y^2} = \frac{\triangle T_{m+1,n} - 2\triangle T_{m,n} + \triangle T_{m-1,n}}{(\triangle y)^2}$$

2 Opis Projekta

Na začetku imamo 4 različne oblike modelov, kar pomeni, da bo porazdelitev temperature pri vsakem drugačna. Ko smo enkrat določili pogoje prehoda in robne pogoje, jih vstavimo v diferencialno enačbo. Če želimo obravnavan objekt čim bolje popisati, moramo mrežo kar precej zgostit, kar pa posledično pomeni, da imamo večje število neznank. Za enoličen popis, moramo imeti toliko enačb, kot je neznank.
Zgeneriramo matriko koeficientov A, vektor zanih vrednosti b in vektor neznank T. Neznane vrednosti le še izračunamo s pomočjo sistema enačb. Ko sistem enačb uredimo in komponente spravimo v pripadajoč vektor oziroma matriko, dobimo sledečo obliko.

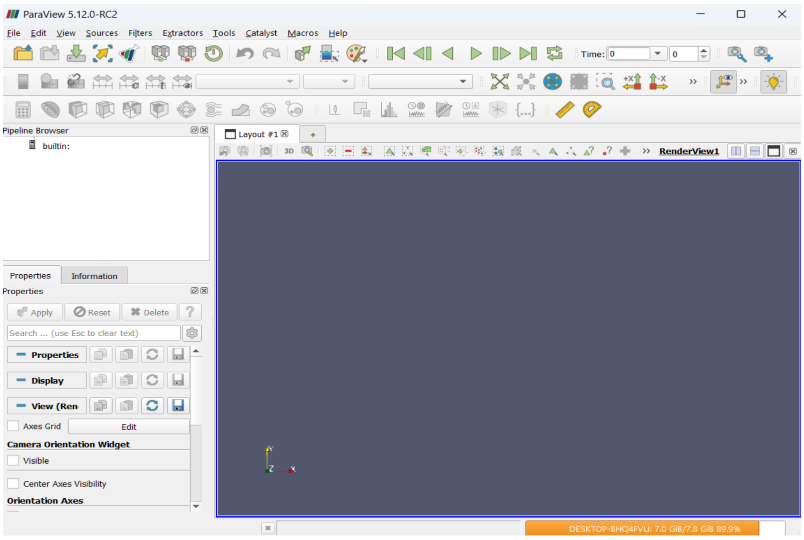
$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots \\ a_{21} & a_{22} & a_{23} & \cdots \\ a_{31} & a_{32} & a_{33} & \cdots \\ \cdots & \cdots & \cdots & \cdots \end{bmatrix} \begin{bmatrix} T_{11} \\ T_{12} \\ T_{13} \\ \cdots \end{bmatrix} = \begin{bmatrix} b_{11} \\ b_{12} \\ b_{13} \\ \cdots \end{bmatrix}$$

3 Struktura kode

Na začetku kode, je definirana funkcija, ki, enostavno povedano, v sami mreži išče določen element in na koncu poda njegove koordinate v obliki vektorja kot [x, y].
V nadeljevanju, koda odpre podano datoteko, v najinem primeru, primer2mreza.txt. Nato prične z branjem in analiziranjem datoteke. Ko jo dokončno analizira, pa začne isakti dane robne in pogoje prestopa, če so možni za ta primer. Če na primer določenega pogoja ne moremo pripisat določenemu primeru, ga koda "ignorira", se premakne dalje in preveri ali je nasljedni pogoj mogoče izvest. V ta nemen v kodi ležeijo pogoji kot sta if in else if.
Pred tem, pa koda določi velikost mreže. To stori tako, da podano datoteko, v najinem primeru primer2mreze.txt, prebere, ustvari dva nova seznama za x in y koordinate, in nato na podlagi teh dveh seznamov odčita samo velikost in obliko mreže.
V Nadeljevanju, se generirajo pripadajoča matrika in vektroji. To se napravi na podlagi tega koliko vozlišč je na voljo in glede na njihovo pozicijo. Na koncu le še določimo željeno število iteracij in korak ter izračunamo sistem enačb. Poleg tega, nam pa še poda čas reševanja sistema enačb. To smo naredili z namenom, da lahko primerjamo s časom reševanja v mathematici in da vidimo kater način je v tem primeru bolj učinkovit glede hitrosti reševanja.

4 Prikaz rešitev

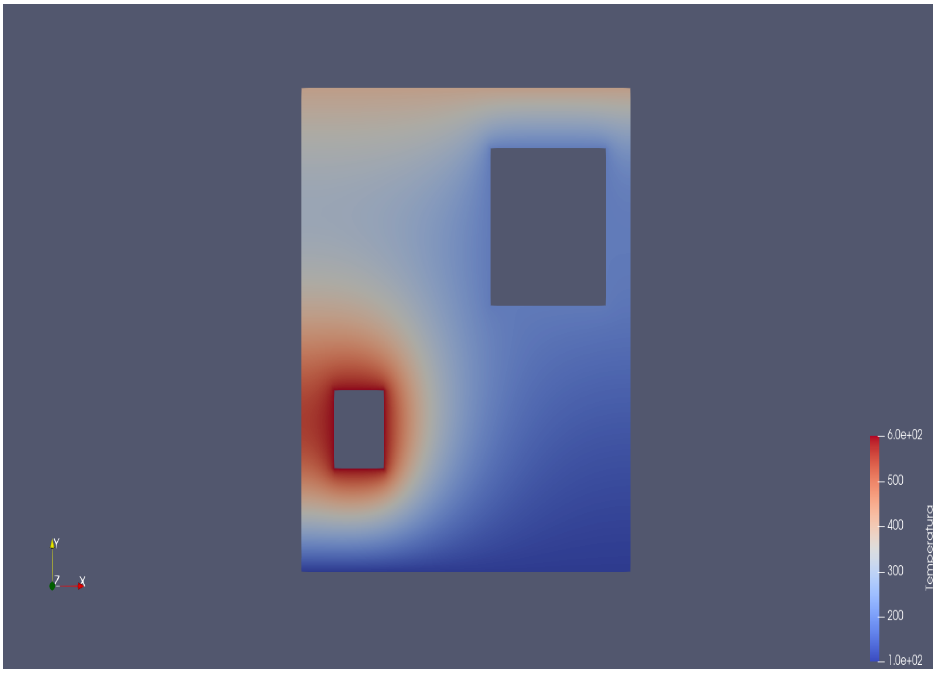
Na koncu pa le še želenim vozliščem pripišemo izračunane temperature in jih prikažemo. To storimo s pomočjo programa paraview. Sicer preden to lahko stroimo, moramo dobljeno rešitev shranit v novo datoteko ter jo nato pretvorit v vtk file z namenom, da jo lahko program prebere in nato tudi prikaže.



Slika 3: Prikaz paraview pograma

5 Zaključek

Kot nam prikazuje spodnja slika iz programa paraview, se maksimalna temperatura nahaja v levem spodnjem kvadratku, če gledamo obravnavani primer 2. Prav tako lahko opazimo, da se temperatura z oddaljevanjem od samega kvadratka, zmanjšuje in to kar znatno.



Slika 4: Prikaz rešitve v paraview