

SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE
STAVEBNÁ FAKULTA

Evidenčné číslo: SvF-16645-110057

METÓDA KONEČNÝCH PRVKOV
V PROGRAME FREEFEM

Bakalárska práca

2023

Patrícia Lukořková

SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE
STAVEBNÁ FAKULTA

Evidenčné číslo: SvF-16645-110057

METÓDA KONEČNÝCH PRVKOV
V PROGRAME FREEFEM
Bakalárska práca

Študijný program: matematicko-počítačové modelovanie
Študijný odbor: matematika
Školiace pracovisko: Katedra matematiky a deskriptívnej geometrie
Vedúci záverečnej práce: Ing. Marek Macák, PhD.

Bratislava 2023

Patrícia Lukoňková



ZADANIE BAKALÁRSKEJ PRÁCE

Študentka: **Patrícia Lukot'ková**
ID študenta: 110057
Študijný program: matematicko-počítačové modelovanie
Študijný odbor: matematika
Vedúci práce: Ing. Marek Macák, PhD.
Vedúci pracoviska: Ing. Marek Macák, PhD.

Názov práce: **Metóda konečných prvkov v programe FreeFEM**

Jazyk, v ktorom sa práca vypracuje: slovenský jazyk

Špecifikácia zadania:

Úlohou študentky je osvojiť si prácu s balíkom FreeFEM. Pre zvolenú geodetickú okrajovú úlohu spočítať 2D a 3D experimenty použitím balíka FreeFEM.

Rozsah práce: 30

Zoznam odbornej literatúry:

1. Reddy, J. *An Introduction to Nonlinear Finite Element Analysis*. New York : Oxford University Press, 2006. 463 s. ISBN 0-19-852529-X.
2. Macák, M. *Numerical Methods in Geodesy*. Dizertačná práca. Bratislava : 2015. 65 s.
3. Minarechová, Z. – Macák, M. – Čunderlík, R. – Mikula, K. On the finite element method for solving the oblique derivative boundary value problems and its application in local gravity field modelling. *Journal of geodesy*, 95. s. 2021.

Termín odovzdania bakalárskej práce: 04. 05. 2023

Dátum schválenia zadania bakalárskej práce: 22. 02. 2023

Zadanie bakalárskej práce schválil: prof. RNDr. Karol Mikula, DrSc. – garant študijného programu

Čestné prehlásenie

Prehlasujem, že som túto záverečnú prácu vypracovala samostatne pod vedením vedúceho záverečnej práce, s použitím literatúry uvedenej v zozname použitej literatúry.

Bratislava 4. 5. 2023

Patrícia Lukoťková

Pod'akovanie

Týmto chcem pod'akovať môjmu školiteľovi Ing. Marekovi Macákovi PhD. za veľmi cenné rady, pripomienky a odbornú pomoc počas vypracovania mojej práce.

Bratislava 4. 5. 2023

Patrícia Lukot'ková

Abstrakt

Názov práce: Metóda konečných prvkov v programe FreeFEM

Abstrakt: Práca sa zaoberá riešením diferenciálnej rovnice pomocou softvéru FreeFEM. Konkrétne Laplaceovou rovnicou pre poruchový potenciál na oblasti medzikružia respektíve medzi dvomi sférami. V prvej časti práce je predstavený samotný program FreeFEM a priblížené jeho používanie medzi čo patrí tvorba sietí v 2D a 3D, definovanie problému či výber elementov. Ďalej je v práci popísaný poruchový potenciál a následne vytvorenie geodetickej okrajovej úlohy. Nasleduje riešenie tejto úlohy pomocou FreeFEM pre rôzne oblasti. V poslednej časti je ukázané zobrazovanie výsledkov zo softvéru FreeFEM pomocou ParaView.

Kľúčové slová: FreeFEM, geodetická okrajová úloha, metóda konečných prvkov, ParaView, poruchový potenciál

Abstract

Title: Finite element method in the FreeFEM program

Abstract: The bachelor thesis deals with the solution of the differential equation using FreeFEM software. Specifically, the Laplace equation for the disturbing potential on the region between two concentric circles and the two spheres. In the first part of the thesis the FreeFEM software itself is introduced and its usage is presented, which includes the creation of 2D and 3D meshes, problem definition or element selection. Next, the thesis describes the disturbing potential and then the creation of a geodetic boundary value problem. It is followed by the solution of this problem using FreeFEM for different domains. In the last section, the display of the results from FreeFEM software using ParaView is shown.

Keywords: FreeFEM, geodetic boundary value problem, finite element method, ParaView, disturbing potential

Obsah

Úvod	8
1 Softvér FreeFEM	9
1.1 História	9
1.2 Inštalácia	10
1.2.1 Linux - Ubuntu	10
1.2.2 Windows	11
1.3 Práca s FreeFem	11
1.3.1 Syntax	11
2 Diferenciálna rovnica pre poruchový potenciál	22
2.1 Teória potenciálu	22
2.2 Geodetická okrajová úloha	23
2.3 Slabá formulácia	24
3 Experimenty	26
3.1 L2 norma chyby a EOC	26
3.2 Experimenty pre 2D oblasť	26
3.2.1 Výpočtová oblasť je medzikružie v prvom kvadrante	27
3.2.2 Výpočtová oblasť je celé medzikružie	30
3.3 Experimenty pre 3D oblasť	33
3.3.1 Výpočtová oblasť je medzi dvomi sférami	33
4 Vizualizácia pomocou ParaView	37
4.1 Vizualizácia riešenia na výseku medzikružia	37
4.2 Vizualizácia riešenia na celom medzikruží	38
4.3 Vizualizácia riešenia na oblasti medzi 2 sférami	38
Záver	39
Literatúra	40

Úvod

Mnoho problémov vo fyzike, inžinierstve či v bankovníctve je modelovaných práve diferenciálnymi rovnicami. Avšak tieto rovnice vedia byť natoľko zložité, že ich nemožno analyticky riešiť. Z toho vzišla potreba vyvinúť numerické metódy ktorými by sa vypočítalo približné riešenie s dostatočne malou chybou. Medzi takéto metódy patrí napríklad metóda konečných prvkov (MKP). [8]

MKP dosiahla prudký rozvoj začiatkom 60. rokov 20. storočia. Jej hlavnou myšlienkou je diskretizácia oblasti na viacero geometricky jednoduchších oblastí - konečných prvkov (elementov). Na každom konečnom prvku sa následne neznáma funkcia aproximuje lineárnou kombináciou elementových aproximačných funkcií a približných uzlových hodnôt neznámej funkcie, ktoré sa hľadajú tak, aby spĺňali slabú formuláciu. [10]

V dnešnej dobe sú numerické metódy implementované v rôznych softvéroch. Softvér FreeFEM využíva pre riešenie problémov práve metódu konečných prvkov. Jedná sa o voľne dostupný a stiahnuteľný softvér, vyvinutý profesorom Hechtom a kolektívom. Využíva riešenie založené na definovaní slabej formulácie, čo z neho robí silný softvér, pretože nie je nutné programovať jednotlivé kroky MKP, ktoré môžu byť veľakrát značne komplikované. [5]

S jedným z inžinierskych problémov sa možno stretnúť aj pri fyzikálnej geodézií. Poruchový potenciál spĺňa Laplaceovu rovnicu mimo oblasti Zeme a spolu s okrajovými podmienkami tvorí geodetickú okrajovú úlohu. Cieľom tejto práce je čitateľovi priblížiť prácu so softvérom FreeFEM, a práve túto okrajovú úlohu s jeho využitím vyriešiť rovnako aj posúdiť, či je softvér vhodný na riešenie geodetických úloh.

V prvej časti práce je predstavený samotný softvér FreeFEM. Čitateľ sa dozvie možnosti ako vytvoriť 2D či 3D sieť, ako definovať samotný problém a v neposlednom rade ako vykresliť alebo exportovať dáta pre iný softvér na vizualizáciu.

V druhej časti je priblížený pojem poruchového potenciálu, definovaná geodetická okrajová úloha ako aj zostavenie slabej formulácie pre túto úlohu, ktorá je potrebná na implementáciu vo FreeFEM.

Tretia časť sa zaoberá už samotnou implementáciu geodetickej okrajovej úlohy na rôznych oblastiach v 2D a 3D priestore.

V poslednej časti práce sú výsledky vypočítané pomocou softvéru FreeFEM exportované do ParaView pre vizuálne porovnanie vykresľovania so samotným softvérom.

Kapitola 1

Softvér FreeFEM

FreeFEM je softvér na numerické riešenie parciálnych diferenciálnych rovníc pomocou metódy konečných prvkov v 1D, 2D alebo 3D priestore.

Bol vyvinutý vo výskumnom inštitúte Laboratoire Jacques-Louis Lions na univerzite Université Pierre et Marie Curie v Paríži profesorom aplikovanej matematiky Frédéricom Hechtom a kolektívom. Medzi dôvody vytvorenia tohto softvéru patrí aj uľahčenie výučby metódy konečných prvkov pre študentov.

Ako samotný názov napovedá, jedná sa o bezplatný, voľne stiahnuteľný softvér, ktorý dokáže bežať na všetkých UNIX operačných systémoch s g++ 3.3 alebo novším (kompilátor pre C++ programy) a OpenGL (grafické aplikačné programové rozhranie), na operačnom systéme Windows XP, 7, 8, 10, 11 a na MacOS 10.

Jeho jazyk je podobný programovaciemu jazyku C++. FreeFEM má pokročilý automatický generátor sietí, ktorý je taktiež schopný dodatočnej adaptácie siete. Má na výber z viacerých konečných prvkov, vrátane Lagrangeových elementov.

Čo sa týka vizualizácie a pozorovania skúmaných problémov, FreeFEM poskytuje farebné vykresľovanie, ponúka možnosť rezania objektov, približovanie a mnoho ďalších funkcií ako aj export dát pre ďalšiu manipuláciu napríklad pomocou Paraview alebo v Matlabe. [5]

1.1 História

Hoci sa v dnešnej dobe jedná o voľne dostupný softvér pre okamžitú inštaláciu, jeho prvá verzia nebola zadarmo.

Prvá verzia bola vytvorená v roku 1987 a bola nazvaná MacFem. táto verzia fungovala len na počítači Macintosh. Verzia s názvom PCFem sa objavila o nejaký čas neskôr. Obe verzie boli napísané v Pascalí a ani jedna nebola zdarma.

V roku 1992 bol softvér prepísaný do jazyka C++ a pomenovaný FreeFEM. v priebehu nasledujúcich rokov pribúdali a stále pribúdajú nové elementy, siete a ďalšie funkcie. V rámci tejto práce je využitá najnovšia verzia 4.12 vydaná v roku 2022. [5]

1.2 Inštalácia

Pre používanie softvéru FreeFEM sú dostupné 2 inštalačné metódy:

- užívateľský prístup (binárny balík)
- prístup vývojárov (zo zdrojového kódu)

Softvér FreeFEM, ktorý bol využitý pre túto prácu bol nainštalovaný metódou s použitím binárneho balíka, táto metóda je dostupná pre operačné systémy Linux, MacOS a Windows.

V nasledujúcich krokoch sú popísané postupy pre úspešnú inštaláciu na operačný systém Windows a Linux, ktoré sú otestované na Windows 11 64x a Linux s distribúciou Ubuntu 20.4. Pre operačný systém MacOS, pre ostatné distribúcie Linuxu ako aj inštaláciu zo zdrojového kódu je dostupný návod v oficiálnej dokumentácii k softvéru FreeFEM [5].

1.2.1 Linux - Ubuntu

V súvislosti s inštaláciou softvéru na distribúciu Ubuntu je potrebné stiahnutie balíčka *.deb z [5]. Pre aktuálne najnovšiu verziu FreeFEM 4.12 má tento súbor pre Ubuntu 20.4 presný názov FreeFEM-4.12-amd64-ubuntu20.04.deb. Po stiahnutí nasledujú 4 kroky inštalácie pre úspešné používanie softvéru.

1. Pred samotnou inštaláciou programu je potrebné si nainštalovať nižšie uvedené balíky pomocou nasledujúcich príkazov:
 - `sudo apt-get install libgsl-dev libhdf5-dev`
 - `sudo apt-get install liblapack-dev libopenmpi-dev freeglut3-dev`
2. Následne je nutné stiahnuť a nainštalovať aktualizácie pre všetky neaktuálne baličky vo vašom systéme pomocou príkazov:
 - `sudo apt-get update`
 - `sudo apt-get upgrade`
3. Teraz sa môže nainštalovať FreeFEM zo stiahnutého súboru *.deb pomocou príkazu:
 - `sudo dpkg -i FreeFEM-4.12-amd64-ubuntu20.04.deb`
4. Po inštalácii je softvér dostupný priamo z príkazového riadku príkazom:
 - `FreeFem++ nazov_suboru.edp`

1.2.2 Windows

Inštalácia softvéru FreeFEM pre operačný systém Windows začína podobne ako vyššie uvedená inštalácia pre Ubuntu. Je nutné si stiahnuť príslušný balíček *.exe opäť z oficiálnej stránky softvéru [5]. Pre Windows 11, 64 bitovú verziu je aktuálne k verzii FreeFEM 4.12 celý názov súboru `FreeFem++-4.12-win11-x64.exe`.

Kroky k úspešnému chodu softvéru sú nasledovné:

1. Samotná inštalácia programu, ako je pre Windows bežné, sa spustí dvojitým kliknutím na stiahnutý súbor `FreeFem++-4.12-win11-x64.exe`
2. Po úspešnej inštalácii sa na pracovnej ploche vytvorí odkaz na softvér FreeFEM, pomocou ktorého sa dajú spúšťať programy. Taktiež je možné využívať príkazový riadok a spúšťať programy rovnakým príkazom ako na Ubuntu:

- `FreeFem++ nazov_saboru.edp`

1.3 Práca s FreeFem

Hlavnou myšlienkou práce s FreeFEM je prepis diferenciálnej rovnice do slabej formulácie. Následne je potrebné napísať kód v jazyku FreeFEM pomocou textového editora, tento súbor musí byť uložený vo formáte `nazov_saboru.edp`, potom stačí spustiť program príkazom `FreeFem++`.

1.3.1 Syntax

Pri využívaní softvéru je nutné poznať jeho syntax. V tejto podkapitole je čitateľovi syntax priblížená a sú predstavené základné funkcie, premenné a operátory pre následné využitie v riešení parciálnych diferenciálnych rovníc. Pre vysvetlenie používania FreeFEM v tejto kapitole sú informácie čerpané primárne z oficiálnej dokumentácie [5].

K vytvoreniu kódu v textovom editore je nutné každú premennú deklarovať konkrétnym typom, taktiež každý výraz je na konci oddelený bodkočiarkou. Pridávanie komentárov je podobné ako v jazyku C++ použitím dvoch lomiek: `// komentar`.

Často využívaná je premenná `verbosity(arg)`, ktorá reguluje množstvo vypísaných informácií po spustení programu. Prednastavená hodnota argumentu je 2, ďalšie možnosti sú 0 - žiadny výpis, 1 - stručný výpis, 10 - obsiahnejší výpis... . Táto premenná sa zadáva na začiatok kódu avšak regulovanie množstvo vypísaných informácií je možné aj pri samotnom spúšťaní programu pomocou prepínača `-v arg` v príkazovom riadku nasledovne: `FreeFem++ nazov_saboru.edp -v arg`.

Derivácie:

Softvér FreeFEM disponuje vstavanými funkciami pre výpočet derivácií, ktoré vyzerajú nasledovne

- 1. derivácie: $\mathbf{dx}() = \frac{\partial}{\partial x}()$, $\mathbf{dy}() = \frac{\partial}{\partial y}()$, $\mathbf{dz}() = \frac{\partial}{\partial z}()$,
- 2. derivácie: $\mathbf{dxx}() = \frac{\partial^2}{\partial x^2}()$, $\mathbf{dyy}() = \frac{\partial^2}{\partial y^2}()$, $\mathbf{dzz}() = \frac{\partial^2}{\partial z^2}()$,
- 2. zmiešané derivácie: $\mathbf{dxy}() = \frac{\partial^2}{\partial x \partial y}()$, $\mathbf{dxz}() = \frac{\partial^2}{\partial x \partial z}()$, $\mathbf{dyx}() = \frac{\partial^2}{\partial y \partial x}()$,
 $\mathbf{dyz}() = \frac{\partial^2}{\partial y \partial z}()$, $\mathbf{dzx}() = \frac{\partial^2}{\partial z \partial x}()$, $\mathbf{dzy}() = \frac{\partial^2}{\partial z \partial y}()$.

Integrály:

Keďže FreeFEM pracuje so slabou formuláciou diferenciálnej rovnice, ktorá pozostáva z integrálov, softvér má dostupné funkcie na výpočet integrálu cez 1D, 2D, 3D oblasti:

- 1D: $\mathbf{int1d}(M, \mathbf{label})() = \int_{\Omega} f() dx$, kde $\Omega \in \mathbb{R}$,
- 2D: $\mathbf{int2d}(M, \mathbf{label})() = \int_{\Omega} f() dx dy$, kde $\Omega \in \mathbb{R}^2$,
- 3D: $\mathbf{int3d}(M, \mathbf{label})() = \int_{\Omega} f() dx dy dz$, kde $\Omega \in \mathbb{R}^3$.

Kde M označuje sieť vytvorenú cez danú oblasť, a \mathbf{label} je nepovinný argument typu `int`, ktorý sa zadáva, ak má byť integrál vypočítaný len na určitej hranici siete M , s označením \mathbf{label} . Definovanie siete je konkrétne popísané v nasledujúcej časti.

Sieť:

Ďalšia súčasť riešenia metódou konečných prvkov je výpočtová oblasť reprezentovaná sieťou geometricky jednoduchých oblastí, ktorým sa hovorí konečné prvky (elementy). Na jej vytvorenie v softvéri FreeFEM je nutné použiť niektorú z nasledujúcich premenných:

- 1D: `meshL`
- 2D: `mesh`
- 3D: `mesh3`
- siete na hranici v 3D:
 - `meshS`
 - `meshL`

Sieť cez oblasť v 2D priestore v softvéri FreeFEM pozostáva z trojuholníkových elementov, ktorú je jednoducho možné vytvoriť pomocou funkcie `buildmesh()`, ktorá vracia premennú, sieť, typu `mesh`. Do tejto funkcie vstupujú ako argumenty jednotlivé hrany oblasti oddelené znamienkom `+`.

Hrany sa definujú parametrickým vyjadrením pomocou premennej `border`, dôležité je dodržať rovnaký smer všetkých hrán inak kompilátor zobrazí chybové hlásenie. Pri zadávaní hrán do funkcie `buildmesh()` je nutné definovať veľkosť siete, teda počet dielikov na koľko sa má daná hrana rozdeliť. V prípade potreby vytvorenia oblasti obsahujúcej dieru je potrebné k veľkosti siete doplniť aj záporné znamienko.

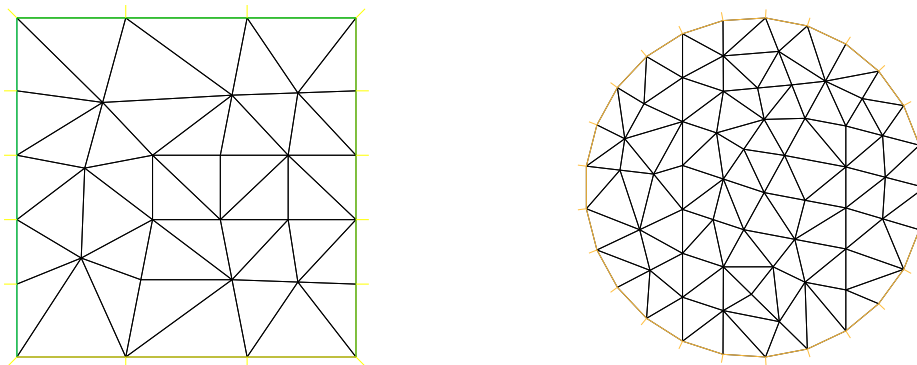
Na obrázku 1.1 sú zobrazené siete vytvorené funkciou `buildmesh()`. Sieť na obrázku 1.1 vľavo je definovaná na štvorci $[0, 1] \times [0, 1]$, zostrojená nasledujúcimi príkazmi:

```
int nx=3; //pocet dielikov v smere osi x
int ny=5; //pocet dielikov v smere osi y
// parametricke definovanie hranic oblasti:
border b1(t=0., 1.){x=1*t; y=0.; label=1;}; //spodna
border b2(t=0., 1.){x=1; y=1*t; label=2;}; //prava
border b3(t=0., 1.){x=1-1*t; y=1; label=3;}; //horna
border b4(t=0., 1.){x=0.; y=1-1*t; label=4;}; //lava
// vytvorenie siete:
mesh m = buildmesh(b1(nx) + b2(ny) + b3(nx) + b4(ny));
```

Nasledujúcim príkladom definovania siete funkciou `buildmesh()` je kruhová oblasť. Na obrázku 1.1 napravo je definovaná sieť na kruhovej oblasti s polomerom $r=1$ a stredom v bode so súradnicami $[0, 0]$ nasledujúcim príkazom:

```
real r=1.; //polomer
int n=25; //pocet dielikov na hranici
// parametricke definovanie hranice oblasti:
border circle(t=0, 2*pi) {x= r * cos(t); y= r * sin(t); label=1;};
// vytvorenie siete:
mesh m = buildmesh(circle(n));
```

Označenie `label` pri definovaní hrán oblasti nie je povinný argument, ak nie je uvedený, je prednastavený na hodnotu 1. V týchto príkladoch označenie `label` nie je nutné, no v prípade potreby integrovania cez niektorú z hrán siete `m` sa do funkcie integrálu zadáva práve táto hodnota. Čo bude využité v slabej formulácii pri riešení konkrétnej parciálnej diferenciálnej rovnice v kapitole 3.



Obr. 1.1: Príklady 2D sietí vytvorené funkciou `buildmesh()`

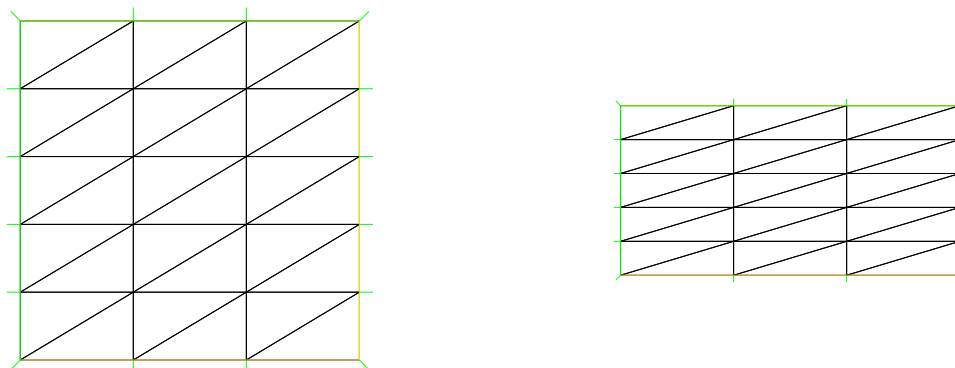
V prípade jednoduchých obdĺžnikových oblastí existuje na vytvorenie siete priamo vstavaná funkcia `square()`, ktorá tiež vráti premennú, sieť, typu `mesh`. Do tejto funkcie vstupuje viacero argumentov, pričom povinné sú prvé dva, ktoré určujú veľkosť dielikov siete v smere x a v smere y . Ďalšie dva parametre slúžia na definovanie hrán siete, ktoré sú definované parametricky. Ak sa neuvedie inak, prednastavený je jednotkový štvorec $[0, 1] \times [0, 1]$.

Na obrázku 1.2 sú znázornené siete vytvorené funkciou `square()`. Sieť na obrázku 1.2 vľavo je definovaná na preddefinovanom štvorci $[0, 1] \times [0, 1]$ príkazmi:

```
int nx=3; //pocet dielikov v smere osi x
int ny=5; //pocet dielikov v smere osi y
// vytvorenie siete:
mesh m = square(nx, ny);
```

Sieť zobrazená na obrázku 1.2 napravo je definovaná opäť pomocou funkcie `square()`, no tentokrát na ľubovolnej obdĺžnikovej oblasti $[x_0, y_0] \times [x_1, y_1]$ pomocou nasledujúcich príkazov:

```
int nx=3; //pocet dielikov v smere osi x
int ny=5; //pocet dielikov v smere osi y
real x0 = 0.;
real x1 = 2.;
real y0 = 0.;
real y1 = 1.;
// vytvorenie siete:
mesh m = square(nx, ny, [x0+(x1-x0)*x, y0+(y1-y0)*y]);
```



Obr. 1.2: Príklady 2D sietí vytvorené funkciou `square()`

Na vyššie uvedenom obrázku 1.2 vľavo a obrázku 1.1 vľavo je vidieť rovnaké štvorcové oblasti $[0, 1] \times [0, 1]$, ktorých sieť je definovaná pomocou 2 rôznych funkcií `square()` a `buildmesh()`. Ako je vidieť, trojuholníkové elementy vytvorené pomocou funkcie `buildmesh()` nie sú pravidelné a rovnako veľké ako pri funkcii `square()`.

V prípade práce v 3D priestore, softvér FreeFEM vytvára sieť pozostávajúcu z elementov v tvare tetraédra (štvorsten). Pri vytváraní každého programu s použitím siete v priestore 3D je nevyhnutné na začiatok načítať plugin `msh3`.

Podobne ako pre vytvorenie siete v 2D priestore existuje funkcia `square()`, ktorá slúži na vytvorenie siete cez obdĺžnikovú oblasť, v 3D priestore je jej analógiou funkcia `cube()`, ktorá vracia premennú, sieť, typu `mesh`. Do nej vstupuje viacero argumentov, pričom povinné sú prvé 3, ktoré označujú počet dielikov v smere osi x, v smere osi y a v smere osi z. Ďalšie parametre slúžia na definovanie rozmeru a sú zadávané parametrickým vyjadrením, bez ich definovania je za oblasť považovaná jednotková kocka $[0, 1] \times [0, 1] \times [0, 1]$. Použitie tejto funkcie, je ukázané v dvoch nasledujúcich programoch.

Na obrázku 1.3 vľavo je oblasť jednotková kocka $[0, 1] \times [0, 1] \times [0, 1]$ a kód na vytvorenie siete pomocou funkcie `cube()` cez túto oblasť je nasledovný:

```
load "msh3" //nacistanie povinného pluginu pre 3d siet
int nx=3; //pocet dielikov v smere x
int ny=5; //pocet dielikov v smere y
int nz=2; //pocet dielikov v smere z
// vytvorenie siete:
mesh3 m = cube(nx, ny, nz);
```

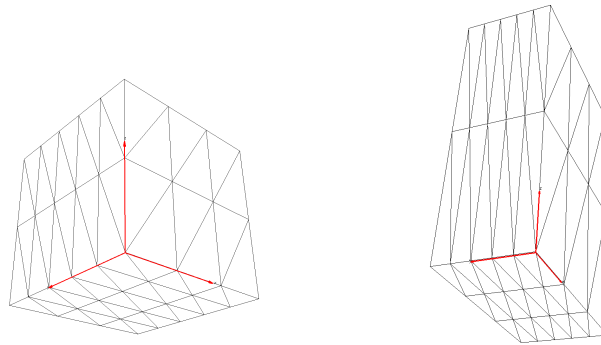
Na obrázku 1.3 vľavo je oblasť ľubovoľne zvolený kváder $[x_0, x_1] \times [y_0, y_1] \times [z_0, z_1]$, cez ktorý je vytvorená sieť taktiež použitím funkcie `cube()` nasledovným postupom:

```
load "msh3" //nacistanie povinného pluginu pre 3d siet
int nx=3; //pocet dielikov v smere x
int ny=5; //pocet dielikov v smere y
```

```

int nz=2; //pocet dielikov v smere z
real x0 = 0.;
real x1 = 2.;
real y0 = 0.;
real y1 = 1.;
real z0 = 0.;
real z1 = 3.;
// vytvorenie siete:
mesh3 m = cube(nx, ny, nz, [x0+(x1-x0)*x, y0+(y1-y0)*y, z0+(z1-z0)*z]);

```



Obr. 1.3: Príklady 3D sietí vytvorené funkciou `cube()`

Ďalšou možnosťou ako vytvoriť sieť cez 3D oblasť, je vytvoriť sieť cez jej hranicu a následne spustením generátor sietí **TetGen**, generovať sieť cez vnútornú oblasť. **TetGen** je softvér vytvorený Doktorom Hang Siom z výskumného inštitútu Aplikovanej analýzy a Stochastiky v Berlíne v Nemecku. Tento softvér slúži na generovanie tetraedrálnej siete na 3D oblasti definovanej jej hranicou. Pre jeho vyúžívanie je nutné na začiatku kódu načítať knižnicu `tetgen`, a na jeho spustenie slúži kľúčové slovo `tetg()`. Tento príkaz obsahuje viacero parametrov, okrem prvého, ktorý je povinný a tvorí ho sieť na hranici sem patria aj:

- **switch** = prepínače, reťazec písmen, pomocou ktorých kontrolujeme správanie generátora,
- **nbofholes** = počet dier v oblasti,
- **holelist** = $[x_1, y_1, z_1, x_2, y_2, z_2, \dots]$ - kde x_i, y_i, z_i sú súradnice bodu vo vnútri i -tej diery,
- **nbofregions** = počet častí (regions) v oblasti,
- **regionlist** = $[x_1, y_1, z_1, Lab_1, mVol_1, x_2, y_2, z_2, \dots]$ - kde x_i, y_i, z_i sú súradnice, Lab_i je označenie oblasti a $mVol_i$ je maximálny objem bodu vo vnútri i -tej časti oblasti.

Medzi prepínače, v argumente `switch` patria napríklad:

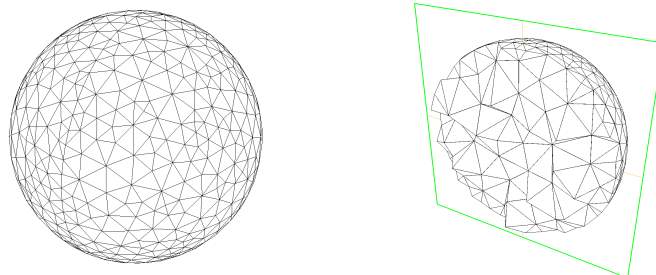
- `p` - úprava elementov na hranici na tetraédre,
- `a` - vytvorenie siete s obmedzením objemu elementov daným v parametri `regionlist`,
- `A` - priradenie vlastností oblasti podľa parametra `regionlist`,
- `Q` - zobrazenie len chybových výstupov v termináli,
- `YY` - zachovanie siete na vnútornej a vonkajšej hranici.

Ako bolo spomenuté vyššie, na využitie generátora `TetGen` je najskôr potrebné vytvoriť sieť na hranici. Pre vytvorenie siete na hranici je možnosť využiť napríklad nástroj `MeshSurface.idp`, po jeho zahrnutí na začiatku programu sú dostupné konštruktory siete cez oblasť sféry `Sphere()`, či cez oblasť elipsoidu `Ellipsoide()`.

Funkcia `Sphere(R, h, L, OX, OY, OZ, orient)`, obsahuje 7 parametrov. `R` definuje jej polomer, `h` je veľkosť siete, `L` je označenie, `OX, OY, OZ` predstavujú stred a `orient` definuje orientáciu. Pokiaľ nie je uvedené inak, stred predstavuje bod so súradnicami `[0, 0, 0]`. Vo funkcii `Ellipsoide(RX, RY, RZ, h, L, OX, OY, OZ, orient)` je zmena len pri definovaní dĺžky polosí v smere jednotlivých osí.

Nasledujúci kód definuje sieť cez oblasť tvaru sféry s polomerom `r` a stredom v bode so súradnicami `[0, 0, 0]` pomocou generátora sietí `tetg()` a konštruktora `Sphere()`. Takto vytvorená sieť je zobrazená spolu s jej rezom na obrázku 1.4

```
load "msh3" //nacitanie povinneho pluginu pre 3d siet
load "tetgen" //nacitanie generatora siete
include "MeshSurface.idp" // zahrnutie nastrojca pre konštruktory siete
real r = 1.; // polomer sfery
real h = 0.2; // velkost siete
meshS s = Sphere(r, h, 1, 1); // vytvorenie siete na hranici:
// vytvorenie siete cez celu oblasť:
real[int] rl = [0., 0., 0., 1, 0.005];
mesh3 m = tetg(s,switch="paAY",regionlist=rl);
```



Obr. 1.4: Príklad siete cez 3D oblasť vytvorenej pomocou `tetg()` a `Sphere()`

Okrajové podmienky:

Riešenie diferenciálnych rovníc pomocou metódy konečných prvkov obsahuje aj zahrnutie okrajových podmienok. Softvér FreeFEM ponúka jednoduché spôsoby ich definovania.

Pri definovaní Dirichletových okrajových podmienok nezáleží, či je definovaná na hranici 2D oblasti alebo 3D oblasti, podmienka je definovaná rovnako v oboch prípadoch: Nech neznáma funkcia u spĺňa dirichletovú okrajovú podmienku, ktorá je definovaná na hranici h

$$u|_h = g, \quad h \subset \mathbb{R} (\mathbb{R}^2).$$

Do kódu pre softvér FreeFEM sa definuje pomocou funkcie

$$\text{on}(h, u=g).$$

Neumannova a Newtonova okrajová podmienka sa vo FreeFEM zadáva priamo do slabej formulácie daného problému.

2D: Nech neznáma funkcia u spĺňa Neumannovú okrajovú podmienku na hranici h , ktorá je definovaná nasledovne

$$\frac{\partial u}{\partial \vec{n}}|_h = g, \quad h \subset \mathbb{R},$$

potom sa vo FreeFEM definuje

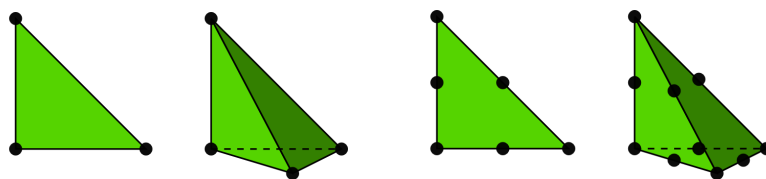
$$\text{int1d}(M, h)(g*w), \tag{1.1}$$

pričom M označuje sieť w je váhová funkcia.

Pre úlohy riešené na 3D oblasti sa pri definovaní Neumannovej podmienky v slabej formulácii zmení v rovnici 1.1 krivkový integrál na plošný $\text{int2d}(M, h)(g*w)$.

Elementy:

Ďalšou dôležitou súčasťou riešenia diferenciálnych rovníc metódou konečných prvkov je voľba elementov. FreeFEM ponúka na výber spomedzi viacerých možností. V aktuálnej verzii softvéru sú dostupné napríklad Lagrangeove elementy, lineárny - P1, kvadratický - P2.



Obr. 1.5: Lineárny a kvadratický element v 2D a 3D [7]

Riešenie úlohy:

V softvéri FreeFEM vystupuje priestor `fespace`. Pomocou neho sa definuje nový dátový typ, ktorý je charakterizovaný sieťou cez danú oblasť a zvoleným typom elementov, `fespace Nazov(Siet', TypKonecnýchPrvkov)`, tohto typu bude hľadaná funkcia ako aj váhová funkcia vystupujúca v slabej formulácii. Znamená to, že softvér automaticky nájde hodnoty hľadanej funkcie v uzloch elementov danej siete, napríklad pri lineárnych elementoch (P1) to budú hodnoty vo vrcholoch trojuholníkových respektíve tetraedrálных elementov.

Na riešenie parciálnej diferenciálnej rovnice je možné použiť dátový typ `solve` alebo `problem`. Definovanie problému je v oboch prípadoch rovnaké, rozdiel nastáva len v tom, že použitím `solve` sa problém okrem deklarácie aj automaticky vyrieši. Naopak použitím `problem` sa daný problém len deklaruje a jeho spustenie nastáva až zavolaním názvu problému, ktorý sa použil pri deklarácií. Pri oboch možnostiach možno definovať metódu riešenia, dostupná je napríklad metóda združených gradientov (CG), Choleskyho rozklad (Cholesky) či Zovšeobecnená metóda minimálnych rezíduí (GMRES).

Takže pre samotné riešenie diferenciálnej rovnice v programe FreeFEM je potrebné vytvoriť sieť cez danú oblasť, zvoliť si typ elementu, zostrojiť slabú formuláciu a zahrnúť okrajové podmienky. Nižšie sú uvedené pseudokódy na ilustrovanie riešenia diferenciálnej rovnice vo FreeFEM a porovnanie rozdielu použitia pri riešení dátové typy `solve` a `problem`.

```
fespace Priestor(Siet, TypKonecnýchPrvkov, Metoda)
Priestor u; // hladana funkcia
Priestor w; // vahova funkcia
// deklaracia problemu a jeho automaticke riesenie
solve NazovProblemu(u,w) = Slaba Formulacia Problemu so zahrnutim OP;
```

V prípade, kedy chceme problém definovať, ale riešenie spustiť v inej časti programu využívame typ `problem`. Takto riešený problém je ilustračne ukázaný v programe nižšie:

```
fespace Priestor(Siet, TypKonecnýchPrvkov, Metoda)
Priestor u; // hladana funkcia
Priestor w; // vahova funkcia
// deklaracia problemu BEZ JEHO RIESENIA
problem NazovProblemu(u,w) = Slaba Formulacia Problemu so zahrnutim OP;
// spustenie riesenia
NazovProblemu;
```

Vizualizácia a export dát:

Zobrazenie siete, riešenia alebo ich export je dôležitou súčasťou využívania každého softvéru pre metódu konečných prvkov. V softvéri FreeFEM je možné použiť viacero funkcií.

Vstavaná funkcia pre vizualizáciu je funkcia `plot()`, ktorá obsahuje viacero parametrov, medzi základné patria:

- `Názov siete`, ktorá sa má vykresliť,
- `Názov funkcie`, ktorá predstavuje riešenie diferenciálnej rovnice,
- `wait = 0` alebo `1` definuje, či sa má čakať pred pokračovaním chodu programu,
- `ps = "názov súboru.formát"`, pod akým sa vizualizácia uloží, formát musí byť `.ps` alebo `.eps`,
- `fill = 0` alebo `1` definuje, či sa farebne vyplní celá sieť podľa hodnôt riešenia,
- `value = 0` alebo `1` definuje, či sa zobrazí škála hodnôt riešenia,
- `cmm = "text"`, ktorý sa zobrazí pri vykreslení.

Povinné je definovanie aspoň jedného z prvých dvoch parametrov, teda buď `Názov siete` alebo `riešenia`. V prípade vykreslenia len riešenia diferenciálnej rovnice je možné stlačením klávesy `M` zobrazíť dodatočne sieť.

Silnejším nástrojom na vykresľovanie je funkcia `medit()`, na jej využitie je nutné načítať na začiatku kódu balíček `medit` pomocou príkazu `load "medit"`. Hlavná výhoda využívania funkcie `medit()` na vykresľovanie je, že v prípade 3D oblastí je možné vykresliť ich rez. Parametre vstupujúce do funkcie sú:

- `"Názov"`, ktorý sa zobrazí pri vykreslení,
- `Názov siete`, ktorá sa má vykresliť,
- `Názov funkcie`, ktorá predstavuje riešenie diferenciálnej rovnice,
- `wait = 0` alebo `1` definuje, či sa má čakať pred pokračovaním chodu programu,
- `save = "riesenie.sol"` - uloženie riešenia vo formáte `.sol`.

V prípade využitia tejto funkcie je povinný parameter `"Názov"` a `Názov siete` pre zobrazenie siete, v prípade zobrazenia riešenie je potrebné pridať `Názov funkcie`. Ostatná manipulácia s vizualizáciou sa vykonáva pomocou pravého tlačidla na myši:

- Farebné zobrazenie riešenia: `Data` → `Toggle metric`,
- Rez : `Clipping` → `Toggle clip`,
 - Zarovanie rezu: `Clipping` → `Toggle capping`,
- Export: `File` → `Hardcopy EPS`.

FreeFEM ponúka uloženie siete a riešenia vo formáte, ktorý je možné následne načítať napríklad v programe Paraview alebo Matlab.

Pre Paraview je nutné načítanie knižnice `load "iovtk"` a následne použitie funkcie `savevtk()`, ktorá obsahuje parametre:

- "nazov.vtu" - reťazec, ktorý obsahuje názov súboru vo formáte .vtu,
- NázovSiete,
- NázovFunkcieRiešenia,
- dataname="nazov", pod ktorým sa v Paraview zobrazia dáta,
- order= int[int] pole obsahujúce 0(P0) alebo 1(P1).

Kapitola 2

Diferenciálna rovnica pre poruchový potenciál

V tejto kapitole je priblížený pojem poruchového potenciálu Zeme, následne zostrojenie geodetickej okrajovej úlohy, ktorá predstavuje parciálnu diferenciálnu rovnicu pre poruchový potenciál, následne je tu aj vytvorenie slabej formulácie danej rovnice pre použitie v softvéri FreeFEM.

2.1 Teória potenciálu

Zem podľa Newtonových zákonov nad svojim povrchom vytvára gravitačné pole. Jeho prejavom je pôsobenie gravitačnej sily v okolí Zeme tak aj na jej povrchu. Okrem gravitačnej sily, v každom bode okolia Zeme ako aj na jej povrchu pôsobí taktiež odstredivá sila, ktorá je dôsledkom rotácie Zeme okolo vlastnej osi. Rotáciou gravitačného poľa dochádza k vytvoreniu tiažového pola Zeme, ktoré je charakterizované vektorom tiažového zrýchlenia \vec{g} . Toto tiažové pole možno popísať veličinou, ktorá sa nazýva tiažový potenciál W , ktorý je tvorený gravitačným W_g a odstredivým W_o potenciálom [1]

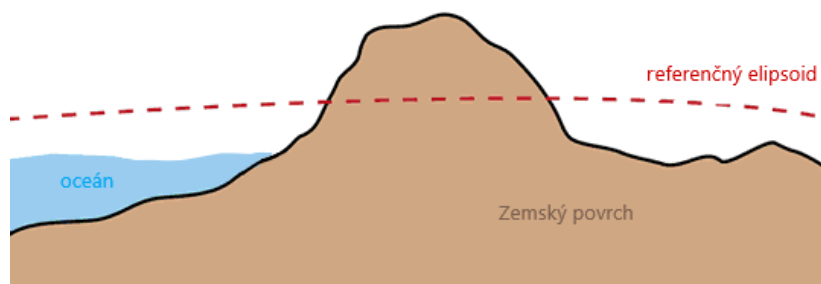
$$W(X) = W_g(X) + W_o(X), \quad X \in \mathbb{R}^3. \quad (2.1)$$

Potom vektor tiažového zrýchlenia \vec{g} predstavuje gradient tiažového potenciálu W

$$\vec{g} = \nabla W, \quad (2.2)$$

$$\nabla W = \left(\frac{\partial W}{\partial x}, \frac{\partial W}{\partial y}, \frac{\partial W}{\partial z} \right). \quad (2.3)$$

Zemský povrch je tvorený rovinami, horstvami, moriami či ostrovmi a preto sa kvôli svojej zložitosti využívajú jeho geometrické aproximácie. Najčastejšie sa jedná o referenčný elipsoid.



Obr. 2.1: Zemský povrch a jeho aproximácia referenčným elipsoidom [3]

V histórii bolo definovaných množstvo približných rozmerov Zeme. Parametre elipsoid sa získavajú zo stupňových pozemných meraní, vzhľadom na nepravidelnosť zemského povrchu môže byť pre rôzne oblasti vhodné použiť rôzne elipsoidy. Najlepšou aproximáciou Zeme by bol trojosový zemský elipsoid, ktorého geometria je stále veľmi zložitá a preto sa v geodetickej praxi využívajú dvojsovové rotačné elipsoidy, ktorým sú pridelené geometrické a fyzikálne vlastnosti Zeme. [1]

Potenciál generovaný referenčným elipsoidom označme ako normálový tiažový potenciál U , ktorý je zostrojený podobne ako skutočný tiažový potenciál W [11]

$$U(X) = U_g(X) + U_o(X), \quad X \in \mathbb{R}^3, \quad (2.4)$$

kde U_g označuje gravitačný potenciál a U_o odstredivý potenciál.

Odčítaním skutočného W a normálneho U poruchového potenciálu vznikne poruchový potenciál T

$$T(X) = W(X) - U(X), \quad x \in \mathbb{R}^3. \quad (2.5)$$

Zanedbaním atmosféry, poruchový potenciál T je harmonická funkcia, ktorá v priestore mimo Zeme spĺňa Laplaceovu rovnicu. [9]

$$\Delta T(X) = 0 \quad x \in \mathbb{R}^3, \quad (2.6)$$

2.2 Geodetická okrajová úloha

Rozdiel medzi vektorom skutočného tiažového zrýchlenia \vec{g} a vektorom normálneho tiažového zrýchlenia $\vec{\gamma}$ sa definujú ako vektor poruchovej anomálie $\vec{\delta g}$ [6],

$$\vec{\delta g} = \vec{g} - \vec{\gamma}, \quad (2.7)$$

Vývoj satelitných technológií do geodézie priniesol nové možnosti. Z gravimetrických a satelitných meraní dostávame veľkosť g vektora skutočného tiažového zrýchlenia \vec{g} v bode P na zemskom povrchu. Veľkosť vektora normálneho tiažového zrýchlenia v rovnakom bode sa označuje γ , potom rozdielom týchto dvoch veľkostí vznikne tiažová porucha [12]

$$\delta g = g - \gamma. \quad (2.8)$$

Geodetická okrajová úloha s pevnou hranicou na oblasti Ω je definovaná nasledovne:

$$\Delta T(X) = 0 \quad X \in \Omega, \quad (2.9)$$

$$\Delta T(X) = \frac{\partial^2 T}{\partial^2 x} + \frac{\partial^2 T}{\partial^2 y} + \frac{\partial^2 T}{\partial^2 z}, \quad (2.10)$$

prislúchajúce okrajové podmienky:

$$\begin{aligned} \frac{\partial T}{\partial \vec{n}} &= \delta g^*(X) & X \in \Gamma \\ T(X) &= T_S(X) & X \in \partial\Omega - \Gamma \end{aligned} \quad (2.11)$$

pričom $\frac{\partial T}{\partial \vec{n}} = \delta g(X) * \cos(\theta) = \delta g^*(X)$, kde θ je uhol medzi vonkajším normálovým vektorom \vec{n} na Γ a vektorom smeru normálneho tiažového zrýchlenia \vec{s} , ďalej T_S označuje hodnotu poruchového potenciálu získaného z meraní satelitov. Oblasť Ω je ohraničená zvnútra hranicou Γ , ktorá predstavuje povrch Zeme a zvonka hranicou $\partial\Omega - \Gamma$ umelo vytvorenou vo výške satelitu [11].

2.3 Slabá formulácia

Nižšie je zobrazený postup vytvorenia slabej formulácie pre rovnicu 2.9, ktorá je potrebná pri riešení v softvéry FreeFEM.

Rovnicu (2.9) je nutné vynásobiť váhovou funkciou w a zintegrovať cez oblasť Ω

$$\int_{\Omega} w \Delta T \, d\Omega = 0 \quad (2.12)$$

$$\int_{\Omega} -w \left[\frac{\partial}{\partial x} \left(\frac{\partial T}{\partial x} \right) + \frac{\partial}{\partial y} \left(\frac{\partial T}{\partial y} \right) \right] \, d\Omega = 0 \quad (2.13)$$

pre 3D verziu by sa ešte pridal člen: $\frac{\partial}{\partial z} \left(\frac{\partial T}{\partial z} \right)$.

Následne sa na (2.13) aplikuje Greenova veta [4]:

$$\int_G f \frac{\partial g}{\partial x_i} dx = \int_{\partial G} f g n_i dS - \int_G \frac{\partial f}{\partial x_i} g dx \quad (2.14)$$

Po jej aplikácii vznikne nasledový tvar rovnice:

$$\int_{\partial\Omega} -w \left(\frac{\partial T}{\partial x} n_x + \frac{\partial T}{\partial y} n_y \right) d\partial\Omega + \int_{\Omega} \left(\frac{\partial w}{\partial x} \frac{\partial T}{\partial x} + \frac{\partial w}{\partial y} \frac{\partial T}{\partial y} \right) d\Omega = 0, \quad (2.15)$$

kd' výrazy v zátvorkách možno upraviť na skalárny súčin vektorov:

$$\left(\frac{\partial T}{\partial x} n_x + \frac{\partial T}{\partial y} n_y \right) = \nabla T \cdot \vec{n} = q_n \quad (2.16)$$

$$\left(\frac{\partial w}{\partial x} \frac{\partial T}{\partial x} + \frac{\partial w}{\partial y} \frac{\partial T}{\partial y} \right) = \nabla w \cdot \nabla T \quad (2.17)$$

Dosadením (2.16) a (2.17) do (2.15) vznikne konečný tvar slabej formulácie:

$$\int_{\Omega} \nabla w \cdot \nabla T d\Omega = \int_{\partial\Omega} w q_n d\partial\Omega \quad (2.18)$$

Kapitola 3

Experimenty

V tejto kapitole budú predstavené experimenty vykonané na jednoduchých oblastiach, na ktorých sa úkaze práca so softvérom. Rovnako tak v tejto kapitole bude pri experimentoch vypočítaná L_2 norma chyby približného riešenia pre zjemňujúcu sa sieť spolu s experimentálnym radom konvergenzie.

3.1 L_2 norma chyby a EOC

Pre vypočet chyby pri experimentoch vo FreeFEM sa bude využívať L_2 norma chyby, ktorá sa vypočíta ako

$$\|T - T_e\|_{L_2} = \sqrt{\int_{\Omega} (T(x) - T_e(x))^2 d\Omega}, \quad (3.1)$$

kde T je približné riešenie a T_e je presné riešenie.

Ďalej bude odhadnutý aj rád presnosti, pomocou výpočtu experimentálneho rádu konvergenzie, ktorý sa počíta nasledovným spôsobom

$$EOC(h, h^*) = \log_{\frac{h}{h^*}} \left(\frac{\text{chyba}(h)}{\text{chyba}(h^*)} \right) = \frac{\log_e \left(\frac{\text{chyba}(h)}{\text{chyba}(h^*)} \right)}{\log_e \left(\frac{h}{h^*} \right)}, \quad (3.2)$$

kde h je veľkosť elementov a h^* je veľkosť elementov po zjemnení siete.

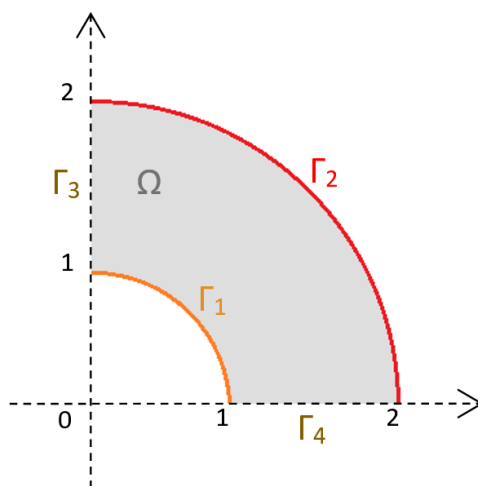
3.2 Experimenty pre 2D oblasť

Presné riešenie T_e pre Laplaceovu rovnicu 2.9 na oblasti medzikružia je dané ako prirodzený logaritmus vzdialenosti od stredu oblasti

$$T_e = -\log(\sqrt{x^2 + y^2}). \quad (3.3)$$

3.2.1 Výpočtová oblasť je medzikružie v prvom kvadrante

Prvý experiment v programe FreeFEM v 2D bude uvažovať oblasť Ω ako 1/4 medzikružia v prvom kvadrante s vnútorným polomerom $r_1 = 1$ a vonkajším polomerom $r_2 = 2$, stred bude bod so súradnicami $[0, 0]$.



Obr. 3.1: Výpočtová oblasť pre 1. prípad v 2D

Pre takúto oblasť sa k rovnici 2.9 budú uvažovať okrajové podmienky v tvare: Dirichletova okrajová podmienka na vonkajšie hrane:

$$T(X) = -\log(r_2), \quad X \in \Gamma_2 \quad (3.4)$$

Neumannová okrajová podmienka na vnútornej hrane:

$$\frac{\partial T(X)}{\partial \vec{n}} = -\frac{1}{r_1}, \quad X \in \Gamma_1, \quad (3.5)$$

Nulové Neumannové okrajové podmienka na zvyšných hranách:

$$\frac{\partial T(x)}{\partial \vec{n}} = 0, \quad x \in \Gamma_3, \Gamma_4, \quad (3.6)$$

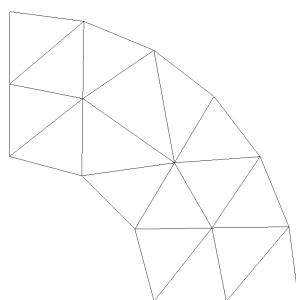
kde \vec{n} je vonkajší normálový vektor na príslušnú hranicu. Softvér FreeFEM má opačne otočenú normálu \vec{n} a preto sa Neumannová podmienka v programe definuje bez znamienka mínus ako bude vidieť v kódoch nižšie.

Postup implementácia úlohy do softvéru FreeFEM bude pozostávať z vytvorenia siete cez danú oblasť zobrazenú na obrázku 3.1. Potom sa prepíše slabá formulácia problému do požadovaného tvaru pre softvér, zahrnú sa okrajové podmienky. Nakoniec sa vykreslí riešenie a vypočíta chyba pre zjemňujúcu sa sieť.

Vytvorenie a vykreslenie siete

Nasledujúca časť kódu zobrazuje sieť cez oblasť Ω na obrázku 3.1, ktorá má hranu Γ_1 a Γ_4 rozdelenú na 2 dieliky, vnútornú hranu Γ_1 na 3 dieliky a vonkajšiu Γ_2 hranu na 6 dielikov. Každá hrana je zadaná parametrickým vyjadrením, a následné sú jednotlivé hrany spojené a vytvorená sieť pomocou funkcie `buildmesh()`. Tvorenie siete je detailnejšie popísané v časti 1.3.1.

```
load "medit";
real r1 = 1.; //vnutorny polomer
real r2 = 2.; //vonkajsi polomer
real h=r2-r1; //vyska
int gamma1=1; //oznacenie vonkajsej hrany
int gamma2=2; //oznacenie vnutornej hrany
// HRANY OBLASTI
border inner(t=0, pi/2) {x= r1*cos(pi/2-t); y= r1*sin(pi/2-t);
    label=gamma1;}; //vnutorna hrana
border outer(t=0, pi/2) {x= r2*cos(t); y= r2*sin(t); label=gamma2;};
    //vonkajsia hrana
border left(t=0, 1) {x=0; y=r2-h*t;}; // gamma3
border right(t=0, 1) {x=r1+h*t; y=0;}; // gamma4
// VYTVORENIE SIETE
mesh m = buildmesh(right(2) + outer(6) + left(2) + inner(3));
medit("SIET", m); //Vykreslenie
```



Obr. 3.2: Vykreslenie siete cez oblasť Ω , ktorá je na obrázku 3.1

Riešenie úlohy

Na vyriešenie úlohy program FreeFEM využíva slabú formuláciu problému.

$$\int_{\Omega} \nabla w \cdot \nabla T \, d\Omega - \int_{\partial\Omega} w \frac{\partial T}{\partial \vec{n}} \, d\partial\Omega = 0 \quad (3.7)$$

Prepis slabej formulácie v programe vyzerá spolu so zahrnutím Neumannovej okrajovej podmienky nasledovne:

$$\text{int2d}(m) (\text{dx}(T)*\text{dx}(w)+\text{dy}(T)*\text{dy}(w))-\text{int1d}(m,\text{gamma1})(w*n)$$

Zahrnutie Dirichletovej okrajovej podmienky sa vo FreeFEM spraví pripísaním `+on(gamma2,T=d)` za slabú formuláciu.

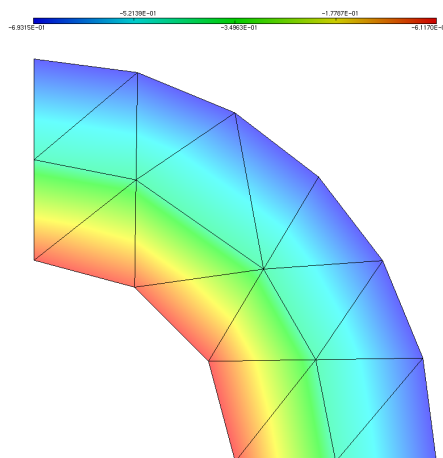
- `m` = sieť cez oblasť Ω
- `gamma1` = označenie hrany Γ_1
- $n = \frac{\partial T}{\partial \vec{n}} = \frac{1}{r_1}$ (Neumannova okrajová podmienka)
- `gamma2` = označenie hrany Γ_2
- `d` = $-\log(r_2)$ (Dirichletova okrajová podmienka)

Nižšie uvedená časť kódu predstavuje riešenie problému pomocou `solve` s použitím Lagrangeovských lineárnych elementov P1.

```

real n = 1/r1; //Neumann
real d = -log(r2); //Dirichlet
fespace SpaceP1(m, P1); //definovanie priestoru
SpaceP1 T, w;
//RIESENIE
solve Laplace(T,w,solver=CG)
  = int2d(m) ( dx(T)*dx(w) + dy(T)*dy(w) )
  - int1d(m,gamma1)( w*n ) //neumann
  + on(gamma2, T=d) //dirichlet
  ;
medit("RIESENIE",m,T); //Vykreslenie

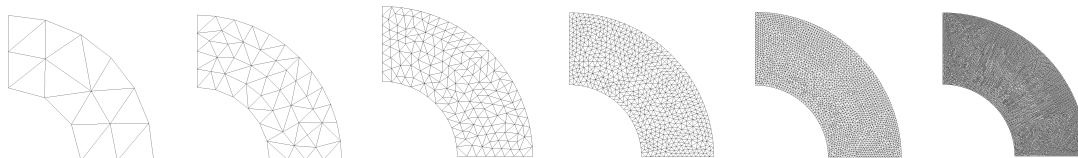
```



Obr. 3.3: Vykreslenie riešenia cez vytvorenú sieť

L2 norma chyby a jej EOC

Pre zjemňujúcu sa sieť na danej oblasti je v nižšie uvedenej tabuľke 3.1 zobrazená L_2 norma chyby riešenia spolu s experimentálnym rádom konvergencie. Zjemňovanie je vždy dvojnásobné, teda na každej hrane sa zdvojnásobí počet dielov ako je vidieť na obrázku nižšie.



Obr. 3.4: Vykreslenie zjemňujúcej sa siete

V tabuľke nižšie prvý stĺpec definuje počet dielikov na každej hrane pre zjemňujúcu sa sieť, tak ako je to zadávané aj v samotnom softvéri.

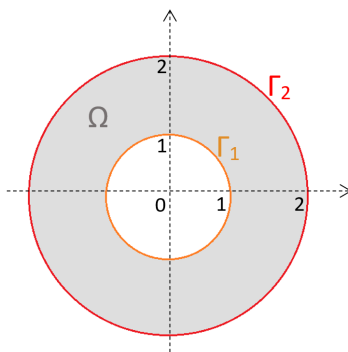
$n_{\Gamma_1} \times n_{\Gamma_2} \times n_{\Gamma_3} \times n_{\Gamma_4}$	L_2 norma	EOC_{L_2}
$3 \times 6 \times 2 \times 2$	0.0149868	
$6 \times 12 \times 4 \times 4$	0.0033381	2.166
$12 \times 24 \times 8 \times 8$	0.0008693	1.941
$24 \times 48 \times 16 \times 16$	0.0002179	1.996
$48 \times 96 \times 32 \times 32$	0.0000532	2.034
$96 \times 192 \times 64 \times 64$	0.0000137	1.954

Tabuľka 3.1: L^2 chyba a EOC pre zjemňujúcu sa sieť.

Ako možno vidieť v tabuľke, chyba metódy so zjemňovaním siete klesá a to vždy približne štvornásobne, rovnako vidíme, že rád presnosti metódy je 2.

3.2.2 Výpočtová oblasť je celé medzikružie

V druhom prípade sa bude uvažovať oblasť Ω celé medzikružie ako je vidieť na obrázku 3.5 nižšie.



Obr. 3.5: Výpočtová oblasť pre 2. prípad v 2D

Pre takúto oblasť sa k rovnici 2.9 bude uvažovaná rovnaká Dirichletová 3.9 aj Neumannová okrajová podmienka 3.10 ako to bolo v predchádzajúcom prípade akurát už tu nebude vystupovať nulová Neumannová okrajová podmienka.

Vytvorenie a vykreslenie siete

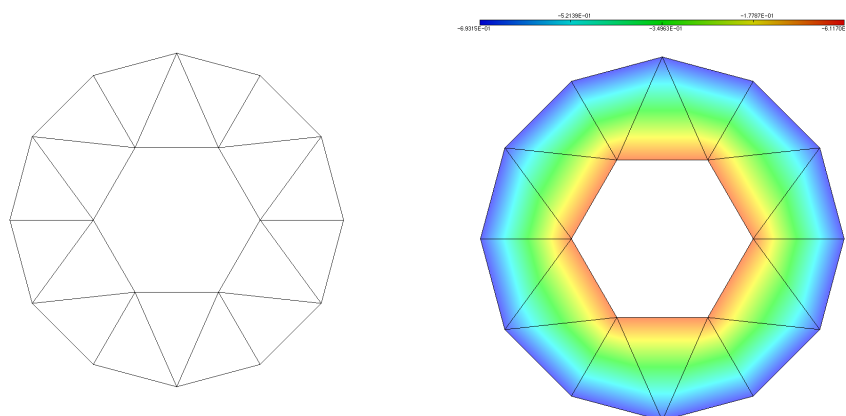
Nasledujúci kód ukazuje ako vytvoriť sieť cez vyššie zobrazenú oblasť Ω , jej vonkajšia hranica Γ_2 je rozdelená na 12 dielikov a vnútorná Γ_1 na 6 dielikov. Tvorenie siete je detailnejšie popísané v časti 1.3.1.

```
load "medit";
real r1 = 1.; //vnutorny polomer
real r2 = 2.; //vonkajsi polomer
int gamma1=1; //oznacenie vonkajsej hrany
int gamma2=2; //oznacenie vnutornej hrany
// HRANY OBLASTI
border inner(t=0, 2*pi) {x= r1*cos(t); y= r1*sin(t); label=gamma1;};
border outer(t=0, 2*pi) {x= r2*cos(t); y= r2*sin(t); label=gamma2;};
// VYTVORENIE SIETE
mesh m = buildmesh(inner(-6) + outer(12));
medit("SIET", m); // vykreslenie
```

Riešenie úlohy

Pri riešení úlohy cez oblasť Ω nenastáva v kóde žiadna zmena voči predchádzajúcemu prípadu 3.2.1.

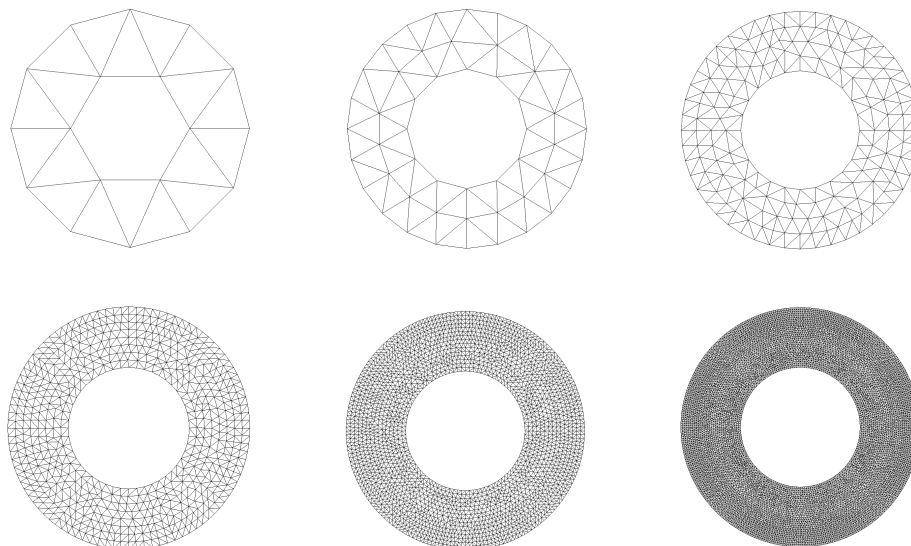
Definovaná sieť v kóde 3.2.2, a riešenie problému na takejto oblasti vyzerá nasledovne



Obr. 3.6: Vykreslenie siete spolu s riešením

L2 norma chyby a jej EOC

Na nižšie zobrazených obrázkoch je znázornené zjemňovanie siete cez oblasť Ω , spôsobom, že na vnútornej hranici Γ_1 aj na vonkajšej hranici Γ_2 sa počet dielikov vždy zdvojnásobí.



Obr. 3.7: Vykreslenie zjemňujúcej sa siete m

K riešeniam na vyššie uvedených sieťach prislúcha nasledujúca tabuľka, ktorá zobrazuje L_2 normu chyby a aj experimentálny rad konverencie, ktorým odhadnem rád presnosti použitej metódy. V prvom stĺpci je uvedený počet dielikov na vnútornej Γ_1 a na vonkajšej Γ_2 hranici.

$n_{\Gamma_1} \times n_{\Gamma_2}$	L_2 norma	EOC_{L_2}
6×12	0.119585	
12×24	0.029214	2.0333
24×48	0.007466	1.9681
48×96	0.001770	2.0759
96×192	0.000438	2.0129
192×384	0.000108	2.0109

Tabuľka 3.2: L^2 chyba a EOC pre zjemňujúcej sa sieť

V tabuľke 3.2 je vidieť, že so zjemňujúcou sa sieťou klesá aj chyba, a to približne 4 násobne s dvojnásobne hustejšou sieťou, pričom zvolená metóda je druhého rádu presnosti.

3.3 Experimenty pre 3D oblasť

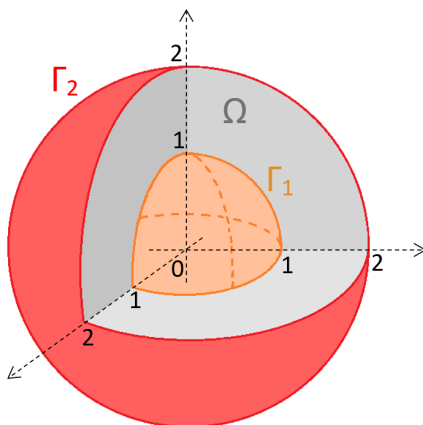
Presné riešenie pre rovnicu 2.9 na oblasti medzi dvomi sférami je definované nasledovne:

$$T_e = \frac{1}{\sqrt{x^2 + y^2 + z^2}}, \quad (3.8)$$

kde $\sqrt{x^2 + y^2 + z^2}$ vyjadruje vzdialenosť od stredu oblasti, ktorý je v bode $[0, 0, 0]$.

3.3.1 Výpočtová oblasť je medzi dvomi sférami

Pre riešenie diferenciálnej rovnice v 3D priestore sa bude uvažovať oblasť Ω ako oblasť vytvorená dvomi sférami, jedna s polomerom $r_1 = 1$ bude tvoriť vnútornú časť hranice a druhá s polomerom $r_2 = 2$ bude tvoriť vonkajšiu časť hranice. Stred oboch sfér bude v bode so súradnicami $[0, 0, 0]$. Na obrázku nižšie, je táto oblasť zobrazená.



Obr. 3.8: Výpočtová oblasť pre 3D prípad

Pre túto oblasť a rovnicu 2.9 sa budú v experimentoch uvažovať okrajové podmienky v tvare:

Dirichletova okrajová podmienka na vonkajšie časti hranice:

$$T(X) = \frac{1}{r_2}, \quad X \in \Gamma_2 \quad (3.9)$$

Neumannová okrajová podmienka na vnútornej časti hranice:

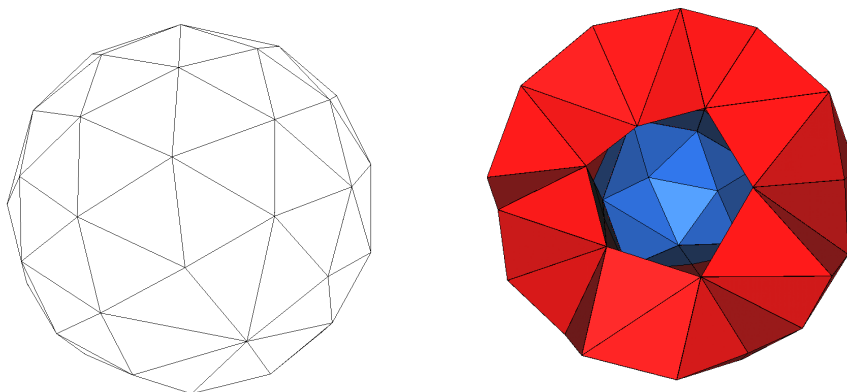
$$\frac{\partial T(X)}{\partial \vec{n}} = -\frac{1}{r_1^2}, \quad X \in \Gamma_1, \quad (3.10)$$

Softvér FreeFEM má opačne otočenú normálu \vec{n} a preto sa Neumannová podmienka v programe definuje bez znamienka mínus ako bude vidieť v kódach nižšie.

Vytvorenie a vykreslenie siete

Pre vytvorenie siete cez oblasť zobrazenú na obrázku 3.8 v softvéri FreeFEM je využitá funkcia `Sphere()` pre definovanie siete na hranici a následne pomocou funkcie `tetg()` je vytvorená sieť z elementov tvaru tetraédra medzi hranicami. Detailnejší popis týchto funkcií je v časti 1.3.1.

```
load "msh3"
load "tetgen"
load "medit"
include "MeshSurface.idp"
real r1=1.; //vnutrony polomer
real r2=2.; // vonkajsi polomer
real hs = 1.; //velkost siete na hranici
int gamma1=1; //oznacenie vonkajsej hrany
int gamma2=2; //oznacenie vnutornej hrany
// VYTVORENIE SIETI NA HRANICI
meshS m1 = Sphere(r1, hs, gamma1, -1);
meshS m2 = Sphere(r2, hs, gamma2, 1);
meshS mS = m1+m2;
// VYTOVRENIE SIETE CEZ CELY OBJEM
real[int] hole = [0.,0.,0.]; //suradnice
real voltet=(hs^3)/6.; //objem elementu
real[int] region = [1.5,0.,0.,1,voltet]; //vlastnosti oblasti
mesh3 m =
    tetg(mS,switch="paAYY",nbofholes=1,holelist=hole,nbofregions=1,
        regionlist=region);
medit("SIET", m); // vykreslenie
```



Obr. 3.9: Vykreslenie siete m

Riešenie

Opäť, ako pri predchádzajúcom príklade treba slabú formuláciu problému previesť do reči softvéru a zahrnúť okrajové podmienky. Prepis slabej formulácie 3.7 a zahrnutie Neumannovej okrajovej podmienky vyzerá nasledovne:

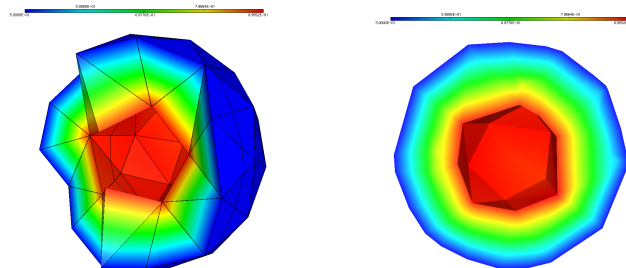
$$\text{int3d}(m) (\text{dx}(T)*\text{dx}(w)+\text{dy}(T)*\text{dy}(w)+\text{dz}(T)*\text{dz}(w))-\text{int2d}(m,\text{gamma1})(w*n)$$

Rovnakým spôsobom ako v 2D prípade, zahrnutie Dirichletovej okrajovej podmienky nastáva pripísaním výrazu `+ on(gamma2, T=d)` za slabú formuláciu.

- `m` = sieť cez oblasť Ω
- `gamma1` = označenie hrany Γ_1
- `n` = $\frac{\partial T}{\partial \vec{n}} = \frac{1}{r_1^2}$ (Neumannova okrajová podmienka)
- `gamma2` = označenie hrany Γ_2
- `d` = $\frac{1}{r_2}$ (Dirichletova okrajová podmienka)

Nižšie uvedená časť kódu predstavuje riešenie problému pomocou `solve` s použitím Lagrangeovských lineárnych elementov P1 a pomocou vyššie definovanej slabej formulácie a vytvorenej siete cez danú oblasť.

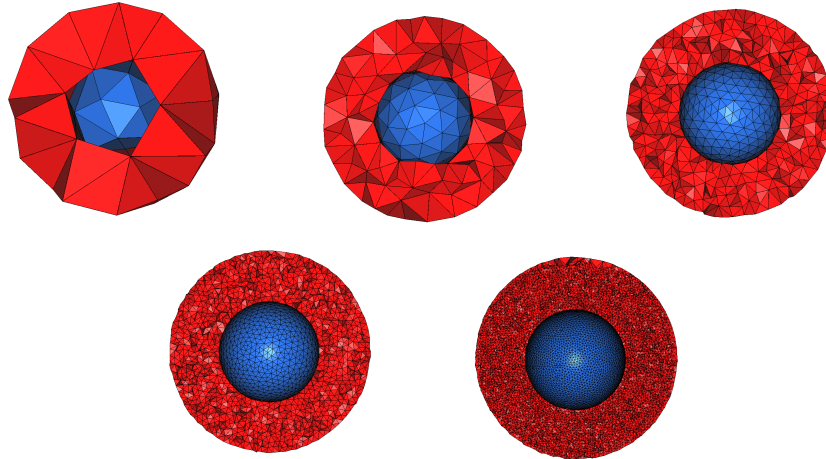
```
real n = 1/r1*r1; // Neuman
real d = 1/r2; // Dirichlet
fespace SpaceP1(m, P1);
SpaceP1 T, w;
// RIESENIE
solve Laplace(T,w, solver = CG)
  = int3d(m)( dx(T)*dx(w) + dy(T)*dy(w) + dz(T)*dz(w) )
  - int2d(m,gamma1)( w*n ) //neumann
  + on(gamma2, T=d) //dirichlet
  ;
medit("Solution",m,T); //vykreslenie
```



Obr. 3.10: Vykreslenie riešenia

L2 norma chyby a jej EOC

Na nižšie zobrazených obrázkoch je znázornené zjemňovanie siete tak, že na vnútornej hranici Γ_1 aj na vonkajšej hranici Γ_2 sa veľkosť siete vždy zmenší o polovicu. Následne pomocou tejto veľkosti je definovaná sieť celej oblasti na základe maximálneho objemu elementu.



Obr. 3.11: Vykreslenie zjemňujúcej sa siete

K riešeniam vypočítaným na vyššie uvedených sieťach prislúcha nasledujúca tabuľka, ktorá zobrazuje L_2 normu chyby spolu s experimentálnym radom konvergenzie, ktorým odhadnem rád presnosti použitej metódy. V prvom stĺpci je uvedená veľkosť siete na vnútornej Γ_1 a na vonkajšej Γ_2 hranici.

veľkosť siete	L_2 norma	EOC_{L_2}
1.0000	0.31064	
0.5000	0.07608	2.02954
0.2500	0.01644	2.21007
0.1250	0.00430	1.93232
0.0625	0.00136	1.65884

Tabuľka 3.3: L^2 chyba a EOC pre zjemňujúcu sa sieť

V tabuľke 3.3 je vidieť, že so zjemňujúcou sa sieťou klesá aj chyba a to približne štvornásobne s dvojnásobne hustejšou sieťou, pričom zvolená metóda je druhého rádu presnosti.

Kapitola 4

Vizualizácia pomocou ParaView

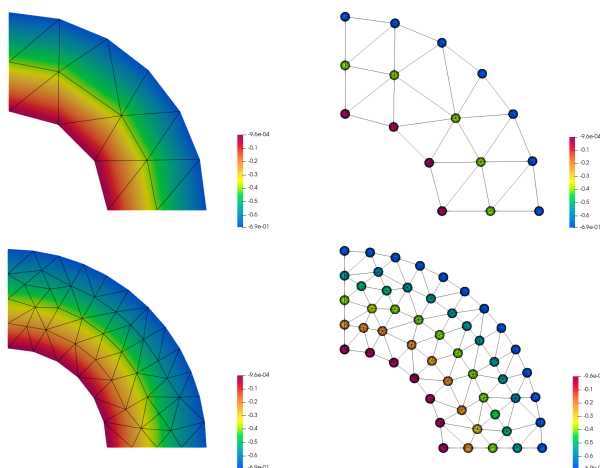
Ako bolo uvedené v prvej kapitole, dáta z FreeFEM možno exportovať aj pre vizualizáciu v iných softvéroch. V tejto kapitole je uvedená vizualizácia výsledkov z predchádzajúcej kapitoly pomocou softvéru ParaView.

Vo všetkých 3 nasledujúcich vizualizáciách je použitý nižšie uvedený kód pre export dát z FreeFEM. Detailnejší popis použitia funkcie `savevtk()` je v 1. kapitole.

```
load "iovtk";  
int [int] o = [1];  
savevtk("riesenie.vtk", m, T, dataname="T", order=o);
```

4.1 Vizualizácia riešenia na výseku medzikružia

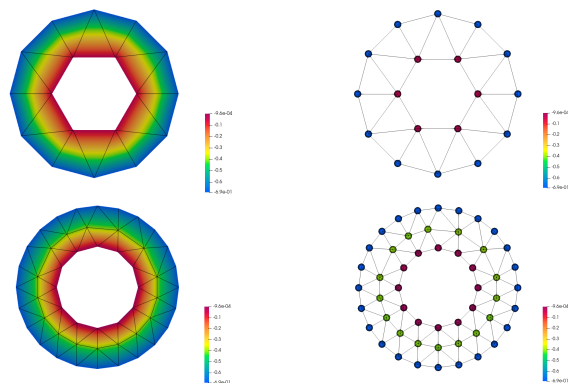
Na obrázku nižšie je zobrazené riešenie vypočítané v časti 3.2.1 spolu s riešením aj na jemnejšej sieti. Na obrázkoch vpravo sú vidieť body, v ktorých sa počítalo riešenie. Keďže som pracovala s lineárnymi elementami, tieto body predstavujú vrcholy elementu.



Obr. 4.1: Vizualizácia riešenia v ParaView

4.2 Vizualizácia riešenia na celom medzikruží

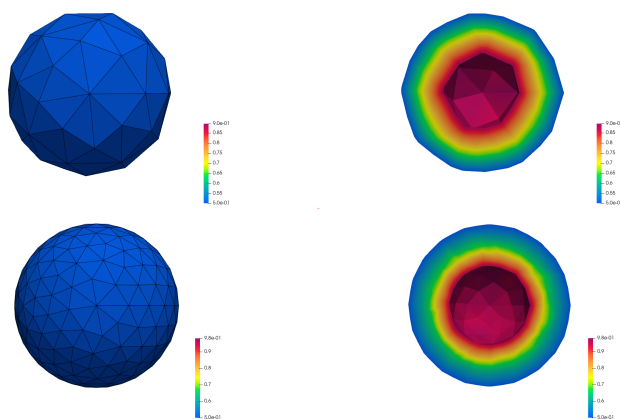
Na obrázku nižšie je opäť zobrazené riešenie tenzoraz vypočítané v časti 3.2.2, rovnako aj riešenie na jemnejšej sieti.



Obr. 4.2: Vizualizácia riešenia v ParaView

4.3 Vizualizácia riešenia na oblasti medzi 2 sférami

Na obrázku nižšie je opäť zobrazené riešenie tenzoraz vypočítané v časti 3.3.1, rovnako aj riešenie na jemnejšej sieti. Na obrázkoch vpravo je rez cez tieto oblasti pre lepšiu viditeľnosť riešenia.



Obr. 4.3: Vizualizácia riešenia v ParaView

Ako je vidieť na obrázkoch vyššie, v ParaView je možné napríklad zobrazit samotné body, v ktorých bolo riešenie počítané. Softvér ParaView má mnoho ďalších veľmi zaujímavých možností pre vykresľovanie riešenia, ktoré vo FreeFEM nie sú zrealizovateľné. No samotná funkcia `medit` vo FreeFEM je na základné vykresľovanie postačujúca pretože rovnako ako v ParaView je možné oblasť prezerat, čo je dôležité pre 3D oblasti a teda nie je nutné kvôli rezu dáta exportovať do ParaView.

Záver

V bakalárskej práci bol predstavený softvér FreeFEM, ktorý využíva metódu konečných prvkov na riešenie problémov. Využitie softvéru bolo ukázané na riešení Laplaceovej rovnice pre poruchový potenciál mimo povrchu Zeme spolu s Dirichletovou a Neumannovou okrajovou podmienkou. Táto úloha sa riešila na 3 rôznych oblastiach, v 2D priestore to bola časť medzikružia a celé medzikružie, v 3D priestore bola oblasť tvorená priestorom medzi dvomi sférami. Pri všetkých experimentoch bolo vykonané zjemňovanie siete pre pozorovanie či vôbec, a ak, tak ako veľmi sa L_2 norma chyby znižuje a aký je experimentálny rad konvergenzie (EOC), ktorým bol odhadnutý rád presnosti. Zjemňovanie siete bolo vo všetkých prípadoch dvojnásobné pričom vypočítaná chyba sa znižovala vždy o približne štvrtinu. Pomocou EOC bol rád pri všetkých experimentoch vypočítaný na hodnotu približne 2.

Čo sa týka práce so softvérom FreeFEM, pri vytváraní siete v 2D nenastal žiadny problém ani pri následnom zjemňovaní. Naopak, pri definovaní siete v 3D, nie len, že samotné vytváranie je značne komplikovanejšie ale pri určitom prekročení veľkosti elementu došlo k nerovnomernému generovaniu siete na hranici sféry, čo však nemalo významný vplyv na vypočítanú chybu. Tento jav nebolo možné ovplyvniť pretože softvér sieť generuje sám na základe definovaných vstupných argumentov. Spôsob, ktorým by sa tomu dalo predísť je načítanie siete z externého súboru. FreeFEM podporu napríklad načítanie siete formátu `.vtk`. Avšak cieľom práce bola práca so samotným softvérom a jeho generovaním siete.

Silnou stránkou softvéru je využívanie slabej formulácie, takže je definovanie riešenia problému pomerne jednoduché pokiaľ užívateľ vie zostaviť slabú formuláciu. Nevýhodou však je, že pri definovaní okrajových podmienok je možné zadávať len funkciu závislú od x, y prípadne z . A teda pokiaľ je pre daný problém definovaná okrajová podmienka inou hodnotou v každom uzle, je nutné tieto hodnoty nejakým spôsobom preložiť vhodnou funkciou čo predstavuje značný problém a preto v tejto práci takéto okrajové podmienky nie sú využité.

Ďalším pokračovaním výskumu využitia softvéru FreeFEM pre geodetickú úlohu, ktoré sa pre krátkosť času nedalo stihnúť, by mohlo byť využitie možnosti načítania siete z externého súboru a zamerať sa na nájdenie vhodnej funkcie pre okrajové podmienky, kvôli práci s reálnymi dátami.

Literatúra

1. BITTERER, L. Geodézia, Interný učebný text katedry geodézie pre študentov bakalárskeho štúdia odboru geodézia a kartografia. 2003. Dostupné tiež z: <https://svf.uniza.sk/kgd/uchadzaci/studijna-literatura>. [Online; dostupné dňa 26-Apríl-2023].
2. BRIMICH, L. Tiažové pole Zeme. 2000,2006. Dostupné tiež z: <http://www.fyzikazeme.sk/mainpage/index.html>. [Online; dostupné dňa 26-Apríl-2023].
3. Geoid illustration. 2023. Dostupné tiež z: <https://www.usgs.gov/media/images/geoid-illustration>. [Online; dostupné dňa 26-Apríl-2023].
4. HANDLOVIČOVÁ, Z.; TIBENSKÝ, M. *Základy funkcionálnej analýzy a variačného počtu*. Slovenská technická univerzita v Bratislave, 2016.
5. HECHT, F. New development in FreeFem++. *J.Numer.Math.* 2022. Dostupné tiež z: <https://freefem.org/>.
6. HOFMANN-WELLENHOF B.; MORITZ, H. *Physical Geodesy*. Springer Science Business Media, 2006. ISBN 9780072466850.
7. LOGG, A.; MARDAL, K.-A.; WELLS, G. *Automated Solution of Differential Equations by the Finite Element Method*. Springer Berlin, Heidelberg, 2012. Dostupné z DOI: <https://doi.org/10.1007/978-3-642-23099-8>.
8. MURÍN, J.; HRABOVSKÝ, J.; KUTIŠ, V. Metóda konečných prvkov, Vybrané kapitoly pre mechatronikov. 2014. Dostupné tiež z: <http://uamt.fei.stuba.sk/web/?q=sk/content/met%C3%B3da-kone%C4%8Dn%C3%BDch-prvkov>. [Online; dostupné dňa 26-Apríl-2023].
9. SCHWARZOVÁ, P. *Metóda nekonečných prvkov ako nástroj riešenia geodetickej okrajovej úlohy, diplomová práca*. Slovenská technická univerzita v Bratislave, 2021.
10. TOMEK, L.; MIKULA, K. Metóda konečných prvkov. 2022. Dostupné tiež z: <https://sites.google.com/site/lukastomekmath/vyucba/mkp?authuser=0>. [Online; dostupné dňa 26-Apríl-2023].
11. TOMKA, M. *Inovatívne spôsoby určovania tiažového poľa Zeme, Bakalárska práca*. Slovenská technická univerzita v Bratislave, 2012.

12. ČUNDERLÍK, R.; MOJZEŠ, M.; MIKULA, K. A comparison of the variational solution of the Neumann geodetic boundary value problem with the geopotential model EGM-96. *Contributions to Geophysics and Geodesy*. 2004, roč. 34, č. 3, s. 209–225. Dostupné tiež z: https://www.math.sk/gbvp/?page_id=56. [Online; dostupné dňa 26-Apríl-2023].