



IEGULDĪJUMS TAVĀ NĀKOTNĒ

ESF projekts

**”Atomāro un nepārtrauktās vides tehnoloģisko fizikālo procesu modelēšana, matemātisko metožu pilnveide un kvalitatīvā izpēte”,  
Nr. 2009/0223/1DP/1.1.1.2.0/09/APIA/VIAA/008**

Tematiskā aktivitāte

**4.2. Tehnoloģisko procesu matemātiskā modelēšana**

Tematiskā apakšaktivitāte

**4.2.3. Magnetoreoloģisko suspensiju īpašību matemātiskā modelēšana izmantojot kombinētu īpašību aprakstu mikro- un makromērogā**

**Programmu kompleksa MRmultiPart dokumentācija.**

# **MRmultiPart - Vairāku patvaļīgas formas feromagnētisku daļiņu mijiedarbības noteikšana ārējā magnētiskajā laukā (programmatūras apraksts).**

Programmatūras apraksts satur informāciju par izmantoto matemātisko modeli, īsu programmēšanas metodiku un galveno lietoto objektu un funkciju (objektu metožu) aprakstu, kā arī ievaddatu un izvaddatu paskaidrojumu.

Programma veidota kā Linux vidē „komandlīnijā” izsaucams izpildāmais fails, līdz ar to sērijveida aprēķinus var veidot kā Linux „čaulas” (*shell*) komanfailus (*script*).

Ievaddatu (galīgo elementu režģa) sagatavošanai papildus nepieciešama no interneta resursiem lejup lādējama brīvā programmatūra GMSH [1,2], kuras iespējas ļauj to lietot arī fizikālo lauku vizualizācijai rezultātu pēcstrādē.

Programmatūras kompilēšanai nepieciešamas UMFPACK (SuiteSparse) un BLAS bibliotēkas.

## Saturs:

1	Programmas mērķis.....	4
2	Programmas matemātiskais modelis .....	4
3	Programmas raksturojums .....	6
3.1	Programmas struktūra .....	7
3.1.1	GEM integrāļu tabulas <i>tetrShapeFunctions_class</i> .....	7
3.1.2	GEM tetraedru režģa apraksts: <i>mesh_MR_tetr</i> .....	8
3.1.3	Lineāru algebrisku vienādojumu sistēmu apraksts: <i>MR_lavs</i> .....	9
3.1.4	Lineāru algebrisku vienādojumu sistēmu apraksts: <i>MR_solver</i> ....	10
3.1.5	Galveināis fails ar main funkciju: <i>MRmultiPart.cc</i> .....	11
4	Programmatūras <i>MRmultiPart</i> kompilēšana.....	11
5	Programmatūras <i>MRmultiPart</i> izmantošana. ....	11
5.1	Ievaddatu – GMSH režģa sagatavošana.....	12
5.2	Ievadfaila sagatavošana .....	18
5.3	Datu saglabāšana .....	20
5.4	Aprēķinu piemērs – spēku izvadfails.....	20
5.5	Aprēķinu piemērs – fizikālie lauki .....	21
6	Programmas modifikācijas .....	22
7	Literatūras saraksts.....	22

## 1 Programmas mērķis

Programmas *MRmultiPart* un tās modifikāciju pamatuzdevums magnētiskās mijiedarbības (spēku un spēku momentu) aprēķins starp magnetizējamu daļiņu pāriem vairāku daļiņu ansablī. No datora resursu šī brīža iespējam saprātīga daļiņu skaita robeža ir 10 vai mazāks skaits daļiņu. Pamatā programma tiek lietota 2, 3 un 4 magnetizējamu daļiņu mijiedarbības aprēķinam, kā daļiņu formu izvēloties sfēras vai cilindriskus stieņus ar noapaļotiem galiem.

Iegūtās magnētiskās mijiedarbības kā funkcijas no daļiņu izvietojuma tālāk tiek lietotas magnētiskās mijiedarbības starp daļiņām magnetoreoloģiskajā suspensijā [3] aprakstam, kā arī novērtēšanai vienkāršotā veidā.

Alternatīva pieeja mijiedarbības aprakstam iespējama konstantas magnētiskās caurlaidības  $\mu$  gadījumā, lietojot analītiskas metodes, kas noved pie multipolu rindām un koeficientu iegūšanas, risinot relatīvi lielas aizpildītas vienādojumu sistēmas. Turklāt konstanta  $\mu$  gadījumā tiek ignorēti magnetizācijas piesātināšanās efekti, kas ir rada neprecizitātes magnetoreoloģiskā efekta aprakstam.

## 2 Programmas matemātiskais modelis

Programma balstās uz skalārā potenciāla izmantošanu Maksvela vienādojumu risināšanā magnetostatikas problēmai ar uzdotu ārējo lauku un nelineāri magnetizējamiem apakšapgabaliem. Spēku aprēķinam tiek izmantota Maksvela spriegumu tenzora pieeja, integrējot pa apakšapgabalus atdalošām virsmām. Zemāk sniegts īss matemātiskā modeļa izklāsts.

Magnetostatisko lauku apraksta divi Maksvela vienādojumi:

$$\vec{\nabla} \cdot \vec{B} = 0, \quad (1)$$

$$\vec{\nabla} \times \vec{H} = 0, \quad (2)$$

kuriem jāpievieno materiālā sakarība

$$\vec{B} = \mu_0 \mu \vec{H}. \quad (3)$$

Nelineāras magnetizācijas gadījumā relatīvā magnētiskā caurlaidība ir funkcija no lauka vielā, atkarībā no lietošanas ērtuma  $\mu = \mu(H)$  vai arī  $\mu = \mu(B)$ .

Trīsdimensionālos aprēķinos ērti lietot skalāro potenciālu  $\Psi$ :

$$\vec{H} = \vec{\nabla} \Psi. \quad (4)$$

Kombinējot (1), (3) un (4), iegūst risināmo vienādojumu

$$\nabla \cdot (\mu \vec{\nabla} \Psi) = -\vec{\nabla} \cdot \vec{J}. \quad (5)$$

Parasti to papildina pirmā veida (Dirihlē) robežnosacījumi (uzdota funkcijas vērtība)

$$\Psi|_{\Gamma_1} = \tilde{\Psi}, \quad (6)$$

vai otrā veida (Neimana) robežnosacījums (uzdots normālais atvasinājums)

$$\vec{n} \cdot \vec{\nabla} \Psi = \tilde{H}. \quad (7)$$

Lai varētu pielietoto Galīgo Elementu Metodi (GEM) [4, 5], stiprais formulējums (5), (6), (7) tiek pārveidots par vājo formulējumu:

$$\int_{\Omega} \mu \vec{\nabla} v \cdot \vec{\nabla} \Psi \, d\Omega = \oint_{\Gamma_2} \mu v \tilde{H} \, d\Gamma, \quad (8)$$

pie nosacījuma, ka

$$v|_{\Gamma_1} = 0, \quad (9)$$

Spēku un spēku momentu aprēķins programmā *MRmultiPart* balstīts uz Maksvela spriegumu tenzora [6]

$$T_{ij} = \frac{1}{2\mu_0} (2 B_i B_j - B^2 \delta_{ij}) \quad (10)$$

izmantošanu, integrējot pa daļiņas norobežojošām virsmām, kurās  $\mu = 1$ .

Spēks:

$$\vec{F} = \oint_{\Gamma} \sum_{i,j=1}^3 T_{ij} n_j \vec{e}_i d\Gamma \quad (10)$$

Spēka moments ( $\varepsilon_{ijk}$  – absolūti antisimetriskais trešā ranga tenzors):

$$\vec{N} = \sum_{i,j,k=1}^3 \left( \oint_{\Gamma} x_i \sum_{m=1}^3 T_{jm} n_m d\Gamma \right) \varepsilon_{ijk} \vec{e}_k \quad (11)$$

Spēku aprēķinu rezultātu precizitāti ietekmējošie faktori aprakstīti darbā [7]. Tā kā lauka aprēķini balstās uz skalārā potenciāla  $\Psi$  iegūšanu, pēc formulas (4) tiek iegūts  $\vec{H}$  vektorlauks, vektorlauku  $\vec{B}$  aprēķina pēc formulas (3) izmantojot relatīvās magnētiskās caurlaidības lauka vērtības. Magnetizāciju iegūst kā

$$\vec{M} = (\mu - 1) \vec{H}. \quad (12)$$

Nelineārās magnetizācijas aprakstam lieto Frēliha-Kennelija likumu

$$\mu(H) - 1 = \frac{M_S(\mu_{ini} - 1)}{M_S + (\mu_{ini} - 1)H} \quad (13)$$

Kas bezdimensionālajam magnētiskajam laukam  $\check{H} = H/M_S$  izsakās kā

$$\mu(\check{H}) - 1 = \frac{(\mu_{ini} - 1)}{1 + (\mu_{ini} - 1)\check{H}} \quad (14)$$

Šeit magnētiskā lauka bezdimensionalizācijai tiek izmantota feromagnētiskā materiāla piesātinājuma magnetizācijas  $M_S$  vērtība.

### 3 Programmas raksturojums

Programmatūra *MRmultiPart* veidota, izmantojot objektorientētu pieeju. Realizācija veikta programmēšanas valodā *C++*. Programmas kompilācijai nepieciešamas brīvpieejas bibliotēkas:

- *UMFPACK* – *unsymmetric multifrontal sparse LU factorization package* [8, 9];
- *BLAS* – *Basic Linear Algebra Subprograms* [10].

Abas bibliotēkas tiek izmantotas reti aizpildītas vienādojumu sistēmas atrisināšanai.

Programmas ievaddatu sagatavošanai nepieciešama brīvā programmatūra GMSH [1,2].

### 3.1 Programmas struktūra

Programmu *MRmultiPart* veido deviņi faili (viens galvenās funkcijas fails un 4 c++ objektu klašu avota un „header” failu pāri):

- *MRmultiPart.cc* – satur C++ programmas *main* funkciju, programmas modifikāciju gadījumā izmaiņas parasti tiek veiktas šai failā;
- *tetrShapeFunctions\_class.cpp* un *tetrShapeFunctions\_class.hpp* – satur formas funkciju integrāļu tabulas un ar lokālām koordinātēm saistītos aprēķinus tetraedra formas 3D galīgajiem elementiem;
  - *mesh\_MR\_tetr.cpp* un *mesh\_MR\_tetr.hpp* – satur galīgo elementu režģa aprakstu un ar to saistītās objektu klases. Failos iekļautie objekti ļauj nodrošināt GMSH vai cita tipa režģa ielasīšanu un informācijas transformāciju konkrētā gadījuma vajadzībām;
- *MR\_lavs.cpp* un *MR\_lavs.hpp* – lineāru algebrisku vienādojumu sistēmu risināšanas objektu klases, pamatfunkcijas ir vienādojumu sistēmu matricas saglabāšana datora atmiņā un UMFPACK bibliotēkas izsaukšanas c++ objekta klases saskarne (*interface*);
- *MR\_solver.cpp* un *MR\_solver.hpp* – matemātiskās problēmas galīgo elementu apraksta komplektācijas objektu klašu faili, nodrošina vienādojumu sistēmas komplektāciju atbilstoši galīgo elementu metodes algoritmam.

#### 3.1.1 GEM integrāļu tabulas *tetrShapeFunctions\_class*

Integrāļu tabulas tiek balstās uz precīzo integrēšanu, tās aizpildās automātiski, veidojot noteiktas kombinācijas formas funkciju un to atvasinājumu reizinājumus. Papildus šajos failos ir objektu klases, kas ļauj aprēķināt skaitliskos (tuvinātos) integrāļus, kā arī objektu klases funkcijas vērtību aprēķinam elementu iekšējos punktos pēc uzdotām koordināšu vērtībām. Galveno tipu, kā arī un objektu klašu hierarhijas saraksts:

- struct tetrShapeFunctionMember
- typedef `std::vector<tetrShapeFunctionMember>`  
tetrShapeFunctionArray;
- class tetrShapeFunctionCombination // patvaljiiga funkcija, kas  
izteikta formas funkcijaas
- typedef `std::vector<tetrShapeFunctionCombination>`  
tetrShapeFunCombArray;
- class tetrShapeFunCombDerivative // funkcijas atvasinaajums, 3  
komp.: d/dL1, d/dL2, d/dL3
- class tetrFunctionsList
- typedef `std::vector<tetrShapeFunCombDerivative>`  
tetrShapeFunCombDerivArray;
- class tetrDerivativeList
- class tetrLinearShapeFunctions : public tetrFunctionsList
- class tetrQuadShapeFunctions : public tetrFunctionsList
- class tetrCustomIntegralTable
- class tetrLinLin\_IntegralTable : public tetrCustomIntegralTable
- class tetrDifLinDifLin\_IntegralTable : public  
tetrCustomIntegralTable
- class tetrQuadQuad\_IntegralTable : public  
tetrCustomIntegralTable
- class tetrDifQuadDifQuad\_IntegralTable : public  
tetrCustomIntegralTable
- class tetrLinDifQuadDifQuad\_IntegralTable : public  
tetrCustomIntegralTable
- class tetrLinElemTables
- class tetrQuadElemTables

### 3.1.2 GEM tetraedru režģa apraksts: *mesh\_MR\_tetr*

Režģa aprakstīšanai nepieciešama noteikta objektu klašu hierarhija, kas ietver mezglu punktus, nogriežņus (malas, šķautnes), trijstūra virsmas elementus un



tetraedra tilpuma elementus, kā arī šo objektu sarakstus, kopumā apraksta tipisku tetraedru režģi. Klašu hierarhijas saraksts:

- class MR\_node3d\_class
- class MR\_node3d\_list\_class
- class MR\_edge\_class
- class MR\_edge\_list\_class
- class MR\_triface\_class
- class MR\_triface\_list\_class
- class MR\_tetrem\_class
- class MR\_tetrem\_list\_class
- class MR\_tetrmesh\_class
- class MR\_neigh\_class
- class MR\_neigh\_list\_class
- class MR\_triface\_sort\_class
- class MR\_triface\_sortlist\_class
- class MR\_triface\_listset\_class

Galvenā izceļamā klašu metode (funkcija) ir

```
void MR_tetrmesh_class :: read_linear_mesh_gmsh(char *filename)
```

Tā nodrošina GMSH lieneāru galīgo elementu režģa ielasīšanu un transformēšanu programmas iekšējā formātā, izveidojot *MR\_tetrmesh\_class* klases objektu.

### **3.1.3 Lineāru algebrisku vienādojumu sistēmu apraksts: *MR\_lavs***

Vienādojumu sistēmas risināšanai tiek lietota UMFPACK metode, kuru realizē atbilstošā bibliotēka komplektā ar BLAS. *MR\_lavs* objektu hierarhija izveido lietotājam ērtu čaulu ap UMFPACK bibliotēkas funkciju. Objektu klašu hierarhijas saraksts:

- class vect\_func
- class CRS\_matrix\_func

- class CCS\_matrix\_func
- class lavs\_UMFPACK

Klases *lavs\_UMFPACK* metode *lavs\_UMFPACK:: Solve()* tiek netieši izsaukta caur *GEM\_Solver :: Solve()*. Metode *lavs\_UMFPACK:: Solve()* izsauc UMFPACK bibliotēkas funkciju

```
UMFpack(size, CCS.col_ptr, CCS.row_ind, CCS.val, xvect, bvect)
```

kur

size : vienādojumu sistēmas izmērs

CCS.col\_ptr, CCS.row\_ind, CCS.val : CCS (*Compressed Column Storage*) formātā [13] saspiesta vienādojumu sistēmas matrica

xvect : vienādojumu sistēmas nezināmo vektors

bvect : vienādojumu sistēmas labās puses vektors

Kļūdas gadījumā tiek atšifrēts kļūdas kods.

### 3.1.4 Lineāru algebrisku vienādojumu sistēmu apraksts: *MR\_solver*

Galīgo elementu metodei atbilstošas vienādojumu sistēmas formēšanas, atrisināšanas un atrisinājuma atpazīšanas fizikālo lauku un interesējošo integrālo lielumu veidā. Satur 2 klases

- class vector\_3D
- class GEM\_Solver

Klase *GEM\_Solver* satur daudzas palīgmetodes (funkcijas), kā galvenās pamata metodes var uzsvērt sekojošās

- void write\_POS\_file raksta fizikālos laukus pēcapstrādes failā, laukus vizualizē ar GMSH
- void create\_LAVS rada vienādojumu sistēmas struktūru atbilstoši ne-nulles elementu izvietojumam matricā

- void Solve risina vienādojumu sistēmu (izsauc UMFPACK)
- double calculate\_H, void calculate\_H\_field, void calculate\_M\_avg, double calc\_mu u.c. aprēķina fizikālos laukus, izejot no  $\Psi$  un  $\mu$  lauku vērtībām
- void calculate\_interaction\_force aprēķina mijiedarbības spēkus un spēku momentus atbilstoši ievaddatos norādītajam.

### 3.1.5 Galveināis fails ar main funkciju: *MRmultiPart.cc*

Šis fails satur tikai galveno funkciju

```
int main(int argc, char *argv[])
```

kuras ķermenī tiek organizēta galveno objekta inicializācija:

```
GEM_Solver problem_2sph;
```

Izmantojot šī objekta metodes, tiek ielasīti ievaddati un organizēts pilns aprēķinu cikls atbilstoši ievaddatu vērtībām, kā arī pieprasīto rezultātu izvads.

## 4 Programmatūras *MRmultiPart* kompilēšana.

Programmatūras kompilēšanai tiek izmantota Code::Blocks (1) vide komplektā ar GCC kompilācijas rīkiem no lietojamās Linux distribūcijas, kas konkrētajā gadījumā ir Ubuntu (2). BLAS un UMFPACK bibliotēkas tiek iegūtas no distribūcijas pakotnēm, GMSH versija 2.5 tiek lejuplādēta no programmatūras interneta vietnes [1].

## 5 Programmatūras *MRmultiPart* izmantošana.

Programma *MRmultiPart* ir piemērota kompilēšanai un lietošanai datoros ar Linux tipa operētājsistēma. Programmas dažādu versiju darbība ir pārbaudīta Linux distribūcijas Ubuntu šā brīža atbalstītajās operētājsistēmas versijās 10.04, 11.04, 11.10, 12.04. Programmas kompilēšanai nepieciešams, lai papildus standarta instalācijai būtu uzinstalētas pakotnes ar:

- *g++*, *C++* valodas kompilatoru (tiek lietota gcc kompilācijas rīku saime);

- *libsuitsparse-dev*, *SuitSparse* bibliotēkas (kas ietver *UMFPACK* programmatūru);
- *libblas-dev*, *BLAS* bibliotēkas.

Programmas statiski kompilēts fails ir palaižams arī uz *Linux* sistēmām, kurām nav instalētas minētās bibliotēkas.

## 5.1 Ievaddatu – GMSH režģa sagatavošana

GMSH režģis tiek gatavots ar GMSH komandfaila (\*.geo) palīdzību. Daži parametri, kā režģa soļa izvēle prasa iepriekšēju pieredzi GEM lietošanā, jo automatizēt režģa izveidi vairāk kā 2 lodveida daļiņām praktiski nav iespējams. Turklāt citas formas daļiņu aplūkošana prasa individuālu pieeju katra režģa ģenerēšanā. Kā paraugs sniegts ģeometrijas fails 2 lodēm bez saskares:

### Ģeometrijas (\*.geo) fails 2 lodēm bez saskares

```
Function lode
  np[0] = p0; Point(np[0]) = {xc, yc, zc, lc};
  np[1] = newp; Point(np[1]) = {xc + r, yc, zc, lc};
  np[2] = newp; Point(np[2]) = {xc - r, yc, zc, lc};
  np[3] = newp; Point(np[3]) = {xc, yc + r, zc, lcn};
  np[4] = newp; Point(np[4]) = {xc, yc - r, zc, lcn};
  np[5] = newp; Point(np[5]) = {xc, yc, zc + r, lc};
  np[6] = newp; Point(np[6]) = {xc, yc, zc - r, lc};

  c[0] = p0; Circle(c[0]) = {np[6], np[0], np[1]};
  c[1] = newl; Circle(c[1]) = {np[1], np[0], np[5]};
  c[2] = newl; Circle(c[2]) = {np[5], np[0], np[2]};
  c[3] = newl; Circle(c[3]) = {np[2], np[0], np[6]};
  c[4] = newl; Circle(c[4]) = {np[2], np[0], np[3]};
  c[5] = newl; Circle(c[5]) = {np[3], np[0], np[1]};
  c[6] = newl; Circle(c[6]) = {np[1], np[0], np[4]};
  c[7] = newl; Circle(c[7]) = {np[4], np[0], np[2]};
  c[8] = newl; Circle(c[8]) = {np[5], np[0], np[3]};
  c[9] = newl; Circle(c[9]) = {np[3], np[0], np[6]};
  c[10] = newl; Circle(c[10]) = {np[6], np[0], np[4]};
  c[11] = newl; Circle(c[11]) = {np[4], np[0], np[5]};

  ll[0] = newll; Line Loop(ll[0]) = {c[5], -c[0], -c[9]};
  s[0] = news; Ruled Surface(s[0]) = {ll[0]};
  ll[1] = newll; Line Loop(ll[1]) = {c[5], c[1], c[8]};
  s[1] = news; Ruled Surface(s[1]) = {ll[1]};
  ll[2] = newll; Line Loop(ll[2]) = {c[8], -c[4], -c[2]};
  s[2] = news; Ruled Surface(s[2]) = {ll[2]};
  ll[3] = newll; Line Loop(ll[3]) = {c[4], c[9], -c[3]};
  s[3] = news; Ruled Surface(s[3]) = {ll[3]};
  ll[4] = newll; Line Loop(ll[4]) = {c[6], c[11], -c[1]};
  s[4] = news; Ruled Surface(s[4]) = {ll[4]};
```

## Ģeometrijas (\*.geo) fails 2 lodēm bez saskares

```
ll[5] = newll; Line Loop(ll[5]) = {c[11],c[2],-c[7]};
s[5] = news; Ruled Surface(s[5]) = {ll[5]};
ll[6] = newll; Line Loop(ll[6]) = {c[7],c[3],c[10]};
s[6] = news; Ruled Surface(s[6]) = {ll[6]};
ll[7] = newll; Line Loop(ll[7]) = {c[10],-c[6],-c[0]};
s[7] = news; Ruled Surface(s[7]) = {ll[7]};
Surface Loop(sl) = {s[]};
Volume(vol) = {sl};
Physical Volume(pvol) = {vol};
Return
r = 1;
///<#r=1#Lodes radiusss
xc = 0;
zc = 0;
dr = 0.02*r;
///<#dr=r#Attalums starp lodem

// lode 1
yc = r + dr/2;
lc = r*0.1;
///<#lc=r*0.1#Rezgja solis uz lodeem
lcn = r*0.01;
///<#lc=r*0.1#Rezgja solis "centraa"
p0=10;
s1=211;
vol=311;
pvol = 11;
Call lode;

// lode 2
p0=50;
s1=221;
vol=321;
yc = -yc;
pvol = 12;
Call lode;

// aptverosa kaste
att = 5*r;
///<#att = 2*r#Attalums no lodes liidz aptveroshajai kastei
tmp = att + r;

alc = r*0.5;
///<#alc = r#Rezgja solis uz aptveroshaas kastes

ap[1] = newp; Point(ap[1]) = {0, 0, 0, lcn};
ap[2] = newp; Point(ap[2]) = {-tmp, 0, -tmp, alc};
ap[3] = newp; Point(ap[3]) = {tmp, 0, -tmp, alc};
ap[4] = newp; Point(ap[4]) = {-tmp, 0, tmp, alc};
ap[5] = newp; Point(ap[5]) = {tmp, 0, tmp, alc};

al[1] = newl; Line(al[1]) = {ap[2], ap[3]};
al[2] = newl; Line(al[2]) = {ap[3], ap[5]};
```

## Ģeometrijas (\*.geo) fails 2 lodēm bez saskares

```
al[3] = newl; Line(al[3]) = {ap[5], ap[4]};
al[4] = newl; Line(al[4]) = {ap[4], ap[2]};
al[5] = newl; Line(al[5]) = {ap[1], ap[2]};
al[6] = newl; Line(al[6]) = {ap[1], ap[3]};
al[7] = newl; Line(al[7]) = {ap[1], ap[4]};
al[8] = newl; Line(al[8]) = {ap[1], ap[5]};
all[1] = newll; Line Loop(all[1]) = {al[5], al[1], -al[6]};
all[2] = newll; Line Loop(all[2]) = {al[6], al[2], -al[8]};
all[3] = newll; Line Loop(all[3]) = {al[8], al[3], -al[7]};
all[4] = newll; Line Loop(all[4]) = {al[7], al[4], -al[5]};
as[1] = news; Plane Surface(as[1]) = {all[1]};
as[2] = news; Plane Surface(as[2]) = {all[2]};
as[3] = news; Plane Surface(as[3]) = {all[3]};
as[4] = news; Plane Surface(as[4]) = {all[4]};

tmp2 = att + 2.5*r;
ap[15] = newp; Point(ap[15]) = {-tmp, tmp2, -tmp, alc};
ap[16] = newp; Point(ap[16]) = {tmp, tmp2, -tmp, alc};
ap[17] = newp; Point(ap[17]) = {-tmp, tmp2, tmp, alc};
ap[18] = newp; Point(ap[18]) = {tmp, tmp2, tmp, alc};

al[15] = newl; Line(al[15]) = {ap[15], ap[16]};
al[16] = newl; Line(al[16]) = {ap[16], ap[18]};
al[17] = newl; Line(al[17]) = {ap[18], ap[17]};
al[18] = newl; Line(al[18]) = {ap[17], ap[15]};
all[5] = newll; Line Loop(all[5]) = {al[15], al[16], al[17], al[18]};
as[5] = news; Plane Surface(as[5]) = {all[5]};

ap[19] = newp; Point(ap[19]) = {-tmp, -tmp2, -tmp, alc};
ap[20] = newp; Point(ap[20]) = {tmp, -tmp2, -tmp, alc};
ap[21] = newp; Point(ap[21]) = {-tmp, -tmp2, tmp, alc};
ap[22] = newp; Point(ap[22]) = {tmp, -tmp2, tmp, alc};

al[19] = newl; Line(al[19]) = {ap[19], ap[20]};
al[20] = newl; Line(al[20]) = {ap[20], ap[22]};
al[21] = newl; Line(al[21]) = {ap[22], ap[21]};
al[22] = newl; Line(al[22]) = {ap[21], ap[19]};
all[6] = newll; Line Loop(all[6]) = {al[19], al[20], al[21], al[22]};
as[6] = news; Plane Surface(as[6]) = {all[6]};

al[23] = newl; Line(al[23]) = {ap[15], ap[2]};
al[24] = newl; Line(al[24]) = {ap[16], ap[3]};
al[25] = newl; Line(al[25]) = {ap[17], ap[4]};
al[26] = newl; Line(al[26]) = {ap[18], ap[5]};

al[27] = newl; Line(al[27]) = {ap[19], ap[2]};
al[28] = newl; Line(al[28]) = {ap[20], ap[3]};
al[29] = newl; Line(al[29]) = {ap[21], ap[4]};
al[30] = newl; Line(al[30]) = {ap[22], ap[5]};

all[7] = newll; Line Loop(all[7]) = {al[23], al[1], -al[24], -
al[15]};
as[7] = news; Plane Surface(as[7]) = {all[7]};

all[8] = newll; Line Loop(all[8]) = {al[24], al[2], -al[26], -
al[16]};
```

### Ģeometrijas (\*.geo) fails 2 lodēm bez saskares

```
as[8] = news; Plane Surface(as[8]) = {all[8]};

all[9] = newll; Line Loop(all[9]) = {al[26], al[3], -al[25], -
al[17]};
as[9] = news; Plane Surface(as[9]) = {all[9]};

all[10] = newll; Line Loop(all[10]) = {al[25], al[4], -al[23], -
al[18]};
as[10] = news; Plane Surface(as[10]) = {all[10]};

all[11] = newll; Line Loop(all[11]) = {al[27], al[1], -al[28], -
al[19]};
as[11] = news; Plane Surface(as[11]) = {all[11]};

all[12] = newll; Line Loop(all[12]) = {al[28], al[2], -al[30], -
al[20]};
as[12] = news; Plane Surface(as[12]) = {all[12]};

all[13] = newll; Line Loop(all[13]) = {al[30], al[3], -al[29], -
al[21]};
as[13] = news; Plane Surface(as[13]) = {all[13]};

all[14] = newll; Line Loop(all[14]) = {al[29], al[4], -al[27], -
al[22]};
as[14] = news; Plane Surface(as[14]) = {all[14]};

Physical Surface(1) = {as[5], as[6], as[7], as[8], as[9], as[10],
as[11], as[12], as[13], as[14]};
Physical Surface(10) = {as[1], as[2], as[3], as[4]};

// telpa ap lodi 1
Surface Loop(212) = {as[1], as[2], as[3], as[4], as[5], as[7], as[8],
as[9], as[10]};
Volume(312) = {211, 212};
Physical Volume(1) = {312};

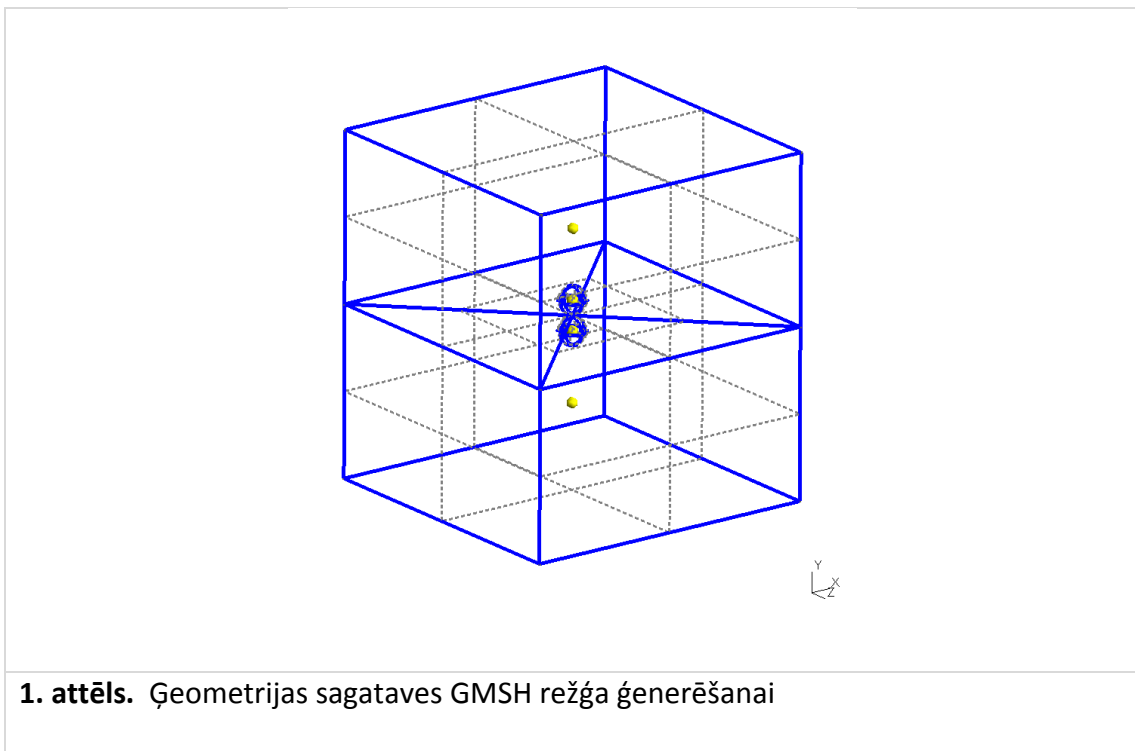
// lode 2
Surface Loop(222) = {as[1], as[2], as[3], as[4], as[6], as[11],
as[12], as[13], as[14]};
Volume(322) = {221, 222};
Physical Volume(2) = {322};
```

Aprēķināmo objektu ģeometriju var veidot vai nu interaktīvi GMSH vidē, vai arī rakstot \*.geo komandfailu. Kopumā pieeju var raksturot kā *bottom-up* procesu:

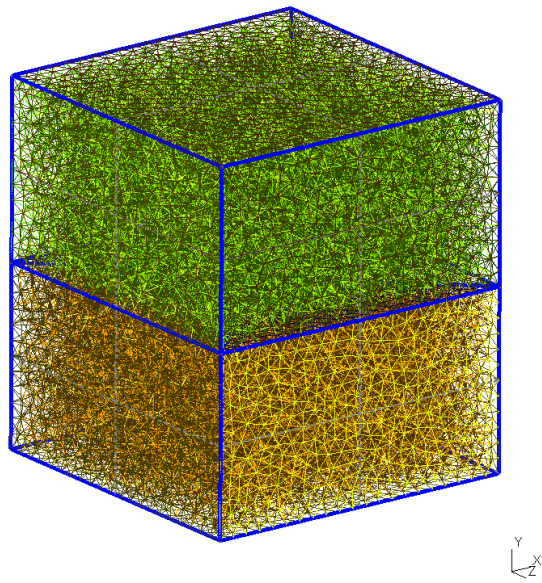
- 1) Izveido punktus (Point)
- 2) Izveido nogriežņus un arkas (Line, Circle), balstoties uz punktiem.
- 3) Izveido līniju cilpas (Line Loop), balstoties uz nogriežņiem un arkām.

- 4) Izveido plakanas vai izliektas virsmas (Plane Surface, Ruled Surface), balstoties uz līniju cilpām.
- 5) Izveido tilpumus (Volume), balstoties uz tos aptverošām virsmām
- 6) Izveido speciālas iezīmes (Physical Surface) virsmām, kurām uzdos robežnosacījumus vai arī uz kurām rēķinās spēkus un spēku momentus
- 7) Izveido speciālas iezīmes (Physical Volume) viena materiāla apgabaliem
- 8) Ar GMSH komandām izveido 3D režģi un veic tā optimizāciju
- 9) Saglabā režģi failā.

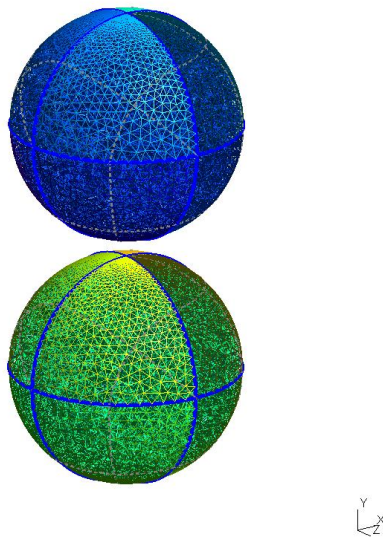
Pēc komandfaila apstrādes GMSH programmatūra no \*.geo faila (1.attēls) izveido tetraedru režģi (2.attēls).







**2. attēls.** No ģeometrijas sagatavēm izveidotais GMSH tetraedru režģis aprēķiniem ar GEM



**3. attēls.** No ģeometrijas sagatavēm izveidotais GMSH tetraedru režģis aprēķiniem ar GEM, palielināts ložu režģis.

## 5.2 Ievadfaila sagatavošana

Ērtības dēļ parasti tiek lietots ievadfails ar fiksētu vārdu „input.dat”. Nepieciešamas gadījumā to var mainīt programmas tekstā, bet tad nepieciešama kompilācija. Sērijveida aprēķiniem var tikt lietotas modificētas programmas versijas ar komandlīnijas parametru izmantošanu.

Ievadfails „input.dat”	
Lodes2.msh	<- rezhgja fails
rezultati	<- izvaddatu faila nosaukuma „baaze”
0.5	<- h0 bezdimensionaali
1.0e-6	<- precizitaate
0.25	<- mu relaksaaciju parametrs
2	<- apakshapgabalu skaits
1 1000 1	<- 1.apg.: mrk, mu_ini, M_s
2 1 0	<- 2.apg.: mrk, mu_ini, M_s
3	<- virsmu skaits, kur uzdod 1.veida RN (Psi=r.H0)
2 5 6	<- virsmu markjeri, skaits atbilst uzdotajam
2	<- virsmu skaits, kur reekjina speaku
7 8	<- virsmu markjeri, kur reekjina speaku
1	<- Vai laukus rēķināt (0 - nee / 1- jaa)
1	<- H vektorlauks
0	<- B vektorlauks
0	<- M vektorlauks
0	<- mu skalaarais lauks
1	<- H  skalaarais lauks
1	<- B  skalaarais lauks
1	<- M  skalaarais lauks

Ievadfailā katrā teksta rindiņā pirmā daļa līdz tukšumam ir izmantojamie dati, atlikusī daļa ir komentārs, kas tiek ignorēts. Izņēmums ir atsevišķas rindiņas ar noteiktu vērtību skaitu, kas visas izvietotas vienā rindiņā un atdalītas ar tukšumiem. Ievadfaila rindiņu skaits ir fiksēts.

Ievadfaila formāta atšifrējums		
Nr	Piemērs	Atšifrējums
1	Lodes2.msh	Režģa faila vārds, režģis tiek sagatavots ar GMSH programmatūru
2	rezultati	Izvadfailu „bāzes” daļa, piemēram, „rezultati” atbilst izvadfailu vardi „rezultāti.force” un „rezultāti.pos”
3	0.5	Bezdimensionāls ārējais lauks $H_0$ , normēts pēc piesātinājuma magnetizācijas

4	1.0e-6	Precizitātes robeža aprēķiniem. Uzdod maksimālo relatīvās magnētiskās caurlaidības $\mu$ izmaiņu pa visu aprēķinu apgabalu vienā iterācijā.
5	0.25	$\mu$ relaksācijas parametrs iteratīvajā procesā
6	2	Apakšapgabalu skaits (ar atšķirīgām magnētiskajām īpašībām)
7	1 1000 1	1. Apakšapgabals, tā $\mu_{ini} = 1000$ un $M_s = 1$ vērtības
8	2 1 0	2. Apakšapgabals, tā $\mu_{ini} = 1$ un $M_s = 0$ vērtības
9	3	Virsmu skaits, kurām uzdod 1. Veida robežnosacījumu, saistītu ar ārējā lauka potenciāla $\Psi_0 = \vec{H}_0 \cdot \vec{r}$ vērtības uzdošanu.
10	2 5 6	Atbilstoši virsmu skaitam dotas (ar tukšumiem atdalītas) virsmu marķieru vērtības. Režģa failā atbilst fizisko virsmu marķieru numuriem.
11	2	Virsmu skaits, kurām rēķina spēkus un spēku momentus.
12	7 8	Atbilstoši virsmu skaitam dotas (ar tukšumiem atdalītas) virsmu marķieru vērtības. Režģa failā atbilst fizisko virsmu marķieru numuriem.
13	1	Norāde, vai rēķināt fiziskos laukus un rakstīt lauku izvadfailu „rezultāti.pos”. Vērtība 0 - nerēķināt, vērtība 1 - rēķināt. Tālāk seko individuāli slēdzīši katram no laukiem.
14	1	Vai rēķināt magnētiskās intensitātes vektorlauku $\vec{H}$ . Vērtība 0 - nerēķināt, vērtība 1 - rēķināt
15	0	Vai rēķināt magnētiskās indukcijas vektorlauku $\vec{B}$ . Vērtība 0 - nerēķināt, vērtība 1 - rēķināt
16	0	Vai rēķināt magnetizācijas vektorlauku $\vec{M}$ . Vērtība 0 - nerēķināt, vērtība 1 - rēķināt
17	0	Vai rēķināt relatīvās magnētiskās caurlaidības $\mu$ skalāro lauku. Vērtība 0 - nerēķināt, vērtība 1 - rēķināt
18	1	Vai rēķināt magnētiskās intensitātes moduļa $ \vec{H} $ skalāro lauku. Vērtība 0 - nerēķināt, vērtība 1 - rēķināt
19	1	Vai rēķināt magnētiskās indukcijas moduļa $ \vec{B} $ skalāro lauku. Vērtība 0 - nerēķināt, vērtība 1 - rēķināt
20	1	Vai rēķināt magnetizācijas moduļa $ \vec{M} $ skalāro lauku. Vērtība 0 - nerēķināt, vērtība 1 - rēķināt

Komentārs magnetizācijas datiem apakšapgabalos: vērtības tā  $\mu_{ini} = 1000$  un  $M_s = 1$  nozīmē, ka tiks lietots Frēliha-Kennelija magnetizācijas likums (14) ar atbilstošiem parametriem. Ja  $\mu_{ini} = 1000$  un  $M_s = 0$ , tad tas nozīmē, ka lietos konstantu relatīvās magnētiskās caurlaidības vērtību  $\mu = \mu_{ini}$  (ja uzdod  $M_s = 0$ ).

### 5.3 Datu saglabāšana

Tiek veidoti divi izvaddatu faili. Vienmēr tiek radīts fails ar nosaukumu „bāze”.force, kur faila vārda daļu „bāze” ņem no ievaddatos norādītās informācijas. Ja ievadfailā ir norādīts, ka jāaprēķina arī fizikālie lauki, tad tiek radīts fails „bāze”.pos. Faila „bāze”.pos ievieto fizikālos laukus, kas norādīti ievadfailā, to vizualizāciju un veic ar GMSH. GMSH vide ļauj veidot tālāku fizikālo lauku apstrādi, parādot dažādus šķēlumus un laukus uz apgabalu virsmām (detaljas meklēt GMSH instrukcijās [1]).

### 5.4 Aprēķinu piemērs – spēku izvadfails

Spēka izvadfaila piemērs „rezultati.force” sniegts zemāk:

Spēku izvadfails „rezultati.force”	
h0=0.01	fi=0
1	3.4052
2	0.108659
3	0.00325423
4	3.51719e-06
Force1.x = -2.45902e-06 Force1.y = 0.00545978 Force1.z = -2.55839e-06	
Force2.x = -3.56176e-06 Force2.y = 0.00554571 Force2.z = -3.00563e-06	
Moment1.x = -2.45902e-06 Moment1.y = 0.00545978 Moment1.z = -2.55839e-06	
Moment2.x = -3.56176e-06 Moment2.y = 0.00554571 Moment2.z = -3.00563e-06	
Magn1.x = 2.74855e-07 Magn1.y = 0.0478523 Magn1.z = -3.89055e-05	
Magn2.x = 4.24444e-05 Magn2.y = 0.0478588 Magn2.z = -1.32295e-05	

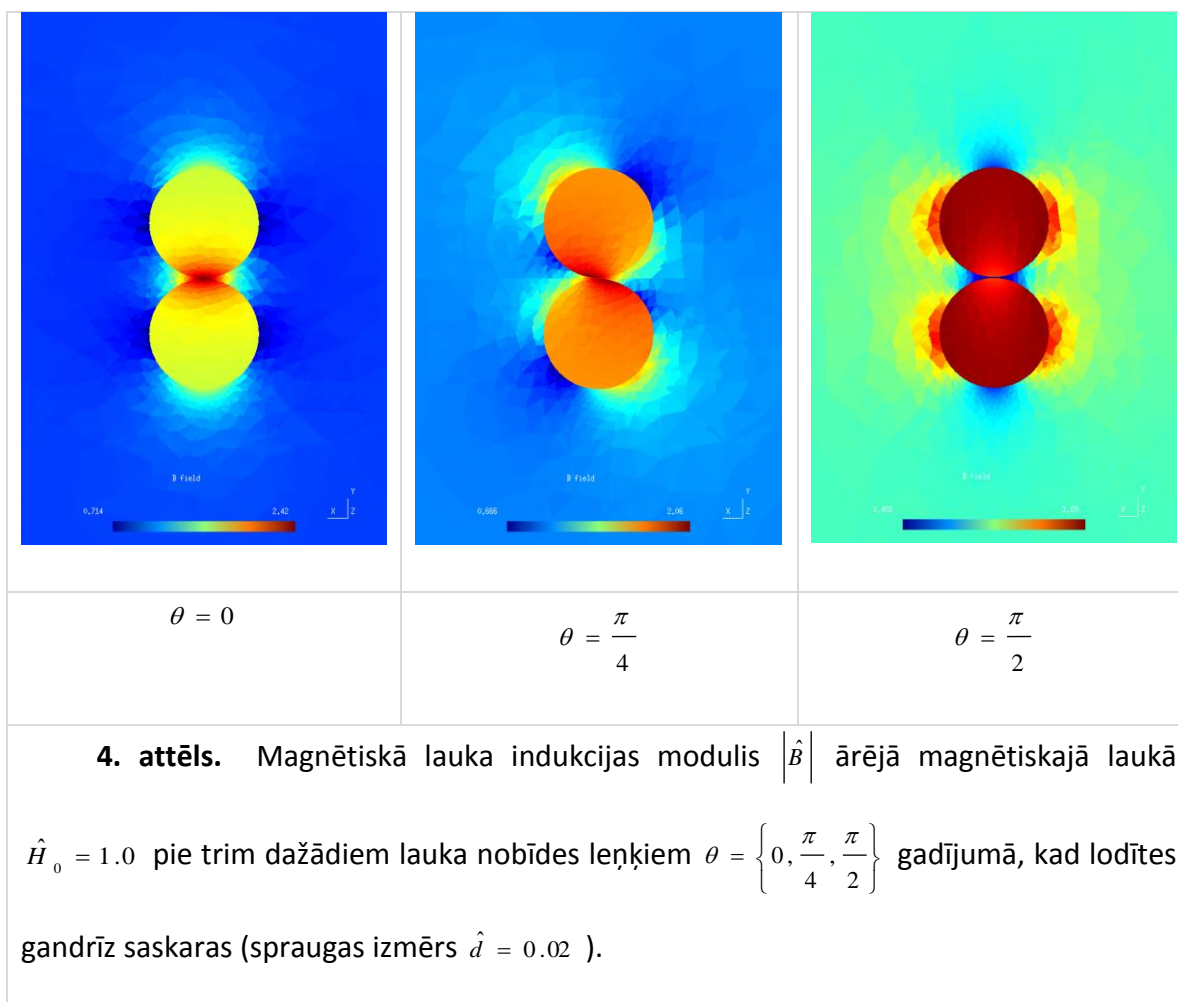
Tajā dotas magnētiskā lauka stipruma ( $h_0$ ) un virziena leņķa ( $fi$ ) vērtības. Tālāk seko sanumurētas kārtējās iterācijās sasniegtās „precizitātes” vērtības (skatīt komentārus par ievadfailu). Pēc šīm vērtībām var izsekot, vai tika sasniegta pieprasītā precizitāte.

Tad seko spēka un spēka momentu vektoru komponentu vērtības (piemēram, Force2.x, Force2.y, Force2.z, Moment1.x, Moment1.y, Moment1.z) atbilstoši uzdoto aprēķinu virsmu skaitam. Kontroles nolūkos iekšējām virsmām tiek dotas divas vērtības.

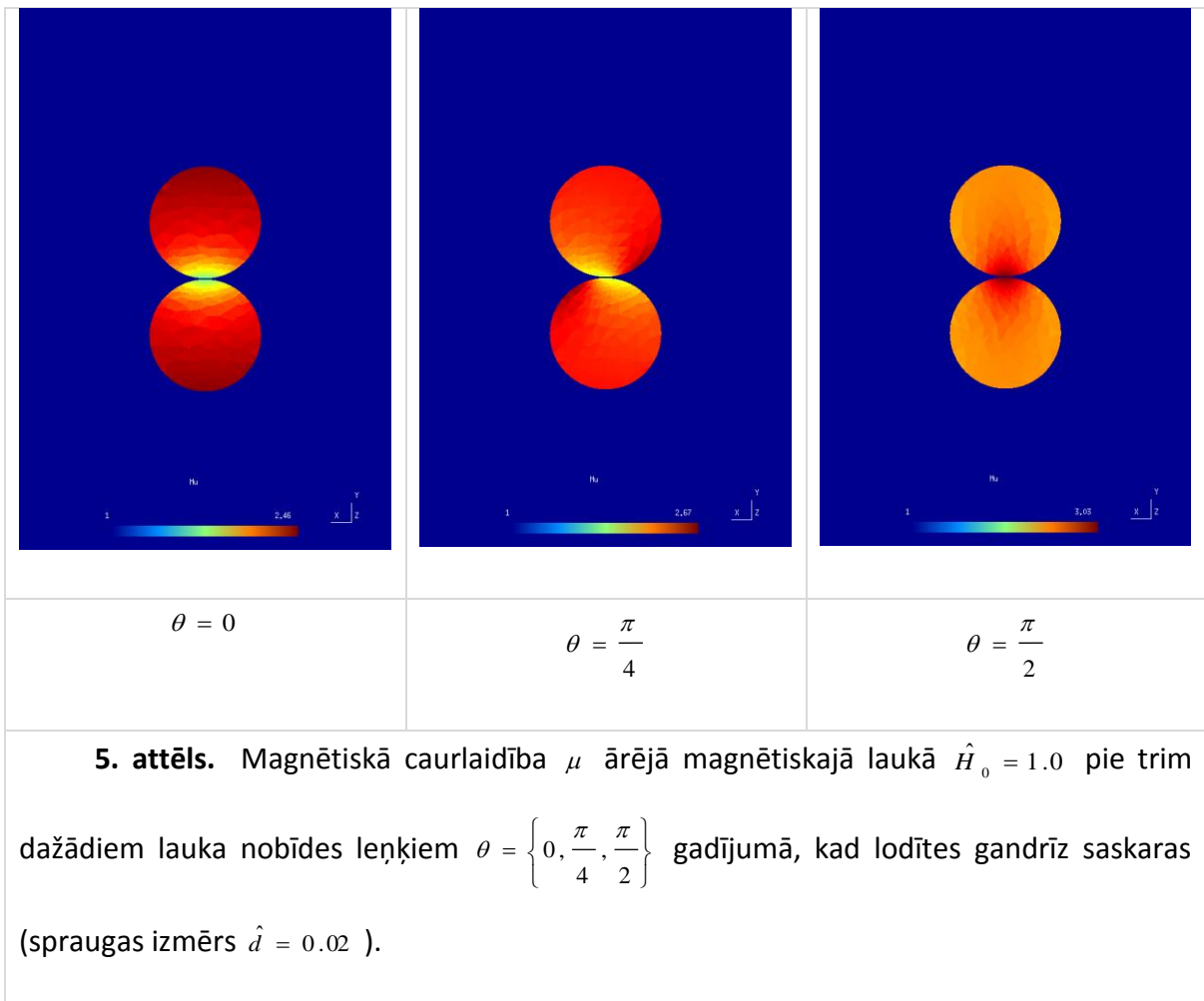
Tāpat apgabaliem ar  $\mu > 1$  tiek aprēķinātas vidējās magnetizācijas vērtības un izvadītas atbilstošajās faila rindiņās (piemēram, Magn1.x, Magn1.y, Magn1.z).

## 5.5 Aprēķinu piemērs – fizikālie lauki

Fizikālo lauku vizualizācijai tiek radīts fails GMSH „MSH ASCII file format” datu formātā, ar faila formāta detaļām var iepazīties [1]. Iespēju demonstrācijai šai atskaitē kalpo 4.attēls un 5.attēls.



4.attēlā parādīts magnētiskā lauka sadalījums pie dažādām ārējā lauka orientācijām (0, 45 un 90 grādi pret simetrijas asi). Lauks parādīts plaknē, ko veido simetrijas ass divām sfērām un lauka virziens. Pie  $\theta = 0$  labi saskatāma ārējā lauka koncentrācija kontaktzonā. 5.attēlā identiskās situācijās parādīta ložu materiāla magnētiskā caurlaidība, demonstrējot nelineārās magnetizācijas efektu, kas spilgti izpaušas pie  $\theta = 0$ .



## 6 Programmas modifikācijas

Programmai *MRmultiPart* eksistē šaurāka pielietojuma versijas sērijveida aprēķiniem, bet tās katrā atsevišķā gadījumā tiek izveidotas kā ļoti specifiska individualizēta modifikācija, tāpēc šeit netiek aplūkotas.

## 7 Literatūras saraksts

1. Programmatūra GMSH v.2.5. (Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities). [Tiešsaiste]: <http://geuz.org/gmsh/>
2. C. Geuzaine and J.-F. Remacle, Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities,

- International Journal for Numerical Methods in Engineering 79(11), pp. 1309–1331, 2009.
3. G.Bossis, O.Volkova, S.Lācis, A.Meunier , Magnetorheology: Fluids, Structures and Rheology , Lecture Notes in Physics, Vol. 594, 2003, pp.202-230
  4. Jian-Ming Jin The Finite Element Method in Electromagnetics, Wiley-IEEE Press, 2002
  5. S.Lācis, O.Volkova, G.Bossis Application of FEM for magnetic force calculation between ferromagnetic spheres, Proceedings of the International Colloquium “Modelling of Material Processing”, Riga, May 28-29, 1999, pp.222-227
  6. J.D. Jackson, Classical Electrodynamics, 3rd edition, Wiley, 1998
  7. F. Henrotte, G. Deliege, K. Hameyer The eggshell approach for the computation of electromagnetic forces in 2D and 3D. COMPEL, Vol.23, 2004, pp.996-1005
  8. UMFPACK, [Tiešsaiste]: <http://www.cise.ufl.edu/research/sparse/umfpack/>
  9. T. A. Davis, A column pre-ordering strategy for the unsymmetric-pattern multifrontal method, ACM Transactions on Mathematical Software, vol 30, no. 2, June 2004, pp. 165-195. T. A. Davis, Algorithm 832: UMFPACK, an unsymmetric-pattern multifrontal method, ACM Transactions on Mathematical Software, vol 30, no. 2, June 2004, pp. 196-199. T. A. Davis and I. S. Duff, A combined unifrontal/multifrontal method for unsymmetric sparse matrices, ACM Transactions on Mathematical Software, vol. 25, no. 1, pp. 1-19, March 1999. T. A. Davis and I. S. Duff, An unsymmetric-pattern multifrontal method for sparse LU factorization, SIAM Journal on Matrix Analysis and Applications, vol 18, no. 1, pp. 140-158, Jan. 1997.
  10. BLAS (Basic Linear Algebra Subprograms). [Tiešsaiste]: <http://www.netlib.org/blas/>

11. Code::Blocks. The open source, cross platform, free C++ IDE. [Tiešsaiste]: <http://www.codeblocks.org/>.
12. Ubuntu. [Tiešsaiste]: <http://www.ubuntu.com>
13. Compressed Column Storage (CCS), Compressed Row Storage (CRS) [Tiešsaiste]: <http://netlib.org/utk/papers/templates/node92.html>; R. Barrett, M. Berry, T. F. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, H. Van der Vorst, Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods, SIAM, 1994