



**LATVIJAS
UNIVERSITĀTE**
ANNO 1919



LATVIJAS UNIVERSITĀTE
FIZIKAS UN MATEMĀTIKAS FAKULTĀTE
FIZIKAS NODAĻA

**LĀDĒTU DAĻINU KUSTĪBAS
MAGNĒTISKĀ LAUKĀ 3D
MODELEŠANA OPENFOAM VIDĒ**

BAKALAURA DARBS

Autors: **Juris Venčels**

Studenta apliecības Nr.: jv10070

Darba vadītājs: Dr.phys. Kaspars Lācis

RĪGA 2013

Anotācija

Darbā tiek aplūkota aprēķinu programmas izveide un tās funkcionalitāte elektriskā un magnētiskā lauka modelēšanai trijās dimensijās, izmantojot magnētiskā vektorpotenciāla “ A , $V-A$ ” formulējumu. Programma veidota uz atvērtā koda bibliotēkas **OpenFOAM** bāzes. Rezultāti salīdzināti ar programmas FEMM aksiāl-simetriskajiem aprēķiniem. Programma nodrošina multi-reģionu aprēķinus, iespēju vienlaicīgi modelēt dažādus procesus, uzdot fizikālu lielumu atkarību no citiem lielumiem.

Programma modelē elektriski lādētu un neitrālu daļiņu 3D kustību elektriskajā un magnētiskajā laukā, izmantojot DSMC un PIC formulējumus. Daļiņu trajektorijas salīdzinātas ar citu formulējumu atrisinājumiem.

Programma paredzēta plazmas modelēšanai magnetronu uzputināšanas procesā, dažādu magnetronu sistēmu izpētē un optimizēšanā.

Atslēgvārdi: OpenFOAM, magnētiskais lauks, plazma, DSMC, PIC

Annotation

Work deals with design and functionality of program for modelling electric and magnetic fields in 3D by using magnetic vector potential “ A , $V-A$ ” formulation. Program is built on open source library **OpenFOAM**. Results were compared with the results of program **FEMM** in the axial-symmetrical case. Program provides multi-region calculations to simulate various processes at the same time and considers dependence of physical properties on other variables.

Program simulates 3D motion of electrically charged and neutral particles in electric and magnetic fields using the **DSMC** and **PIC** formulations. Particle trajectories are compared with solutions from other formulations.

Program is designed to model plasma in magnetron sputtering process, explore and optimize different magnetron systems.

Keywords: **OpenFOAM**, magnetic field, plasma, **DSMC**, **PIC**

Saturs

1 Ievads	3
1.1. Darba mērķis un uzdevumi	4
1.2. Iestrādes, līdzīgu pētījumu un literatūras apskats	5
1.2.1. Programmas pamats, ģeometrijas un režģa ģenerācija	5
1.2.2. Magnētiskā lauka modelēšana	5
1.2.3. Retinātas plazmas modelēšana	6
2 Geometrijas un režģa ģenerācija	7
2.1. Pirms un pēc-apstrādes programma Salome	7
2.1.1. Ģeometrijas veidošana	7
2.1.2. Ģeometrijas eksportēšana uz OpenFOAM	8
2.2. Režģa ģenerācijas programmas blockMesh un snappyHexMesh	9
2.2.1. Režģa ģenerācija	9
2.3. Norādījumi	10
3 Programmas uzbūve	11
3.1. Aprēķinu programma multiRegionPlasmaFoam	11
3.1.1. Failu un mapju struktūra	11
3.1.2. Veiktie programmas pārveidojumi	12
3.2. Aprēķinu piemērs multiRegionPlasmaTest	13
4 Magnētiskā lauka modelēšana	15
4.1. Vispārīgas elektromagnētisma sakarības	15
4.2. \mathbf{A} , V - \mathbf{A} (vektor-potenciāla) formulējums	18
4.2.1. Bezgalīga apgabala robežnosacījumi	19
4.2.2. Robežnosacījumi uz apgabalu robežvirsmām	29
4.2.3. Magnētiskās caurlaidības nelinearitāte	33
4.2.4. Jauktie robežnosacījumi uz apgabalu robežvirsmām	36
4.2.5. \mathbf{A} aprēķins materiāliem ar nehomogēnu μ	42
4.3. Citi formulējumi	42
4.3.1. \mathbf{A} , V - Ψ (skalārā potenciāla) formulējums	42
4.3.2. T - Ω , Ω formulējums	44
5 Retinātas plazmas modelēšana	47
5.1. Elektriski neitrālas gāzes modelēšana	47
5.1.1. Skaitliskās hidrodinamikas (CFD) metode	47
5.1.2. Tiešās simulācijas Montekarlo (DSMC) metode	48
5.2. Lādētu daļiņu un elektromagnētiskā lauka mijiedarbības modelēšana	51

5.2.1. Particle-In-Cell (PIC) metode	52
5.2.2. Vienkāršota PIC realizācija programmā <code>multiRegionPlasmaFoam</code> .	55
5.3. Rezultāti	56
5.3.1. Lādētu daļiņu dinamika ārējā elektromagnētiskajā laukā	56
5.3.2. Lādētu daļiņu savstarpēja mijiedarbība	56
5.3.3. Elektronu dinamika magnetronu sistēmā	61
6 Secinājumi	63
7 Pateicība	65
8 Pielikums	69
8.1. Python programma ģeometrijas apstrādei un virsmas režģu ģenerācijai programmā <code>Salome</code>	69
8.2. Bash skripts <code>.stl</code> failu apvienošanai	70
8.3. <code>multiRegionPlasmaFoam.C</code> faila saturs	71
8.4. OpenFOAM failu saturs μ atkarības no B uzdošanai	72

Apzīmējumu saraksts

OpenFOAM dimensiju vektors [1 2 3 4 5 6 7]

Nr.	Lielums	SI
1	Masa	kilograms [kg]
2	Garums	metrs [m]
3	Laiks	sekunde [s]
4	Temperatūra	kelvins [K]
5	Vielas daudzums	kilomols [kgmol]
6	Strāva	ampērs [A]
7	Gaismas intensitāte	kandela [cd]

Apzīmējumi

		OpenFOAM	Lielsums
A	$\frac{V \cdot s}{m}$	A [1 1 -2 0 0 -1 0]	magnētiskais vektorpotenciāls
B	T	B [1 0 -2 0 0 -1 0]	magnētiskā lauka indukcijas vektors
D	$\frac{C}{m^2}$	netiek lietots	
E	$\frac{V}{m}$	E [1 1 -3 0 0 -1 0]	elektriskā lauka intensitātes vektors
H	$\frac{A}{m}$	H [0 -1 0 0 0 1 0]	magnētiskā lauka intensitātes vektors
M	$\frac{A}{m}$	Hc [0 -1 0 0 0 1 0]	magnetizācijas vektors
J	$\frac{A}{m^2}$	J [0 -2 0 0 0 1 0]	strāvas blīvums
J_M	$\frac{A}{m^2}$	tieki lietots curl(Hc)	
ϕ	V	phi [1 2 -3 0 0 -1 0]	skalārais elektriskais potenciāls
σ	$\Omega^{-1} m^{-1}$	sigma [-1 -3 3 0 0 2 0]	elektriskā caurlaidība
μ		mu [0 0 0 0 0 0 0]	relatīvā magnētiskā caurlaidība
μ_0	$1.26 \cdot 10^{-6} \frac{H}{m}$	mu0 [1 1 -2 0 0 -2 0]	magnētiskā konstante
ϵ_0	$8.85 \cdot 10^{-12} \frac{F}{m}$	epsilon0 [-1 -3 4 0 0 2 0]	elektriskā konstante
ρ	C	rhoQ [0 0 1 0 0 -2 0]	elektriskā lādiņa blīvums

Termini un to skaidrojumi

Aprēķinu programma - programma, kas, izmantojot OpenFOAM programmas bibliotēkas, veic skaitliskus aprēķinus *aprēķinu piemēriem*. Programma satur komandas lauku apstrādei, diferenciālvienādojumu diskretizācijai, to risināšanai, kā arī robežnosacījumu aproksimācijai, aprēķiniem utt.

Apgabals - virtuāls ģeometriskais objekts, kas raksturo reāla ķermenē vai apgabala fizikālās īpašības un ārējo virsmu.

Aprēķinu lauks - apgabala mainīgo vektoriālie un skalārie lauki.

Aprēķinu piemērs - failu un mapju kopa, kā arī skripti aprēķinu programmas inicializācijai. Satur informāciju par modeļa *ģeometriju*, fizikāliem lielumiem, programmas vadības rīkiem utt.

Ģeometrija - faili, kas satur informāciju par modelējamo *apgabalu* virsmu, kā arī to savstarpējo izvietojumu aprēķinu telpā.

Reģioni - *apgabalu* grupēšana pēc to fizikālajām īpašībām. Piemēram, tukšajā jeb “empty” reģionā aprēķini netiek veikti, feromagnētiskie *apgabali* pieder pie nelineārā jeb “nonlinear” reģiona, jo tiem magnētiskā caurlaidība atkarīga no magnētiskā lauka. Dalīnu kustības aprēķins tiek veikts tikai “air” reģionā utt.

Superdalīņa - apraksta lielu kopu ar vienādām daļiņām (atomiem, elektroniem, molekulām). Rēķinot superdalīņas dinamiku, par masu un lādiņu tiek pieņemta vienas daļiņas masa un lādiņš, turpretim efektīvais sadursmes šķērsgriezums ir visu daļiņu efektīvo šķērsgriezumu summa. Rēķinot kvantitatīvus lielumus, superdalīņas masa un lādiņš tiek ņemti kā visu daļiņu masas un daļiņu summas.

- **Globālā iterācija** - aprēķinu solis, kad tiek veikti skaitliskie aprēķini visos apgabaloš. Multi-reģionu programmai aprēķini notiek katram *apgabalam* atsevišķi. Vispirms tiek rēķināti apgabali “*linear*”, tad “*nonLinear*” un visbeidzot “*air*” (skat. 3.1. att.), vērtības uz robežvirsmām starp dažādiem apgabaliem tiek “sašūtas” ar atbilstošiem robežnosacījumiem.

Abreviatūras

CFD - Computational Fluid Dynamics - skaitliskā hidrodinamika;

FEM - Finite Element Method - galīgo elementu metode.

FVM - Finite Volume Method - galīgo tilpumu metode.

FDM - Finite Difference Method - galīgo diferenču metode.

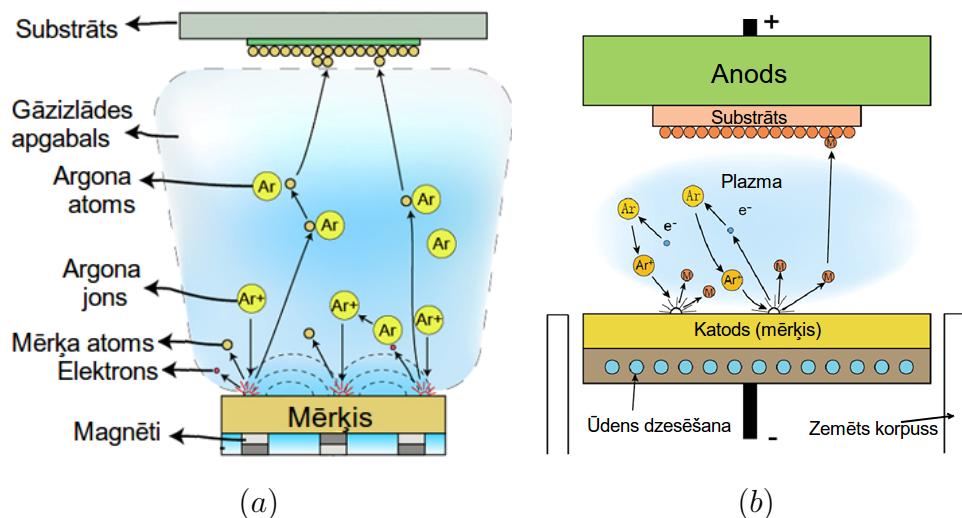
MHD - Magnetohydrodynamics - magnetohidrodinamika.

DSMC - Direct Simulation Monte Carlo - tiesās simulācijas Montekarlo (metode).

PIC - Particle In Cell - “dalīņas šūnā” (metode)

1. Ievads

Ikdienā cilvēki sastopas ar dažādiem tehnoloģiskiem izstrādājumiem, kuru funkcionalitātē būtiska nozīme ir plāno kārtīju pārklājumiem (biezums līdz dažiem mikroniem). Praksē tiek izmantotas gan šādu kārtīju optiskās (celtniecībā izmantotie stikli, optiskie instrumenti, dažādi ekrāni), gan elektriskās (Li jonu akumulatori, solārie paneli) u.c. īpašības. Metodes, ar kādām vēlamais materiāls tiek uzklāts uz izvēlētās pamatnes, ir dažādas. Viena no visplašāk izmantotajām ir materiāla (mērķa) uzklāšana uz pamatnes (substrāta), izmantojot magnetronu uzputināšanu (skat. 1.1. att.).



Att. 1.1: (a) Shematsika magnetronu sistēma, (b) uzputināšanas procesa fizika.
Attēli aizgūti no <http://www.directvacuum.com/sputter.asp>

Magnētu struktūra zem mērķa tiek veidota tā, lai magnētiskais lauks virs tā spētu satver elektronus, kas ar elektrisko lauku tiek virzienā no mērķa (katoda). Papildus šiem elektroniem tiek nodrošināta argona klātbūtne. Argona joni un elektroni veido plazmu mērķa tuvumā (1.1. (a) att.). Pozitīvie joni elektriskā lauka ietekmē triecas mērķī un izsits no tā režģa atomus, kas, saņēmuši šo kinētisko enerģiju, turpina ceļu līdz substrātam (1.1. (b) att.) ([3], [4]).

Magnetronu uzputināšanas metode ir piemērota lielu virsmu pārklāšanai ar viena vai vairāku materiālu plānām kārtījām. Magnetronu sistēmu efektivitāti nosaka uzputināmā materiāla izlietojuma vienmērīgums (erozija). Šo sistēmu tehniskā apkope ir biežākais iemesls ražošanas līniju apturešanai, no kā izriet, ka šīs sistēmas optimizēšana iekārtas lietotājam ļaus paīdzināt sistēmas ekspluatācijas laiku, samazināt ekspluatācijas izdevumus, palielināt ražošanas apjomus.

Papildus laboratorijas testiem, kuros praksē tiek mainīta magnētu konfigurācija un citi sistēmas parametri, bieži tiek izmantota skaitliskā modelēšana. Vispusīgi izstrādāts matemātiskais modelis, kas korekti realizēts programmā, lauj veikt dažādas parametru

studijas, tādējādi ietaupot laika un finanšu līdzekļus.

Atbilstošu programmu kompleksu ir iespējams iegūt divos veidos: iegādāties kādu no pasaule pieejamajām skaitliskās modelēšanas paketēm, piemēram, ANSYS, COMSOL u.c., vai piemērot risināmajam uzdevumam kādu atvērtā koda programmu, kādas bez maksas ir pieejamas interneta. Literatūrā ir aprakstīti rezultāti līdzīgu problēmu risināšanai, kas iegūti ar pašrakstītām programmām, bet šis programmas ir attiecīgo institūciju īpašums. Tādējādi finansiāli izdevīgākais risinājums ir veidot savu programmu kompleksu, kas būtu balstīts uz jau gataviem un notestētiem tehniskiem risinājumiem.

1.1. Darba mērķis un uzdevumi

Šī darba **mērķis** ir izstrādāt programmu kompleksu, kas risinātu sekojošas fizikālas problēmas:

- magnētiskā lauka konfigurācija, ko rada pastāvīgie magnēti;
- elektriskā lauka konfigurācija, kas veidojas magnetronu sistēmā starp katodu un anodu;
- lādētu daļiņu kustība šo lauku ietekmē; lādētu daļiņu savstarpējā mijiedarbība.

Lai sasniegtu mērķi, par aprēķinu programmas pamatu izvēlēta programmēšanas valodā C++ rakstītā atvērtā koda bibliotēka OpenFOAM, jo tā atbilst šādiem kritērijiem:

- bezmaksas, atvērta koda programma ar neierobežotu paralēlo aprēķinu atbalstu;
- tā ietver paketes ātrai režģa ģenerācijai un apstrādei;
- programma spēj vienlaikus modelēt vairākus procesus, nodrošināt dažādu procesu modelēšanu dažādos reģionos un to mijiedarbību;
- brīvi pieejami programmu piemēri, plaš lietotāju loks un atbalsts.

Darba gaitā nepieciešams atrisināt sekojošus **uzdevumus**:

- atrisināt problēmas, kas saistītas ar magnetronu sistēmu ģeometrijas izveidi un tās daļu atšķirīgajām fizikālajām īpašībām;
- izstrādāt un pārbaudīt modeli, kas ļauj skaitliski reķināt kvazistacionāru elektrisko un magnētisko lauku;
- piemērot moduli lādētu daļiņu kustības modelēšanai un notestēt tā darbību ārējo lauku klātbūtnē;
- pārbaudīt daļiņu savstarpējās mijiedarbības pamatprincipus;
- veidot programmu kompleksu tā, lai tas būtu ātri pielāgojams dažādu sistēmu modelēšanai un kvantitatīvai un kvalitatīvai to efektivitātes salīdzināšanai.

1.2. Iestrādes, līdzīgu pētījumu un literatūras apskats

Par mana darba atskaites punktu magnētiskā lauka modelēšanā kalpo *Mārtiņa Sarmas* (*Latvijas Universitāte*) bakalaura darbs “*Magnētiskā lauka 3D modelēšana magnetronu izputināšanas sistēmās*” un programma `potencFoam` [10]. *Sarma* atsaucas uz citu autoru darbiem magnetronu uzputināšanas procesa modelēšanā ([1], [7], [8]), pamatojot modelēšanas formulējumu un paņēmienu izvēli.

1.2.1. Programmas pamats, ģeometrijas un režģa ģenerācija

Uzsākot bakalaura darbu vispirms tika meklēti paņēmieni `Gmsh` režģa kvalitātes uzlabošanai (ģeometrijas un režģa ģenerācija ar `Gmsh` un tā importēšana `OpenFOAM` aplūkota *Mārtiņa Sarmas* darbā [10]). Šim režģa ģenerēšanas paņēmienam tika konstatēti daži trūkumi, piemēram, komplikētu ģeometriju ģenerēšana ir laikietilpīgs un pacietīgs process. `Gmsh` programma vairāku apgabalu ģeometrijām nespēj ġenerēt strukturētu režgi, tai pat laikā tetraedru režģim ir sarežģīti kontrolēt smalcinājumu. Tobrīd tika atrasts alternatīvs režģa ģenerēšanas veids, kurš realizēts `OpenFOAM` multi-reģionu aprēķinu programmā `chtMultiRegionFoam` [28]. Šī programma modelē siltuma plūsmu un citus termodinamiskos procesus, labs piemērs un pamats modificētu multi-reģionu programmu izveidei un robežnosacījumu uzdošanai, tajā ir realizēta automātiska režģa ģenerācija ar `OpenFOAM` programmu `snappyHexMesh` [19]. Programmai `chtMultiRegionFoam` ir atbilstošs aprēķinu piemērs `snappyMultiRegionHeater` [27]. Līdz ar to turpmākā programmas izstrāde notika uz programmas `chtMultiRegionFoam` un aprēķinu piemēra `snappyMultiRegionHeater` pamata (programmas uzbūve aplūkota 3. nodaļā).

Tā kā `snappyHexMesh` ġenerē un smalcina režgi, balstoties uz apgabalu ārējām virsmām, kuras tiek importētas `.stl` failu formātā, tālāk sekoja programmas `Salome` apguve, kura ir lietotājam draudzīga programma ar plašām ģeometrijas manipulācijas iespējām - ģeometriskie pamatelementi, rotācijas, loģiskās operācijas [20]. Šī pamācība ģeometrijas izveidē ar `Salome` un režģa ģenerācija ar `snappyHexMesh` aplūkota 2. nodaļā.

1.2.2. Magnētiskā lauka modelēšana

`OpenFOAM` ietver vienkāršu programmu magnētiskā lauka modelēšanai `magneticFoam` [29], tā rēķina magnētisko lauku caur skalāro potenciālu Ψ . Kā parādīja *Sarma* [10], šī programma neder gadījumiem, kad magnētiskā lauka avoti ir strāvas un virpulšstrāvas, proti, tai ir iespējams tikai uzstādīt konstantu magnētiskā potenciāla vērtību uz kādas no robežvirsmām.

Magnētiskā lauka modelēšanu `OpenFOAM`, izmantojot magnētiskā vektorpotenciāla formulējumu, pētījusi arī *Zhe Huang* maģistra darbā [11]. Darbā aplūkoti aprēķinu pie-

mēri ar taisnstūrveida vados plūstošām strāvām, salīdzināti rezultāti ar COMSOL, dots programmas kods. Mana darba rezultāti 4. nodalā parādīs, ka magnētiskā lauka modelēšana OpenFOAM ir tikai sākuma attīstības stadijā, un vēl ir virkne neatrisinātu problēmu, kas saistītas ar matemātiski korektu robežnosacījumu uzdošanu.

Tika apsvērta iespēja izmantot kādu citu formulējumu, tāpēc ļoti noderīgs un uzskatāms bija *Pietro Testoni* darbs [13], kurā aplūkoti vairāki kvazi-stacionāra magnētiskā lauka aprēķināšanas formulējumi, kā arī veikts to ātrdarbības salīdzinājumu ANSYS programmā. Kaut gan tika mēģināts ieviest citu formulējumu OpenFOAM, tas nevainagojās ar panākumiem. Turpretim izdevās uzlabot magnētiskā vektorpotenciāla aprēķinu apgabala ārējos robežnosacījumus. Atvērto robežnosacījumu formulējums magnētiskajam vektorpotenciālam dots *David Meeker* FEMM pamācībā [14], kur atrodama formula magnētiskā vektorpotenciāla ārējo robežnosacījumu aproksimācijai sfēriskam apgabalam. Arī feromagnētiķu magnētiskās caurlaidības atkarību no magnētiskā lauka izdevās realizēt. Tehniskā realizācija aizgūta no *Anja Miehe* foruma ieraksta [22]. Ierakstā publicēts kods parametru atkarības uzdošanai tabulas veidā, interpolācijai un atbilstošas vērtības noteikšanai.

1.2.3. Retinātas plazmas modelēšana

Plazmas modelim jāmodelē daļiņu mijiedarbība ar ārējiem laukiem, savstarpēja elektromagnētiska mijiedarbība. Šo funkciju veic *Particle-In-Cell* (PIC) metode. Rakstus par plazmas modelēšanu ar PIC metodi un citiem risinājumiem, piemēram, daļiņu lādiņa sadalīšana pa režīga šūnas virsotnēm, kā arī virkni MATLAB kodus publicē “*Particle In Cell Consulting*” blogā [25].

Plazmas modelim arī jāmodelē daļiņu mijiedarbība ar apgabalu virsmām, daļiņu sadursmes un ķīmiskās reakcijas. Šādām funkcijām atbilst “*Direct Simulation Monte Carlo*” (DSMC) metode, kura paredzēta retinātas, neutrālas gāzes modelēšanai. Standa rīta OpenFOAM programmā `dsmcFoam` ir realizēta šī metode [30]. Programma rēķina daļiņu trajektorijas, mijiedarbību ar citām daļiņām un robežvirsmām. Daļiņu haotiskā siltumkustība pakļaujas Boltmaņa sadalījumam. OpenFOAM realizētās programmas pārbaude un rezultāti atrodami darbā [9]. Atbilstošs `dsmcFoam` programmas inicializācijas jeb aprēķinu piemērs ir `freeSpaceStream` [32], kurā tiek modelēta cauri aprēķinu apgabalam lidojošu daļiņu plūsma.

DSMC metodes teorēriskie aspekti, izvedumi atrodami metodes autora *G. A. Bird* grāmatā [5]. OpenFOAM standarta programma `dsmcFoam` balstās uz modeļiem un formulām no šīs grāmatas. *G. A. Bird* mājas lapā [www.gab.com.au/] brīvi lejuplādējamas vairākas DSMC programmas. Otra, cita autora rakstīta, jaunāka grāmata [6] papildina *Bird* grāmatu ar papildus modeļiem daļiņu sadursmju modelēšanai, taču OpenFOAM tie nav realizēti.

2. Ģeometrijas un režģa ģenerācija

Nākamajās nodaļās tiks aplūkots viens multi-reģionu programmai paredzētu režģu ģenerācijas paņēmiens. Galvenie ģeometrijas un režģa ģenerācijas soli:

1. Ģeometrijas izveide ar programmu **Salome**.
2. Ģeometrijas apstrāde, virsmas režģa ģenerācija un eksportēšana *.stl* formātā ar **Python** skriptu, kuru palaiž no **Salome** programmas.
3. *.stl* failu apvienošana ar **Bash** skriptā rakstītu programmu.
4. **snappyHexMesh** parametru uzstādīšana.
5. Režģa ģenerācija ar **snappyHexMesh**.

2.1. Pirms un pēc-apstrādes programma **Salome**

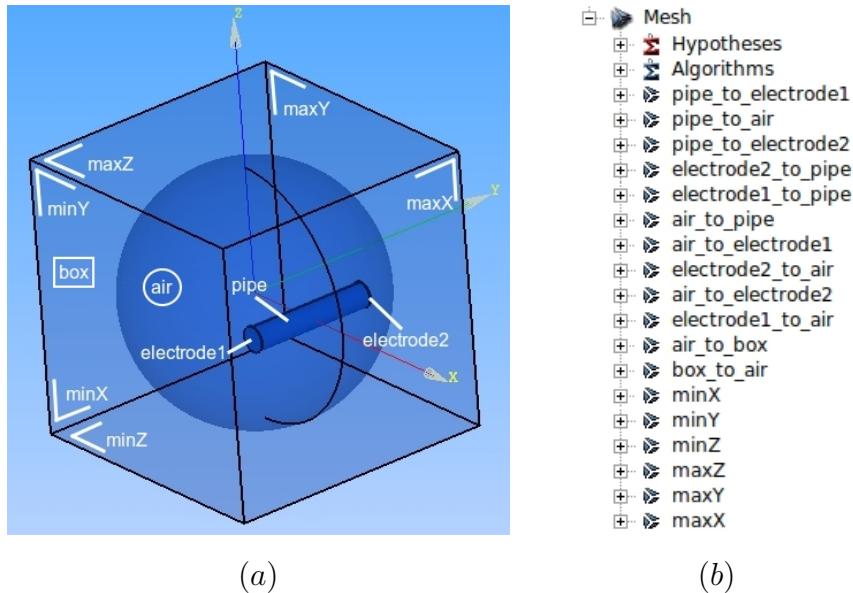
2.1.1. Ģeometrijas veidošana

Pamācības, kā veidot ģeometrijas ar **Salome** atrodamas oficiālajā mājas lapā [20]. Būtiski ir pareizi ģenerēt ģeometrijas *.stl* failus, lai tos varētu importēt **snappyHexMesh** un **blockMesh** programmās.

Aplūkosim vienkāršu ģeometriju - ārējā režģa virsma (*minX*, *minY*, *minZ*, *maxX*, *maxY* un *maxZ*), aprēķinu apgabals *air*, elektrodi (*electrode1*, *electrode2*) un vads *pipe*, skat. 2.1. (a) attēlu. Elektrodi ir uzdoti ļoti plāni, tikai dažu režģa šūnu biezumā, tomēr tie ir telpiski objekti. Lai varētu veiksmīgi importēt ģeometriju, ir jāizpildās šādiem nosacījumiem:

- Visi apgabali ir pilnīgi noslēgti. Nedrīkst būt pat nelielas šķirbas uz šķautnēm, kur savienotas divas virsmas. Šīm virsmām ir jābūt kopīgiem punktiem uz šķautnes.
- Apgabalu virsmām jāievēro pareizs normāles virziens, tam jābūt vērstam uz ārpusi no apgabala.
- Ārējā režģa virsma (*minX*, *minY*, *minZ*, *maxX*, *maxY* un *maxZ*) atbilst **blockMesh** režģa ģenerācijas apgabalam.
- Apgabalu robežvirsmām jābūt nosauktām tā, kā parādīts 2.1. (b) attēlā. Piemēram, riņķis, kas savieno *electrode1* ar *pipe* un pieder *electrode1*, saucas *electrode1_to_pipe*. *air* apgabala čaula starp *air* un *electrode1* (riņķis, apvienots ar cilindrisko virsmu), saucas *air_to_electrode1*.
- Apgabalu ģeometriju jeb iegūtos 2D virsmas režģus jāsaglabā kā *.stl* failus turpmākai to apvienošanai un importēšanai **OpenFOAM**.

Lai vienkāršotu ģeometrijas apstrādi, režģa ģenerāciju un eksportēšanu, tika uzrakstīts skripts (8.1. pielikums), ko palaiž iekš **Salome** ar rīku “Load Script”. Skripts izgriež no *box* apgabala *air* apgabalu. Izgriež no *air* apgabalus *electrode1*, *electrode2* un *pipe*. Atrod kopīgas virsmas starp jebkuriem diviem apgabaliem, grupē tās un pārsauc. Pēc tam veic virsmu režģu ģenerāciju un eksportēšanu.



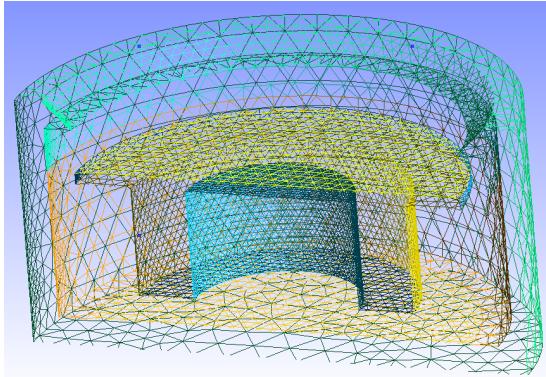
Att. 2.1: (a) Vienkāršs ģeometrijas piemērs, (b) **Salome** robežvirsmu struktūra (2D režģi).

2.1.2. Ģeometrijas eksportēšana uz OpenFOAM

Iepriekšējā apakšnodaļā tika iegūti *.stl* faili atsevišķām robežvirsmām, taču **snappyHexMesh** darbojas ar *.stl* failiem, kas satur informāciju par apgabala pilno ārējo virsmu. Robežvirsmu apvienošanai var izmantot **Bash** programmu (skat. pielikumu 8.2.), kuru palaiž tajā datnē, kurā atrodas nepieciešamie *.stl* faili. Programma paredz, ka ir 5 apgabali *air*, *pipe*, *electrode1*, *electrode2* un *box*. Piemēram, apgabalam *pipe* tā izveidos atsevišķu failu ar nosaukumu *pipe.stl*, kurā iekopēs failu *pipe_to_air.stl*, *pipe_to_electrode1.stl* un *pipe_to_electrode2.stl* saturu. Apgabalam *box* papildus tiks saskopēts visu *max** un *min** failu saturs, jo tā ir apgabala ārejā virsma.

Iegūtos 5 apgabalu *.stl* failus iekopē *constant/triSurface*, kur tos **snappyHexMesh** spēs izmantot režģa ģenerācijai.

.stl režģa fragments attēlots 2.2. attēlā. Tā ir ģeometrija no piemēra, kas tiks aplūkots 4.2.4. nodaļā.



Att. 2.2: 4.2.4. nodaļā aplūkotā piemēra *.stl* virsmas režģa fragments.

2.2. Režģa ģenerācijas programmas **blockMesh** un **snappyHexMesh**

snappyHexMesh un **blockMesh** ir OpenFOAM integrēti režģa ģeneratori. **blockMesh** ģenerē strukturētus režģus, taču aprēķinu ģeometrijas uzdošana šajā programmā ir sarežģīta un laikietilpīga. Teksta veidā tiek ievadīti visi ģeometrijas virsotņu punkti, līdzīgi tiek definētas šķautnes un skaldnes.

Kā papildinājumu **blockMesh** var izmantot automātisko režģa ģeneratoru **snappyHexMesh**, kurš izmanto **blockMesh** ģenerētu taisnstūra paralēlskaldņa formas aprēķinu apgabala strukturēto režģi, tad importē ģeometrijas failus, un veic režģa smalcināšanu (heksaedru dalīšanu 8 daļās) ģeometrijas virsmu tuvumā un citos norādītajos apgabalos, visbeidzot, pārbīda un deformē režģa šūnas līdz tās tuvināti atbilst ģeometrijas virsmai.

Šīs programmas priekšrocība ir kompleksu ģeometriju importēšana *.stl* formātā, kas ir ļoti izplatīts virsmas ģeometrijas formāts CAD programmās. Ģeometrija *.stl* failos tiek uzdota kā mazu trijstūru virsma, katram trijstūrim ir trīs virsotņu punkti un normāles vektors.

2.2.1. Režģa ģenerācija

Vispirms nepieciešams ģenerēt strukturētu režģi, kas aptver visu aprēķinu apgabalu, režģa šūnām jābūt heksaedriem. Režģa parametru uzstādīšanu veic bibliotēka **blockMeshDict**, kura atrodas *constant/polyMesh* mapē.

snappyHexMesh vada **snappyHexMeshDict** bibliotēka, kurai jāatrodas *system* mapē. Ģeometrijas importēšana **snappyHexMesh** notiek slēgtiem apgabaliem. Ģeometrijas failus pievieno bibliotēkas *geometry* sadaļā. Atsevišķu virsmu smalcināšanu var veikt sadaļā *refinementSurfaces*, atsevišķu reģionu režģu smalcināšanu nosaka sadaļa *refinementRegions*.

2.3. Norādījumi

Režģa ģenerācijas problēmas rodas, ja:

- `snappyHexMesh` smalcināšanas inicializācijas punkts “*locationInMesh*” atrodas uz ģeometrijas virsmas (`snappyHexMesh` kļūda). Risinājums - mainīt šī punkta koordinātas.
- Kāda no `blockMesh` režģa šūnas virsmām pilnīgi sakrīt ar ģeometrijas virsmu. Risinājums - mainīt `blockMesh` režģa smalcinājumu.
- Reģiona lineārie izmēri ir mazāki par `blockMesh` šūnas izmēru (`snappyHexMesh` brīdinājums). Risinājums - palielināt `blockMesh` režģa smalcinājumu.
- `.stl` trijsstūri ir pārāk lieli (`snappyHexMesh` kļūda). Risinājums - importēt smalkāku `.stl` virsmas režģi.
- Režģa šūnas izliektas virsmas tuvumā ir pārāk lielas, un tās nevar pārbīdīt un deformēt līdz tuvinātai virsmas atbilstībai (`snappyHexMesh` kļūda vai brīdinājums). Risinājums - palielināt režģa smalcināšanas pakāpi virsmu tuvumā.
- Apgabala virsma nav pilnīgi slēgta (`snappyHexMesh` brīdinājums). Risinājums - ar `Gmsh` iespējams vizuāli pārbaudīt, vai apgabalam ir visas skaldnes un skaldņu savienojumos nav caurumu. Ja tiek konstatēta kāda no nepilnībām, veic ģeometrijas labošanu.
- Ja palaižot programmu `splitMeshRegions` (sīkāk skat. [27] skripta failu *Allrun*) tiek ģenerēti apgabali *domain0*, *domain1* utt., tad jāpalielina `.stl` virsmas režģa smalcinājums.

3. Programmas uzbūve

3.1. Aprēķinu programma `multiRegionPlasmaFoam`

Šajā darbā gan magnētiskā lauka, gan plazmas modelēšanai tiek izmantota pašrakstīta aprēķinu programma `multiRegionPlasmaFoam`, kas veidota uz programmu `chtMultiRegionFoam` [28] un `dsmcFoam` [30] pamata.

`chtMultiRegionFoam` ir viena no atvērtā koda bibliotēkas `OpenFOAM` standartprogrammām, kas ļauj modelēt siltuma plūsmu cietos apgabaloš un gāzēs, kā arī gāzes turbulentu plūsmu. *Mārtiņa Sarmas* bakalaura darbā [10] magnētu magnetizāciju un feromagnētiķu magnētiskā caurlaidība tika uzdota ar `OpenFOAM` papildinājumā `swak4Foam` iekļautās programmas `funkySetFields` palīdzību, kura apgabalus ar citādām īpašībām (nekā pārējai apgabala daļai) uzdot ar analītisku izteiksmju palīdzību, taču to iespējams izdarīt tikai vienkāršām ģeometrijām. Multi-reģionu programmas gadījumā (tehnisko realizāciju skat. 3.1.2. nodaļā) materiālu īpašības var uzdot jebkādas formas apgabalam. Cits iemesls, kāpēc tika izvēlēta multi-reģionu pieeja, ir fundamentāli atšķirīgu procesu uzdošana dažādiem reģioniem, piemēram, programma rēķina plazmas dinamiku tikai *air* reģionā (skat. 3.1.1. nodaļu). Visbeidzot, `chtMultiRegionFoam` tika izvēlēts par programmas pamatu, jo tajā ir integrēts automātiskais jauktā režīga (strukturēts heksaedru režīgs ar nestukturētiem tetraedriem robežvirsmu tuvumā) ģenerators `snappyHexMesh` (skat. 2.2. nodaļu).

Programma `dsmcFoam` ir `OpenFOAM` standartprogramma, tā ļauj modelēt retinātu gāzu dinamiku, aplūkojot gāzi kā diskrētu daļiņu kustību, to sadursmes ar citām daļiņām un sienām. Šī programma tika pielāgota multi-reģionu atbalstam, kā arī tika ieviests daļiņu lādiņa jēdziens, lādētu daļiņu un elektromagnētiskā lauka mijiedarbība.

3.1.1. Failu un mapju struktūra

Programmas failu un mapju struktūru skat. 3.1. attēlā. Svarīgākās komponentes:

- *multiRegionPlasmaFoam.C* ir galvenais programmas fails. Faila saturu skat. pielikumā 8.3. nodaļā.
- *regionProperties.C* un *regionProperties.H* tiek inicializēti šādi reģioni:
 - # *air* - reģions ar vienādojumiem, kas optimizēti, ievērojot, ka vakuumu un gaisa magnētiskā caurlaidība ir 1. Tikai šajā reģionā iespējams rēķināt lādētu daļiņu kustību.
 - # *empty* - tukšs reģions, kurā netiek ielasīti lauki, nenotiek aprēķini. To var lietot gadījumos, ja ģeometrija ir caurumi, kur aprēķinus veikt nav nepieciešams.

- # *linear* - reģions materiāliem ar konstantu magnētisko caurlaidību.
- # *nonlinear* - reģions materiāliem ar mainīgu magnētisko caurlaidību.
- *create *Meshes.H* izveido sarakstu ar reģiona apgabalu režīiem. Turpmāk failu nosaukumos ar “*” aizstāsim reģionu nosaukumus *air*, *empty*, *linear* un *nonlinear*.
- *create *Fields.H* izveido sarakstu ar reģiona apgabalu aprēķinu laukiem.
- *setRegion *Fields.H* tiek izsaukts katrā iterācijas ciklā, programmas lokālajiem mainīgajiem tiek iedotas aprēķinu lauka adreses.
- *solve *.H* tiek izsaukts katrā iterācijas ciklā, nodrošina aprēķinus.
- *interpolateProperties.H* nodrošina nelineāru lielumu tabulu ielasīšanu un interpolāciju. Programmas papildinājums aizgūts no Anja Miehe foruma ieraksta [22].
- *AMixedFvPatchVectorField* un *AOpenFvPatchVectorField* ir magnētiskā vektorpotenciāla **A** robežnosacījumi, attiecīgi, robežnosacījumi starp diviem apgabaliem un atvērtie robežnosacījumi.
- *files* un *options* pievieno ārējās bibliotēkas, piemēram, daļiņu atbalsts tiek nodrošināts, failā *open* norādot adresi līdz *dsmc* bibliotēkai (skat 3.1.2. nodaļu).
- *Allwmake* un *Allwclean* ir *Bash* skripta faili programmas kompilešanai un dzēšanai.

3.1.2. Veiktie programmas pārveidojumi

No programmas **chtMultiRegionFoam** fizikāliem lielumiem temperatūra T ir vienīgais lielums, kas visiem reģioniem ir kopīgs. Šajā programmā tiek realizēta temperatūras “sašūšana” uz apgabalu robežvirsmām, tāpēc šis parametrs kalpoja kā magnētiskā vektorpotenciāla **A** ekvivalenti (par vektorpotenciāla **A** izvēli skat. 4 nodaļā).

Tika veikta pāreja no skalārā lieluma T uz vektoriālo **A**, izveidoti jauni robežnosacījumi (sīkāk skat. 4.2.1., 4.2.2. un 4.2.4. nodaļu) uz iekšējām un ārējām robežvirsmām. No programmas **chtMultiRegionFoam** tika aizgūti koda fragmenti apgabalu un lauku ielasīšanai, aprēķinu cikli, failu struktūra, kura aplūkota iepriekšējā 3.1.1. nodaļā.

Nelineārās magnētiskās caurlaidības atbalstam tiek izmantotas **OpenFOAM** funkcijas *graph* un *interpolateXY* (sīkāk skat. 4.2.3. nodaļu). [22]

Programma **dsmcFoam** daļiņu dinamikas modelēšanai izmanto bibliotēku **dsmc** [31]. **OpenFOAM** bibliotēkas ir apakšprogrammas, kuras kompile atsevišķi, un tad bibliotēkas adresi pievieno aprēķinu programmas failā *open*. Bibliotēka **dsmc** tika papildināta ar elektriskā lādiņa īpašību, kā arī tika realizēta elektriskā un magnētiskā lauka ielasīšana, spēka, paātrinājuma, ātruma un trajektorijas aprēķins.

3.2. Aprēķinu piemērs `multiRegionPlasmaTest`

Aprēķinu piemērs `multiRegionPlasmaTest` veidots uz `snappyMultiRegionHeater` [27] pamata. Aprēķinu piemēra failu un mapju struktūru skat. 3.1. attēlā. Apostrofos liktie nosaukumi var tikt patvalīgi izvēleši.

- Mapē *0* atrodas tukši aprēķinu lauki. Programma `changeDictionary` šajā mapē katram apgabalam izveido mapi ar tiem atbilstošajiem aprēķinu laukiem, kas tiek uzdoti bibliotēkā `changeDictionaryDict`.
- Mapē *triSurface* tiek kopēti ģeometrijas *.stl* faili.
- *muProperties* satur tabulu ar magnētiskās caurlaidības atkarību no magnētiskā lauka intensitātes. Tā kā katram apgabalam atbilst sava mape, tad katram apgabalam ar nelineāru magnētisko caurlaidību, var būt savs fails *muProperties*, līdz ar to dažādas magnētiskās īpašības.
- *blockMeshDict* nodrošina strukturēta režģa ģenerēšanu ar programmu `blockMesh`.
- *regionProperties* atbilstošajiem reģioniem tiek piekārtoti aprēķinu apgabali.
- *changeDictionaryDict* ir programmas `changeDictionary` bibliotēka, kura katram apgabalam uzstāda aprēķinu lauku sākumnosacījumus.
- *decomposeParDict* nodrošina paralelos apgabalus gadījumā, ja aprēķins tiek sadalīts uz vairākiem procesoriem. Bibliotēkā tiek norādīts, kā dalīt apgabalus.
- Mapē *system* faili *fvSchemes* un *fvSolution* ir tukši, jo nepieciešami tikai programmas inicializācijai. Bibliotēkās *fvSchemes* un *fvSolution*, kuras atrodas apgabalu mapēs, attiecīgi tiek norādītas diferenciālvienādojumu diskretizācijas shēmas un risinājuma konvergences kritēriji.
- *snappyHexMeshDict* bibliotēka kalpo režģa programmas `snappyHexMesh` inicializācijai.
- *Allrun* un *Allclean* ir Bash skripta faili aprēķinu programmas palaišanai un aprēķina piemēra rezultātu dzēšanai.



Att. 3.1: Programmas `multiRegionPlasmaFoam` (pa kreisi) un aprēķinu piemēra `multiRegionPlasmaTest` (pa labi) failu struktūra.

4. Magnētiskā lauka modelēšana

Šajā nodalā tiek aplūkoti trīs formulējumi magnētiskā lauka modelēšanai. Vispārīgās elektromagnētisma sakarības un formulējumu izvedumi 4.1., 4.1. un 4.1. nodalās aizgūti no *P. Testoni* darba [13].

4.1. Vispārīgas elektromagnētisma sakarības

Nepārtrauktas vides vienādojumi

Magnētisko lauku un elektrisko lauku saista Maksvela vienādojumi

$$\nabla \times \mathbf{H} = \mathbf{J}, \quad (4.1)$$

$$\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t}, \quad (4.2)$$

$$\nabla \cdot \mathbf{B} = 0, \quad (4.3)$$

kur \mathbf{H} - magnētiskā lauka intensitātes vektors, \mathbf{J} - strāvas blīvums, \mathbf{E} - elektriskā lauka intensitātes vektors, \mathbf{B} - magnētiskā lauka indukcijas vektors, t - laiks. Pirmais vienādojums (4.1) ir Ampēra likums, otrs (4.2) - Faradeja likums, trešais (4.3) - Gausa likums.

Vienādojumā (4.1) netiek ņemta vērā nobīdes strāva $\partial \mathbf{D} / \partial t$. Ja apgabali, kuros plūst virpulšstrāvas, ir mazi, salīdzinot ar elektromagnētisko svārstību viļņa garumu šajos apgabalos, tad lielums $\partial D / \partial t$ var tikt neņemts vērā, jo ir mazs. \mathbf{B} un \mathbf{E} laukus ar materiāla īpašībām saista sakarības

$$\mathbf{B} = \mu \mu_0 \cdot \mathbf{H}, \quad (4.4)$$

$$\mathbf{J} = \sigma \cdot \mathbf{E}, \quad (4.5)$$

kur μ_0 - magnētiskā konstante ($1.257 \cdot 10^{-6} \frac{H}{m}$), μ - relatīvā magnētiskā caurlaidība, σ - elektriskā caurlaidība.

Materiālus pēc μ klasificē diamagnētiķos ($\mu < 1$), paramagnētiķos ($\mu \geq 1$) un feromagnētiķos ($\mu \gg 1$). Feromagnētiķiem μ nav konstants, feromagnētiķu magnētiskās īpašības raksturo \mathbf{H} atkarība no \mathbf{B} , kuru sauc par magnētisko histerēzi, no kurās, savukārt, ar vienādojuma 4.4 palīdzību iespējams aprēķināt μ . Maziem \mathbf{B} histerēzes līkne ir lineāra ($\mathbf{H} \sim \mathbf{B}$) un μ ir konstants, taču, turpinot palielināt \mathbf{B} , feromagnētiķis “piesātinās” un μ vērtība samazinās. μ atkarības no \mathbf{B} ieviešana `multiRegionPlasmaFoam` tiek aplūkota 4.2.3. nodalā.

Materiālus pēc σ iedala dielektriķos, pusvadītājos un vadītājos. σ ir atkarīgs arī no temperatūras, piemēram, pusvadītājiem, pieaugot temperatūrai, σ palielinās, turpretim,

vadītājiem (parasti metāliem) samazinās. Izstrādātā aprēķinu programma nerēķināta siltuma plūsmu un temperatūru, tāpēc σ atkarība netiek nēmta vērā.

E aprēķinus veic caur skalāro elektrisko potenciālu ϕ , kur

$$\Delta\phi = \frac{\rho}{\epsilon_0}, \quad (4.6)$$

kur ρ ir elektriskā lādiņa blīvums, ϵ_0 - elektriskā konstante. **E** rēķina kā

$$\mathbf{E} = \nabla\phi. \quad (4.7)$$

Lādiņa nezūdamību apraksta strāvas nepārtrauktības vienādojums

$$\nabla \cdot \mathbf{J} = 0. \quad (4.8)$$

Izdalīsim trīs strāvu rašanās iemeslus. Pirmkārt, elektriskās strāvas rodas vadītājos ārēja **E** rezultātā, šīs strāvas apzīmēsim ar **J_S**. Otrkārt, mainīgā magnētiskajā laukā (arī kvazistacionārā) vadītājos inducējas strāvas, ko sauc par virpuļstrāvām **j_S**. Treškārt, magnētu magnetizācijas rezultātā rodas virsmas magnetizācijas strāvas **J_M**. **J_S** rēķina kā

$$\mathbf{J} = -\sigma\mathbf{E}. \quad (4.9)$$

Virpuļstrāvas apzīmējums **j_S** turpmāk nekur tieši neparādīsies, jo, piemēram, vienādojumā (4.25) tas ir izsakāms kā

$$\mathbf{j}_S = \sigma \frac{\partial \mathbf{A}}{\partial t}, \quad (4.10)$$

kur **A** ir magnētiskais vektorpotenciāls, t - laiks. **J_M** ar magnēta magnetizāciju **M** saista vienādojums

$$\mathbf{J}_M = \nabla \times \mathbf{M}. \quad (4.11)$$

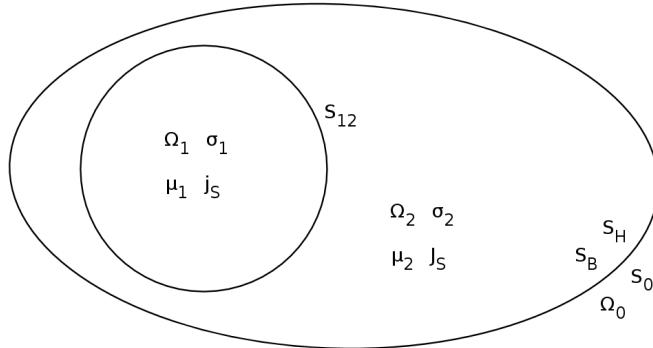
Tā kā **J_M** ir laikā nemainīgs, tad turpmāk pieņemsim, ka **J_M** ir jau ietverts **J_S**.

Problēmas formulējums un robežnosacījumi

Tiešā veidā Maksvela vienādojumus skaitliskos aprēķinos parasti neizmanto (tiešā veidā tos izmanto analītiskajos aprēķinos). Pirmkārt, tas ir saistīts ar šādu aprēķinu sarežģītību (vienādojumu diskretizāciju), otrkārt, ar zemo aprēķinu efektivitāti, respektīvi, lielu aprēķinu laiku. Lai vienkāršotu un paātrinātu magnētisku problēmu skaitliskos aprēķinus, izmanto dažādus formulējumus, kur **B** vai **H** izsaka ar potenciālu palīdzību.

Lai palielinātu aprēķinu ātrumu un mazinātu atmiņas patēriņu, skaitliskajos aprēķinos parasti nošķir apgabalus ar un bez virpuļstrāvām. Šāda problēma attēloota 4.1. attēlā. Tā sastāv no virpuļstrāvas apgabala Ω_1 ar atbilstošu īpatnējo elektrovadītspēju σ_1 un relatīvo magnētisko caurlaidību μ_1 . Šo apgabalu norobežo robežvirsmas S_{12} , kura ir kopīga ar bez-virpuļstrāvu apgabalu Ω_2 , kurā var būt avota strāvas **J_S** (arī apgabalā Ω_1

tādas var būt). Visu aprēķinu apgabalu apzīmēsim ar $\Omega_0 = \Omega_1 + \Omega_2$. Aprēķinu apgabala ārējā robežvirsmu ir S_0 (šajā gadījumā tā ir Ω_2 ārējā robežvirsmu), robežnosacījumus uz šīs virsmas dalām divās komponentēs - normālā S_B (raksturo magnētiskā indukcijas lauku plūsmu caur ārējo robežvirsmu) un tangenciālā S_H (raksturo magnētiskā intensitātes lauka plūsmu caur ārējo robežvirsmu).



Att. 4.1: Aprēķinu apgabalu, lauku un robežvirsmu apzīmējumi.

Uz robežvirsmas S_{12} izpildās robežnosacījumi

$$\mathbf{B}_1 \cdot \mathbf{n}_1 + \mathbf{B}_2 \cdot \mathbf{n}_2 = 0, \quad (4.12)$$

$$\mathbf{H}_1 \times \mathbf{n}_1 + \mathbf{H}_2 \times \mathbf{n}_2 = 0, \quad (4.13)$$

kur \mathbf{n}_1 un \mathbf{n}_2 ir virsmas normāles vektori (turpmāk pieņemsim, ka apgabala virsmas normāles vektori ir vērsti uz apgabala iekšpusi).

Uz robežvirsmas, kur vismaz viens apgabals ir dielektrikis, izmantosim robežnosacījumu

$$\mathbf{J} \cdot \mathbf{n}_1 = 0. \quad (4.14)$$

Ja ārējā robežvirsmu S_0 ir loti tālu no magnētisko lauku avotiem, tad robežnosacījumi ir

$$\mathbf{B} \cdot \mathbf{n} = 0 \quad \text{uz } S_B, \quad (4.15)$$

$$\mathbf{H} \times \mathbf{n} = 0 \quad \text{uz } S_H. \quad (4.16)$$

4.2. \mathbf{A} , \mathbf{V} - \mathbf{A} (vektor-potenciāla) formulējums

Magnētiskā lauka indukciju \mathbf{B} varam izteikt ar patvalīga vektora \mathbf{A} rotoru

$$\mathbf{B} = \nabla \times \mathbf{A} \quad (4.17)$$

jo vienādojums (4.3) atbilst $\nabla \cdot \nabla \times \mathbf{A} = 0$. Sekojoši, vienādojumu (4.2) varam pārrakstīt kā

$$\nabla \times \left(\mathbf{E} + \frac{\partial \mathbf{A}}{\partial t} \right) = 0 \quad (4.18)$$

un, izmantojot sakarību $\nabla \times \nabla V = 0$, kur V ir pietiekami diferencējama skalāra funkcija, iegūstam

$$\mathbf{E} = -\frac{\partial \mathbf{A}}{\partial t} - \nabla V. \quad (4.19)$$

Lielumi \mathbf{A} un V tiek saukti par magnētisko vektorpotenciālu un elektrisko skalāro potenciālu. Ievietojot vienādojumus (4.17) un (4.19) vienādojumos (4.1), (4.4) un (4.5) iegūstam

$$\nabla \times \nu \nabla \times \mathbf{A} + \sigma \left\{ \frac{\partial \mathbf{A}}{\partial t} + \nabla V \right\} = 0 \quad \text{apgabalam } \Omega_1, \quad (4.20)$$

$$\nabla \times \nu \nabla \times \mathbf{A} = \mathbf{J}_S \quad \text{apgabalam } \Omega_2, \quad (4.21)$$

kur $\nu = 1/(\mu\mu_0)$. Ievietojot 4.17 robežnosacījumos (4.12) un (4.13), iegūstam

$$\mathbf{n}_1 \cdot \nabla \times \mathbf{A}_1 + \mathbf{n}_2 \cdot \nabla \times \mathbf{A}_2 = 0, \quad (4.22)$$

$$\nu \nabla \times \mathbf{A}_1 \times \mathbf{n}_1 + \nu \nabla \times \mathbf{A}_2 \times \mathbf{n}_2 = 0. \quad (4.23)$$

Pilnīgai vektorpotenciāla \mathbf{A} definēšanai, izmantosim Kulona nosacījumu (var izmantot arī citu \mathbf{A} normēšanu, taču šī ir visracionālākā)

$$\nabla \cdot \mathbf{A} = 0. \quad (4.24)$$

Šim nosacījumam jāparādās vienādojumos (4.20) un (4.21) kā $\nabla \nu \nabla \cdot \mathbf{A}$, proti,

$$\nabla \times \nu \nabla \times \mathbf{A} - \nabla \nu \nabla \cdot \mathbf{A} + \sigma \left\{ \frac{\partial \mathbf{A}}{\partial t} + \nabla V \right\} = 0 \quad \text{apgabalam } \Omega_1, \quad (4.25)$$

$$\nabla \times \nu \nabla \times \mathbf{A} - \nabla \nu \nabla \cdot \mathbf{A} = \mathbf{J}_S \quad \text{apgabalam } \Omega_2. \quad (4.26)$$

Visbeidzot, jāuzdzod Gausa likums strāvas blīvumam \mathbf{J} . Ievietojot vienādojumu (4.19) iekš (4.5), un nēmot diverģenci, iegūstam

$$\nabla \cdot \left\{ -\sigma \frac{\partial \mathbf{A}}{\partial t} - \sigma \nabla V \right\} = 0 \quad \text{apgabalam } \Omega_1. \quad (4.27)$$

Ja Ω_2 ir dielektriķis, tad strāvas plūsma caur virsmu S_{12} ir 0 jeb $\mathbf{J} \cdot \mathbf{n} = 0$, tad

$$\mathbf{n} \cdot \left\{ -\sigma \frac{\partial \mathbf{A}}{\partial t} - \sigma \nabla V \right\} = 0 \quad \text{apgabalam } \Omega_1. \quad (4.28)$$

Stacionāram elektriskajam un magnētiskajam laukam vienādojumi 4.26 un 4.27 vienkāršojas. Programmā `multiRegionPlasmaFoam` uzdotie vienādojumi apgabalam Ω_1 tiek pierakstīti formā:

```
(4.6)           solve( fvm::laplacian(phi) );
(4.7)+(4.9)   J=-sigma*fvc::grad(phi);
(4.25)+(4.11) solve( -fvm::laplacian(A) ==
                      (J+fvc::curl(Hc))*mu*constant::electromagnetic::mu0 );
(4.17)         B=fvc::curl(A);
```

Apgabalam Ω_2 :

```
(4.6)           solve( fvm::laplacian(phi) );
(4.7)           E=-fvc::grad(phi);
(4.25)          solve( fvm::laplacian(A) );
(4.17)          B=fvc::curl(A);
```

4.2.1. Bezgalīga apgabala robežnosacījumi

Teorētiskais apraksts

Ārējos robežnosacījumus iespējams uzdot ar telpas transformāciju palīdzību, viena tāda transformācija ir *Kelvina transformācija* [17]. Tieki izmantoti divi apaļi aprēķinu apgabali (konspektā tiek apskatīts 2D gadījums), kuru robežvirsmas ir saistītas. \mathbf{A} tiek uzdots polārajās koordinātās, otrajā apgabalā veikta rādiusa transformācija.

Transformāciju realizē arī programma COMSOL. Transformācijas paņēmiens saucas *Infinite elements*. Aprēķinu apgabalam tiek izveidots cits apgabals, kas veido “čaulu” dažu šūnu biezumā, kurā, visticamāk, nepārtrauktās vides vienādojumi tiek transformēti uz bezgalību. [18]

P. Testoni [13], *Z. Huang* [11] un citi autori, veicot magnētiskā lauka modelēšanu ar vektorpotenciālu, uz ārējā apgabala robežas izmanto Dirlhlē robežnosacījumus $\mathbf{A} = 0$, kas izriet no $\mathbf{B} \cdot \mathbf{n} = 0$ un $\mathbf{H} \times \mathbf{n} = 0$. Šī vienkāršā ārējo robežnosacījumu aproksimācija kroplo magnētisko lauku ārējo robežvirsmu tuvumā. Tā kā programma paredzēta elektronu un jonizētu atomu trajektoriju rekināšanai, magnētiskā lauka kroplojumi būtiski ietekmēs tieši vieglo elektronu trajektorijas.

David Meeker FEMM pamācībā [14] atrodama formula jaukti jeb Robina robežnosacījumu uzdošanai vektorpotenciālam uz ārējas aprēķinu apgabala robežas, kas imitē bezgalīgi lielu aprēķinu apgabalu:

$$\frac{1}{\mu\mu_0} \frac{\partial \mathbf{A}}{\partial \mathbf{n}} + c_0 \mathbf{A} + c_1 = 0, \quad (4.29)$$

$$c_0 = \frac{1}{\mu\mu_0 R}, \quad (4.30)$$

$$c_1 = 0, \quad (4.31)$$

kur R ir ārējās robežvirsmas izliekuma rādiuss, c_0 un c_1 ir konstantes.

Vienkāršojot (4.29), iegūstam

$$\frac{\partial \mathbf{A}}{\partial \mathbf{n}} + \frac{\mathbf{A}}{R} = 0 \quad (4.32)$$

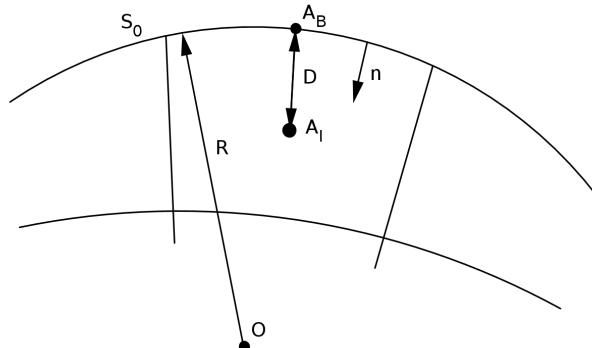
Ar \mathbf{A}_I apzīmēsim vektorpotenciāla vērtību režģa šūnas iekšpusē (4.2. att.), ar \mathbf{A}_B - vērtību uz ārējās robežvirsmas. Vērtībām atbilst līdzīgi nosaukti punkti 4.2. attēlā. Virsmas normāles vektors \mathbf{n} vērsts uz šūnas iekšpusi, D ir attālums starp punktiem \mathbf{A}_I un \mathbf{A}_B . Diskretizējot vienādojumu (4.32), iegūstam

$$\mathbf{A}_B = R \left(\frac{\mathbf{A}_I - \mathbf{A}_B}{D} \right). \quad (4.33)$$

Izsakot no (4.33) vienādojuma \mathbf{A}_B , iegūstam

$$\mathbf{A}_B = \frac{\mathbf{A}_I}{1 + \frac{D}{R}}. \quad (4.34)$$

Formulas (4.34) uzdošana `multiRegionPlasmaFoam` programmā



Att. 4.2: Robina robežnosacījumu uzdošana vektorpotenciālam.

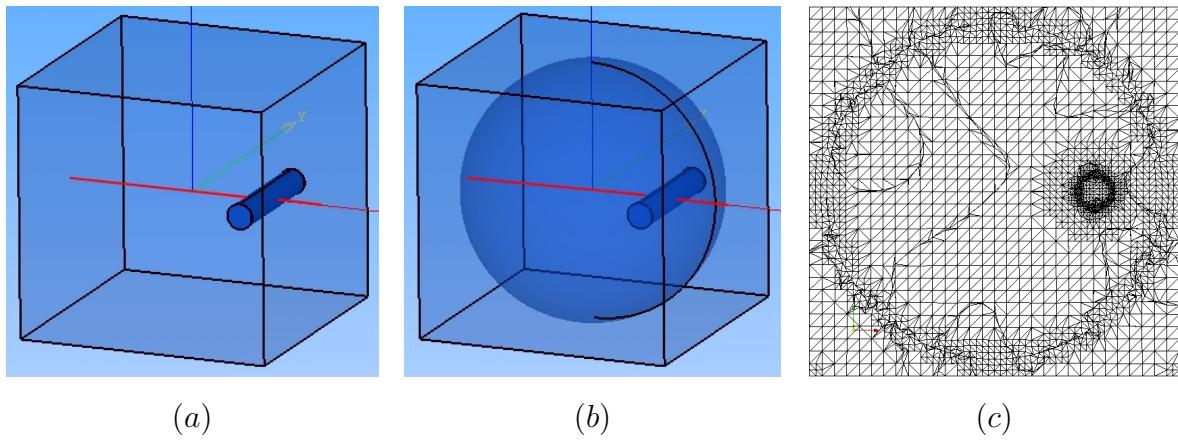
```
transform(I - n*n, intFld)/(1/(patch().deltaCoeffs()*radius_)+1)
```

Rezultāti

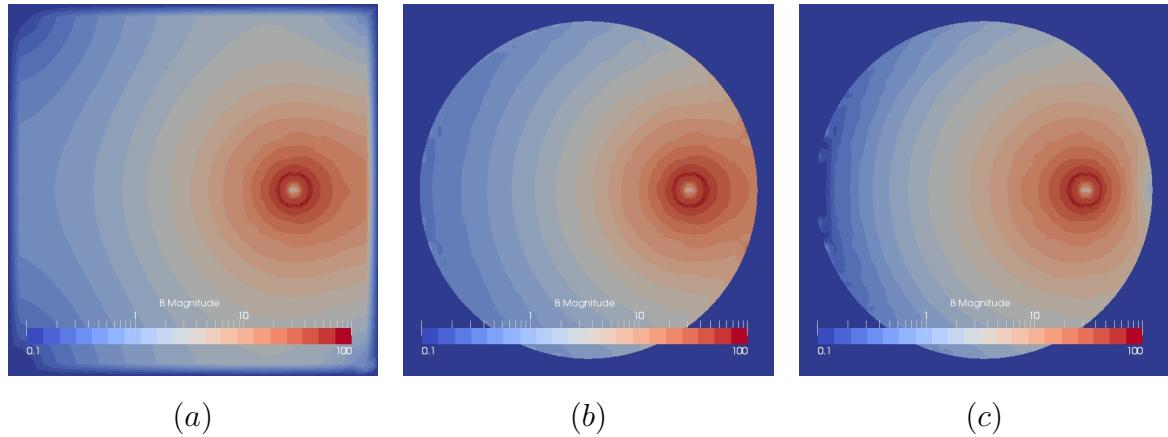
Ārējo robežnosacījumu salīdzināšanai tika veikti pieci testi, no kuriem divi testi veikti apgabalam ar Robina ārējiem robežnosacījumiem pie dažāda režģa smalcinājuma un iterāciju skaita.

Pirmajā testā tika modelēts magnētiskais lauks, kuru veido taisnā vada gabalā plūstoša strāva (skat. 4.3. att.). Vads atrodas tuvu aprēķinu apgabala ārējai virsmai. Testā tika salīdzinātas divas ārējās virsmas - kubs un sfēra, kā arī divi robežnosacījumu veidi - Dirihlē un Robina. Rezultāti attēloti logaritmiskā, ierobežotā no augšas un apakšas, skalā. Aprēķinus kvalitatīvi salīdzina pēc **B** un **A** lauka formas - izolīnijām jābūt koncentriskām ar vadu.

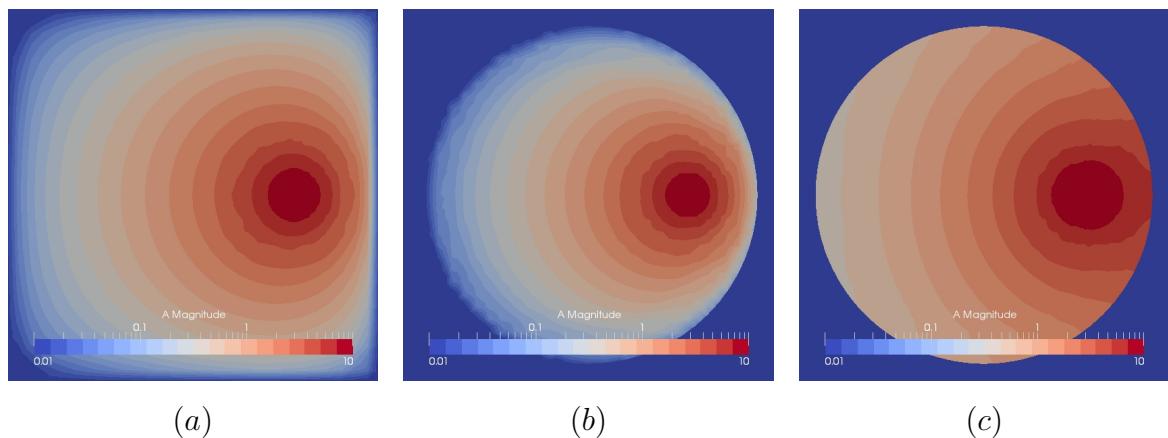
- Pirmajā aprēķinā kubiskai ārējai virsmai uzdoti Dirihlē robežnosacījumi. Rezultātus skat. 4.4. (a) un 4.5. (a) attēlā. **B** izolīnijas tālāk no vada un tuvāk ārējai robežvirsmai ir deformētas, magnētiskais lauks būtiski izkroplojis.
- Otrajā aprēķinā sfēriskai ārējai virsmai uzdoti Dirihlē robežnosacījumi. Rezultātus skat. 4.4. (b) un 4.5. (b) attēlā attēlā.
- Trešajā aprēķinā sfēriskai ārējai virsmai uzdoti Robina robežnosacījumi. Rezultātus skat. 4.4. (c) un 4.5. (c) attēlā attēlā. **B** un **A** izolīnijas ir koncentriskas, to forma deformēta robežvirsmu tuvumā. **B** aprēķinu kvalitāte trešajam aprēķinam ir līdzvērtīga otrajam, būtiska atšķirība vērojama **A** rezultātos, taču no šiem rezultātiem grūti spriest, kuri robežnosacījumi dod labākus rezultātus.



Att. 4.3: Pirmais tests: (a) Taisns vads kubiskā aprēķinu apgabalā, (b) sfēriskā aprēķinu apgabalā, (c) sfēriskam aprēķinu apgabalam atbilstošs režģis (62343 režģa šūnas).
Ar sarkanu līniju attēlotā “ x ” ass.



Att. 4.4: Pirmais tests, **B** amplitūda: (a) pirmais aprēķins - kubiskai ārējai virsmai uzdoti Dirihlē robežnos., (b) otrs aprēķins - sfēriskai ārējai virsmai uzdoti Dirihlē robežnos., (c) trešais aprēķins - sfēriskai ārejai virsmai uzdoti Robina robežnosacījumi.

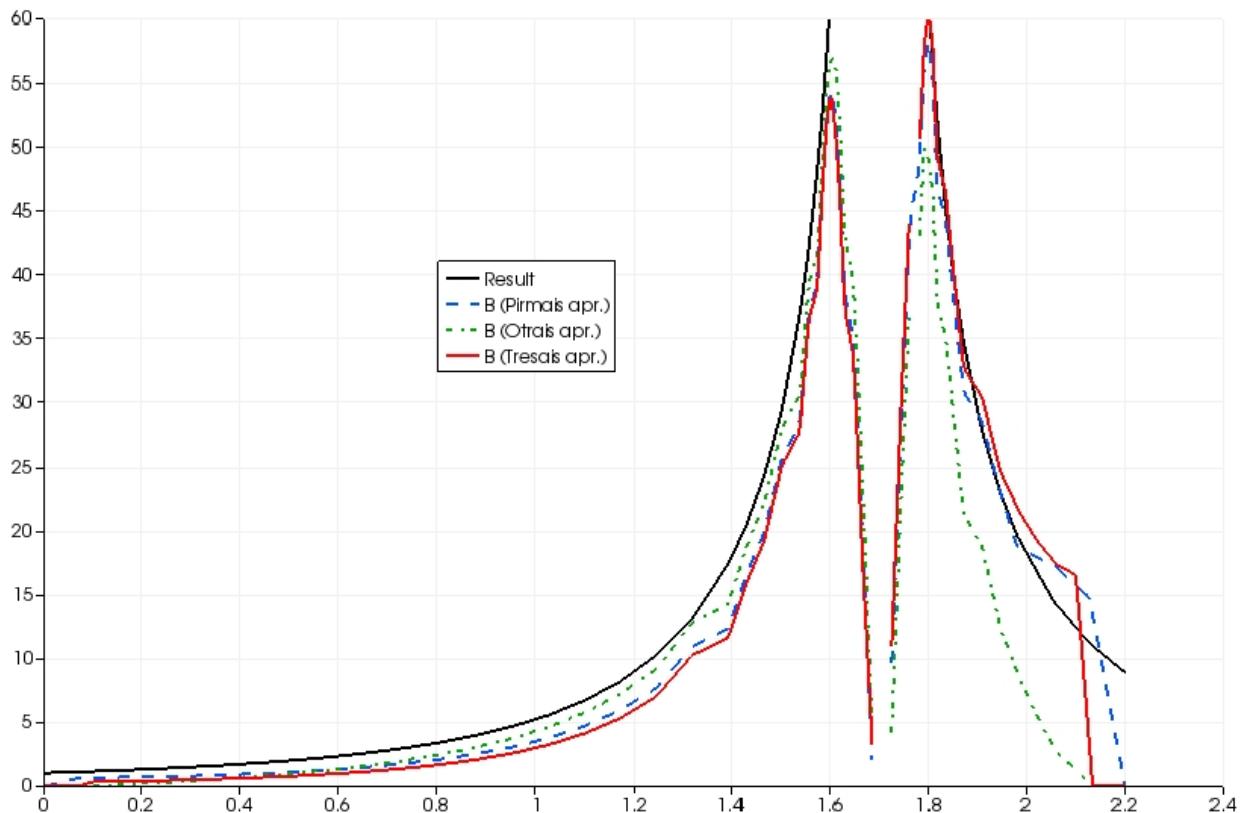


Att. 4.5: Pirmais tests, **A** amplitūda: (a) pirmais aprēķins - kubiskai ārējai virsmai uzdoti Dirihlē robežnos., (b) otrs aprēķins - sfēriskai ārējai virsmai uzdoti Dirihlē robežnos., (c) trešais aprēķins - sfēriskai ārejai virsmai uzdoti Robina robežnosacījumi.

B amplitūdas analītiskais atrisinājums ārpus vada:

$$\mathbf{B} = \frac{\mu_0 I}{4\pi x} \frac{1}{\sqrt{\frac{L}{2} + x^2}} \quad (4.35)$$

kur I ir vadā plūstošās strāvas stiprums, L - vada garums, x - attālums no vada centra. **B** amplitūdas vērtība uz "x" ass (4.3. (a) att. un turpmākajos attēlos attēlota ar sarkanu līniju) attēlota 4.6. attēlā, analītiskajam atrisinājumam atbilst vienādojums (4.35). No grafika redzam, ka pie visiem robežnosacījumiem rezultāti atšķiras no analītiskā atrisinājuma. Šīs neprecizitātes iemesls ir nelielais globālo iterāciju skaits (100 iterācijas), proti, programma rēķina magnētisko lauku katrā reģionā atsevišķi un tad ar robežnosacījumu palīdzību "savieno" reģionus. Ceturtais testā tiks veikta aprēķinu kvalitātes salīdzināšana pie dažāda iterāciju skaita.



Att. 4.6: Pirmais tests, **B** amplitūda uz "x" ass: "melna līkne" - analītiskais atrisinājums (4.35), "zilā līkne" - kubiskai ārējai virsmai uzdoti Dirihlē robežnos. (pirmais aprēķins), "zaļā līkne" - sferiskai ārējai virsmai uzdoti Dirihlē robežnos. (otrais aprēķins), "sarkanā līkne" - sferiskai ārējai virsmai uzdoti Robina robežnosacījumi (trešais aprēķins).

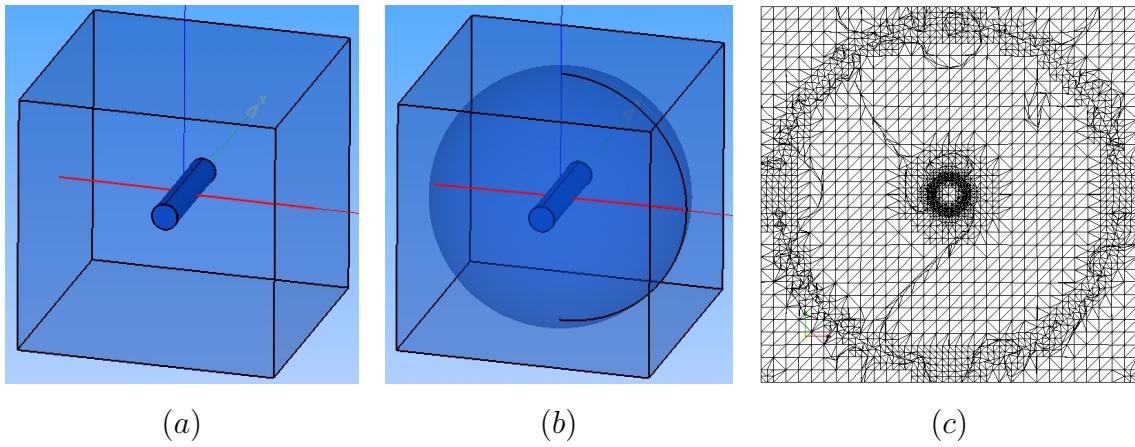
Otrajā testā, līdzīgi kā pirmajā, tika modelēts magnētiskais lauks, kuru veido taisnā vada gabalā plūstoša strāva (skat. 4.7. att.). Vads atrodas aprēķinu apgabala centrā. Testā tika salīdzinātas divas ārējās virsmas - kubs un sfēra, kā arī divi robežnosacījumu veidi - Dirihlē un Robina.

- Pirmajā aprēķinā kubiskai ārējai virsmai uzdoti Dirihlē robežnosacījumi.
- Otrajā aprēķinā sfēriskai ārējai virsmai uzdoti Dirihlē robežnosacījumi.
- Trešajā aprēķinā sfēriskai ārējai virsmai uzdoti Robina robežnosacījumi.

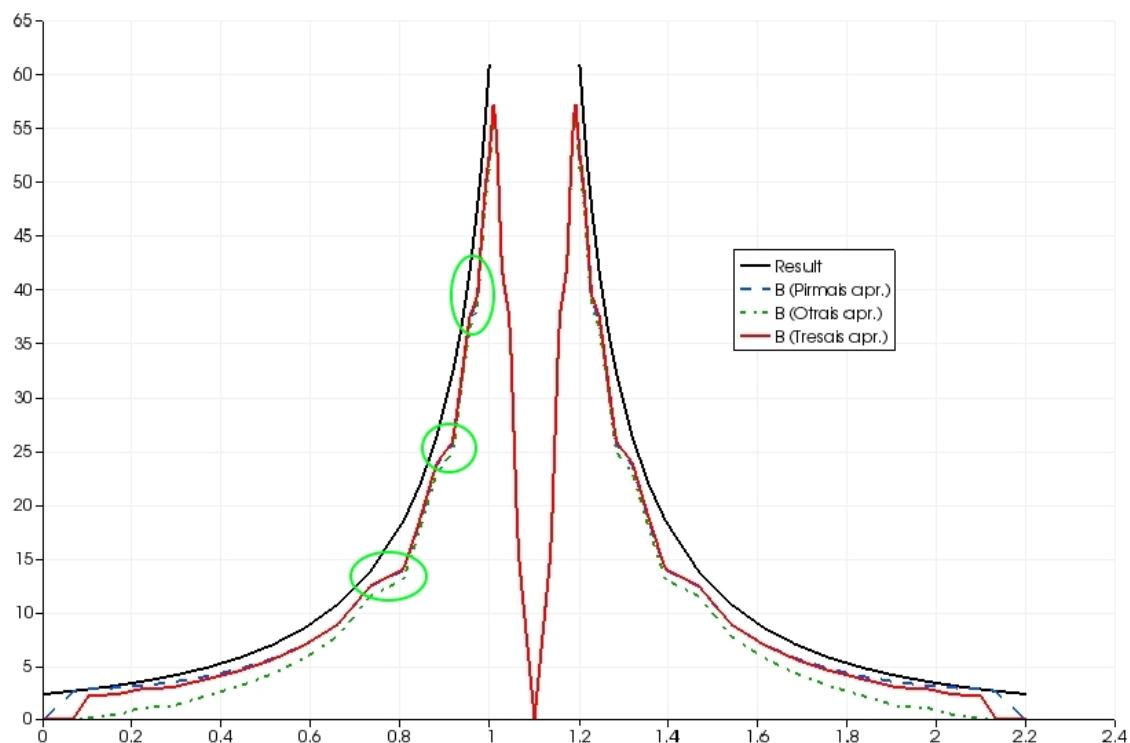
B amplitūdas vērtība uz “ x ” ass attēlota 4.8. attēlā, analītiskajam atrisinājumam atbilst vienādojums (4.35). Līdzīgi kā pirmajā testā, arī šajā skaitlisko aprēķinu rezultāti atšķiras no analītiskā atrisinājuma, kā iemesls ir nelielais laika soļu (globālo iterāciju) skaits (100 laika soļi).

Trešajā testā tika izmantota 4.7. (b) ģeometrija sfēriskam aprēķinu apgabalam ar centrā novietotu vadu, tika veikti 3 aprēķini ar Robina ārējiem robežnosacījumiem pie dažādiem režģa smalcinājumiem. Pirmajā aprēķinā **blockMesh** režģis ir $20 \times 20 \times 20$ (pēc **snappyHexMesh** 51680 šūnas), otrajā aprēķinā **blockMesh** režģis ir $30 \times 30 \times 30$ (pēc **snappyHexMesh** 89664 šūnas), trešajā aprēķinā **blockMesh** režģis ir $50 \times 50 \times 50$ (pēc **snappyHexMesh** 418734 šūnas). Rezultātus skat. 4.10. attēlā. No rezultātiem redzam, ka uz smalka režģa rezultāti, salīdzinot ar analītisko risinājumu, ir sliktāki, nekā uz rupja režģa. Iemesls tam ir nelielais globālo iterāciju skaits (100 iterācijas).

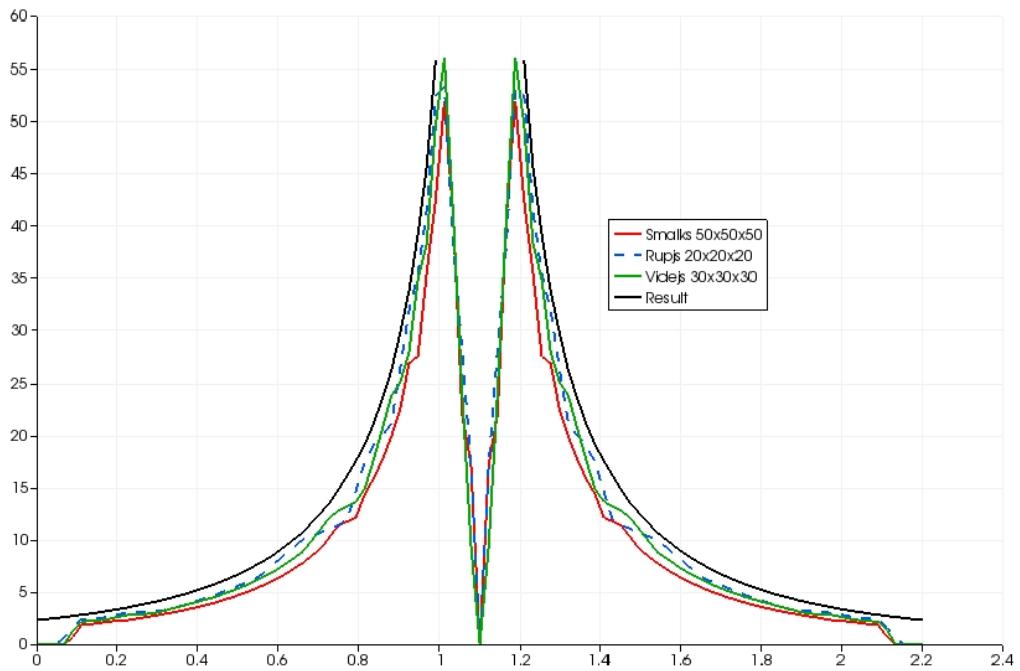
Ceturtajā testā tika izmantota 4.7. (b) ģeometrija sfēriskam aprēķinu apgabalam ar centrā novietotu vadu un smalku režģi, tika veikti 3 aprēķini ar Robina ārējiem robežnosacījumiem pie dažāda globālo iterāciju skaita. Pirmajā aprēķinā **blockMesh** režģis ir $20 \times 20 \times 20$ (pēc **snappyHexMesh** 51680 šūnas), otrajā aprēķinā **blockMesh** režģis ir $30 \times 30 \times 30$ (pēc **snappyHexMesh** 89664 šūnas), trešajā aprēķinā **blockMesh** režģis ir $50 \times 50 \times 50$ (pēc **snappyHexMesh** 418734 šūnas). Rezultātus skat. 4.10. attēlā. No rezultātiem redzam, ka uz smalka režģa rezultāti, salīdzinot ar analītisko risinājumu, ir sliktāki, nekā uz rupja režģa. Iemesls tam ir nelielais laika soļu (globālo iterāciju) skaits (100 iterācijas).



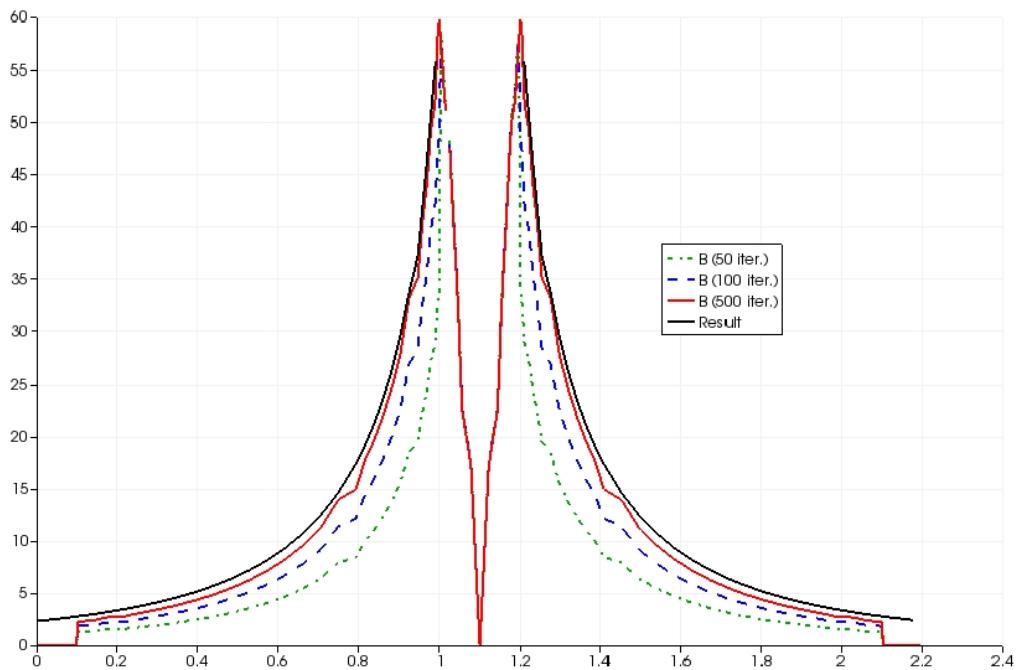
Att. 4.7: Otrais tests: (a) Taisns vads kubiskā aprēķinu apgabalā, (b) sfēriskā aprēķinu apgabalā, (c) sfēriskam aprēķinu apgabalam atbilstošs režģis (89664 režģa šūnas).
Ar sarkanu līniju attēlota “ x ” ass.



Att. 4.8: Otrais tests, \mathbf{B} amplitūda uz “ x ” ass: “melna līkne” - analītiskais atrisinājums (4.35), “zilā līkne” - kubiskai ārējai virsmai uzdoti Dirihlē robežnos. (pirmais aprēķins), “zaļā līkne” - sfēriskai ārējai virsmai uzdoti Dirihlē robežnos. (otrais aprēķins), “sarkanā līkne” - sfēriskai ārējai virsmai uzdoti Robina robežnosacījumi (trešais aprēķins). Ar zaļajām elipsēm atzīmēti tie skaitliskā atrisinājuma līknes posmi, kur uz režģa izmainās smalcinājuma kārta. Uz šīs robežas režģa šūnu lineārie izmēri mainās 2 reizes.



Att. 4.9: Trešais tests, \mathbf{B} amplitūda uz “ x ” ass: “melna līkne” - analītiskais atrisinājums (4.35), “zilā līkne” - rupjš režģis (`blockMesh` uzdots $20 \times 20 \times 20$), “zaļā līkne” - vidējs režģis (`blockMesh` uzdots $30 \times 30 \times 30$), “sarkanā līkne” - smalks režģis (`blockMesh` uzdots $50 \times 50 \times 50$).



Att. 4.10: Ceturtais tests smalkam režģim, \mathbf{B} amplitūda uz “ x ” ass: “melna līkne” - analītiskais atrisinājums (4.35), “zaļā līkne” - 50 globālās iterācijas, “zilā līkne” - 100 globālās iterācijas, “sarkanā līkne” - 500 globālās iterācijas.

Piektajā testā tika modelēts magnētiskais lauks, kuru veido gredzenveida vadā plūstoša strāva (skat. 4.11. att.). Testā tika salīdzināti divi robežnosacījumu veidi - Diriħlē un Robina, kas uzdoti sfēriskam ārējam apgabalam. `multiRegionPlasmaTest` aprēķiniem amplitūda ierobežota no augšas un apakšas, lai atbilstu FEMM aprēķina rezultātu skalai. Tests paredz kvalitatīvu rezultātu analīzi.

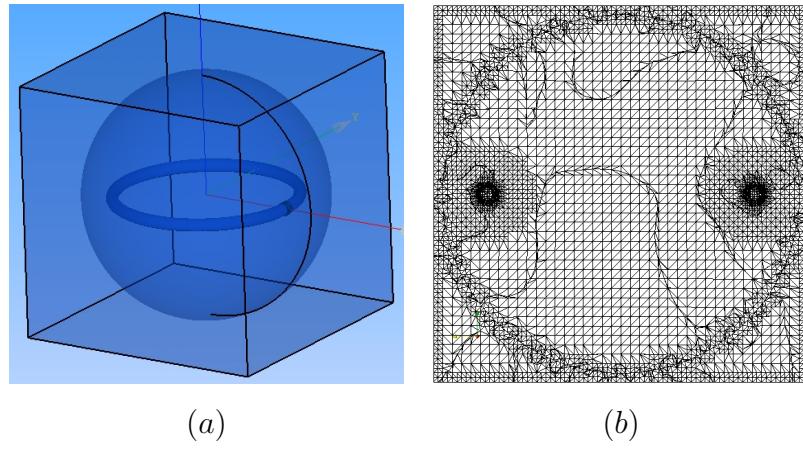
- Pirmajā aprēķinā sfēriskai ārējai virsmai uzdoti Diriħlē robežnosacījumi. Rezultātus skat. 4.12. (a) attēlā.
- Otrajā aprēķinā sfēriskai ārējai virsmai uzdoti Robina robežnosacījumi. Rezultātus skat. 4.12. (b) attēlā.
- Trešais aprēķins veikts ar FEMM programmu. Sfēriskai ārējai virsmai uzdoti Robina robežnosacījumi. Šajā aprēķinā vadā plūstošās strāvas blīvums **J** nemts kā pirmajā un otrajā aprēķinā caur vada šķērsgriezumu plūstošās strāvas vidējais blīvums **J**. Rezultātus skat. 4.12. (c) attēlā.

Otrā un trešā aprēķina rezultāti kvalitatīvi sakrīt. Papildus aprēķins ļauj pārliecināties, ka, FEMM programmā ļoti lielam sfēriskam ārējam apgabalam uzdodot Diriħlē robežnosacījumus, iegūsim Robina robežnosacījumiem līdzīgu rezultātu. Tika ņemta sfēra ar 5 reizes lielāku rādiusu, rezultātus skat. 4.13. attēlā. **B** amplitūda tika ierobežota no augšas un apakšas, lai atbilstu aprēķiniem 4.12. attēlā.

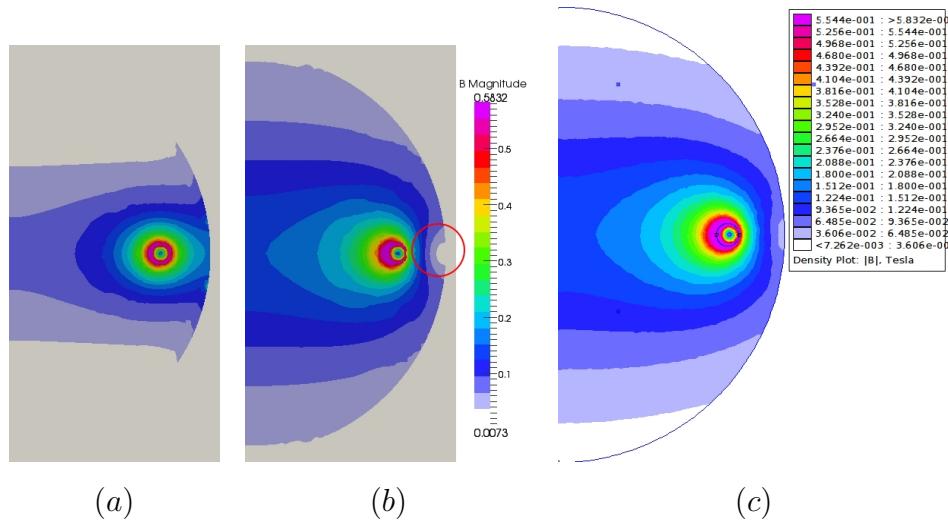
Secinājumi

Robina robežnosacījumi tiek uzdoti **A** ārējās virsmas normālajai komponentei, līdz ar to tangenciālā komponente uz robežvirsmas ir 0, jeb $\mathbf{A} \times \mathbf{n} = 0$. Strādājot ar šiem robežnosacījumiem, var samazināt rēķināmā apgabala izmēru, taču magnētiskā lauka avoti jāliek pēc iespējas tuvāk sfēriskā apgabala centram. Attēlā 4.12. (b) ar sarkanu apli apvilkts apgabals, kur redzams, ka robežvirsmu tuvumā **B** ir ielieks uz iekšu, līdzīga aina novērojama arī FEMM aprēķinā 4.12. (c) attēlā, jo abos gadījumos izmantoti vienādi robežnosacījumi. Šīs neprecizitātes iemesls, visticamāk, ir saistīts ar to, ka vienādojums (4.29) "paredz", ka magnētiskā lauka avoti atrodas sfēras centrā. Robežnosacījumu testu gadījumā magnētiskā lauka avoti tika apzināti novietoti ārējo robežnosacījumu tuvumā, lai kvalitatīvi pārliecinātos par aprēķinu kvalitāti to tuvumā.

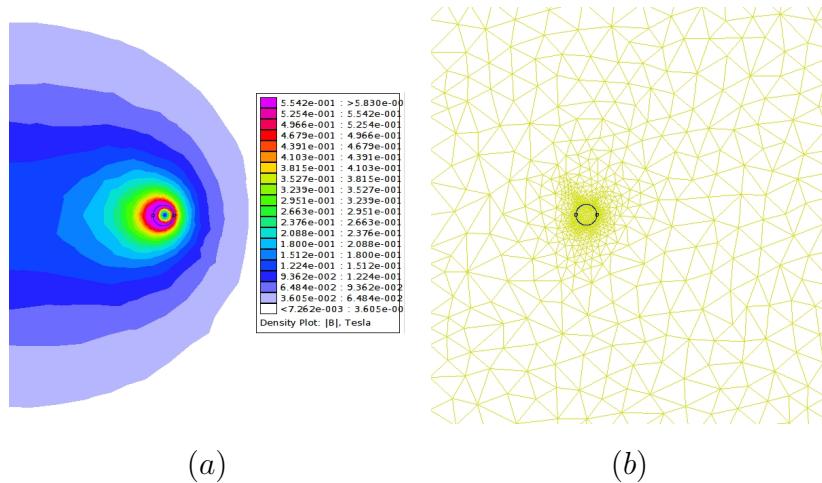
Rēķinot stacionāru magnētisko lauku, ātrākai risinājuma konverģencei, jāsamazina konvergences kritērijus un laika soli (globālo iterāciju skaitu), tas ļaus multi-reģionu programmai biežāk korigēt robežnosacījumus un globālā atrisinājuma konvergēnce uzlabosies.



Att. 4.11: (a) Gredzenveida vads sfēriskā ārējā aprēķinu apgabalā, (c) atbilstošs režģis (453564 režģa šūnas).



Att. 4.12: Magnētiskā lauka indukcija \mathbf{B} pie dažādām ārējām robežvirsmām un robežnosacījumiem: (a) - sfēra, Dirihiļē, (b) - sfēra, Robina robež. nos., (c) - FEMM risinājums. Attēlā (b) ar sarkanu apli izcelta \mathbf{B} klūda, līdzīga neprecizitāte ir arī FEMM aprēķinam (c) attēlā.



Att. 4.13: (a) FEMM risinājums lielam sfēriskam ārējam apgabalam ar Dirihiļē robežnosacījumiem, (b) atbilstošs režģa fragments (aplūkotā apgabala režģis ir rupjš).

4.2.2. Robežnosacījumi uz apgabalu robežvirsmām

Teorētiskais apraksts

Uz robežvirsmas S_{12} izpildās vienādojumi (4.22), (4.24) un (4.28). No (4.22) un (4.24) redzam, ka, zinot \mathbf{A}_1 vērtību, nav iespējams aprēķināt \mathbf{A}_2 vērtību un otrādi, proti, nav iespējams pilnīgi restaurēt \mathbf{A} vērtību, zinot \mathbf{B} jeb $\nabla \times \mathbf{A}$. Tāpēc vienkāršos modeļos robežnosacījumus aplūko kā speciālgadījumu, kad $\mu_1/\mu_2 = 0$, proti, Ω_1 ir diamagnētiķis un Ω_2 ir feromagnētiķis, sekojoši, magnētiskā lauka plūsmas līnijas uz robežvirsmas S_{12} apgabala Ω_1 pusē ir perpendikulāras robežvirsmai. Ņemot vērā “vektorpotenciāla nepārtrauktības nosacījumu” jeb Kulona nosacījumu (4.24) tiek lietoti šādi robežnosacījumi:

$$\frac{\partial \mathbf{A}_2}{\partial \mathbf{n}} = 0 \quad \text{apgabalam } \Omega_1, \quad (4.36)$$

$$\mathbf{A}_1 = \mathbf{A}_2 \quad \text{apgabalam } \Omega_2. \quad (4.37)$$

Tātad, uz apgabala Ω_1 robežas dominē “plūsmas līniju perpendikularitāte”, bet uz Ω_2 robežas “vektorpotenciāla nepārtrauktības nosacījums”.

Rezultāti

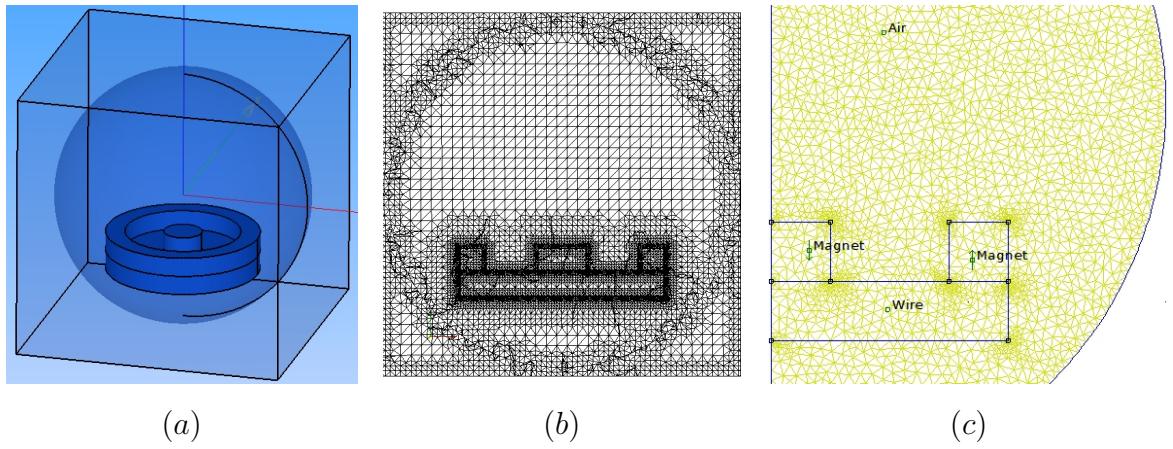
Robežnosacījumu pārbaudei tika veikti divi testi.

Pirmajā testā tika modelēts cirkulāra magnetrona piemērs (skat. 4.14. att.), tas ir, feromagnētisks disks, uz kura atrodas viens cilindrisks un viens gredzenveida magnēts, kuru magnetizācijas ir pretēji vērstas. `multiRegionPlasmaTest` rezultāti (4.15. att.) salīdzināti ar FEMM aksiālsimetrisku aprēķinu (4.16. att.). `multiRegionPlasmaTest` \mathbf{B} lauka amplitūda ierobežota atbilstoši FEMM amplitūdai. Rezultāti sakrīt, tomēr `multiRegionPlasmaTest` magnētiem nespēj attēlot strauju \mathbf{B} izmaiņu, kā to attēlo FEMM. Tests nākamajā nodalā labāk ilustrē šo problēmu, jo režģis uz magnētu iekšējām virsmām nav tik smalks (piemēru skat. 4.21. (a) att.). No vienādojumiem (4.11) un (4.26) izriet, ka magnētu radīto \mathbf{A} rēķina kā

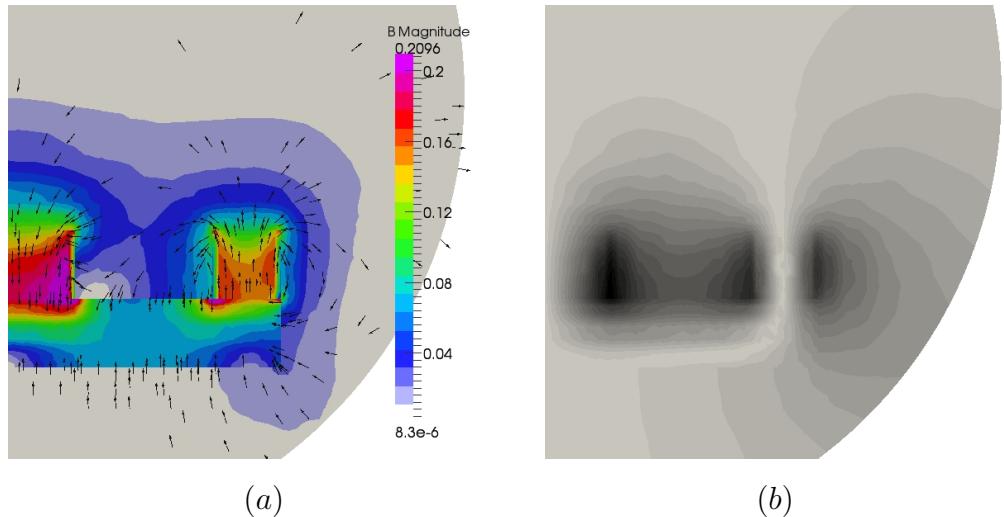
$$\Delta \mathbf{A} = \mu \mu_0 \nabla \times \mathbf{M}, \quad (4.38)$$

kur \mathbf{M} ir magnēta magnetizācijas vektors.

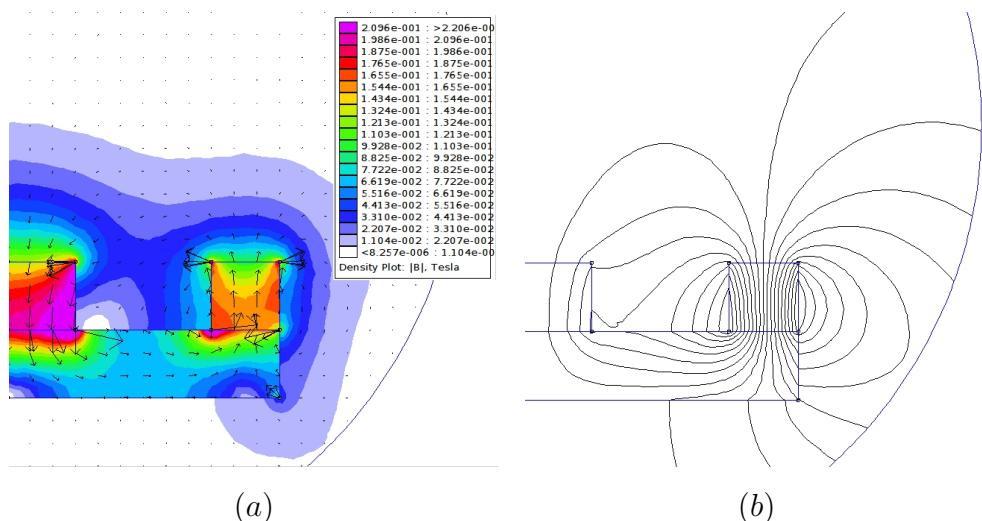
`OpenFOAM` rotora operācijas diskretizācija nenodrošina atbilstošas \mathbf{A} vērtības rēķināšanu uz robežvirsmām vai arī tās ekstrapolāciju uz tām, tāpēc uz magnētu sānu virsmām ir \mathbf{B} lineārs kritums viena režīga slāņa biezumā. Šī slāņa ietekmi uz piemēra aprēķinu var mazināt, smalcinot režīgi uz magnētu iekšējām virsmām. Iespējams, šo problēmu var novērst, rotoram lietojot nevis “Gausa lineāro”, bet kādu citu diskretizācijas shēmu.



Att. 4.14: (a) Cirkulārs magnetrons (feromagnētisks disks, uz kura atrodas viens cilindriskis un viens gredzenveida magnēts, kuru magnetizācijas ir pretejī vērstas), (b) režģis (585543 režģa šūnas), (c) FEMM piemērs (3674 režģa šūnas).



Att. 4.15: Cirkulāra magnetrona `multiRegionPlasmaTest` aprēķins: (a) **B** amplitūda un nenormēti vektori, (b) **A** amplitūdas sadalījums.

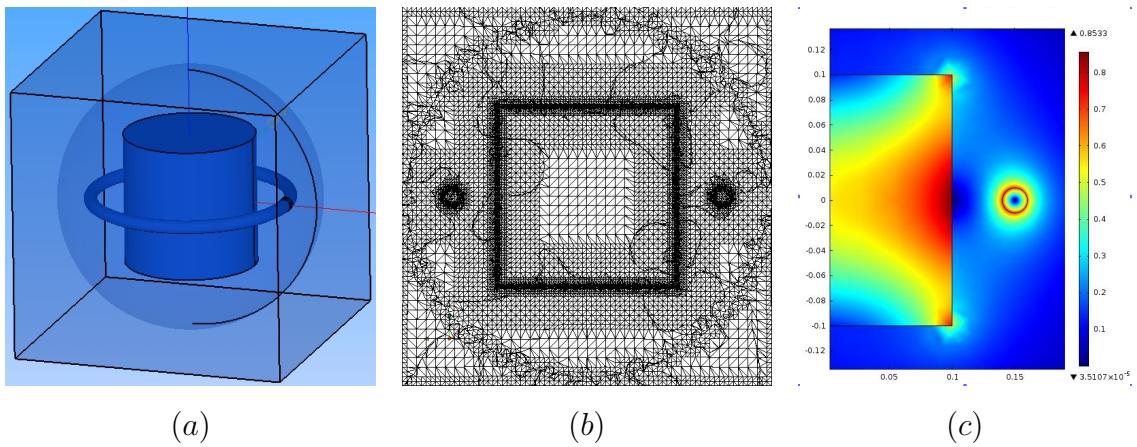


Att. 4.16: Cirkulāra magnetrona FEMM aksiālsimetrisks aprēķins: (a) **B** amplitūda un normēti vektori, (b) magnētiskās plūsmas līnijas.

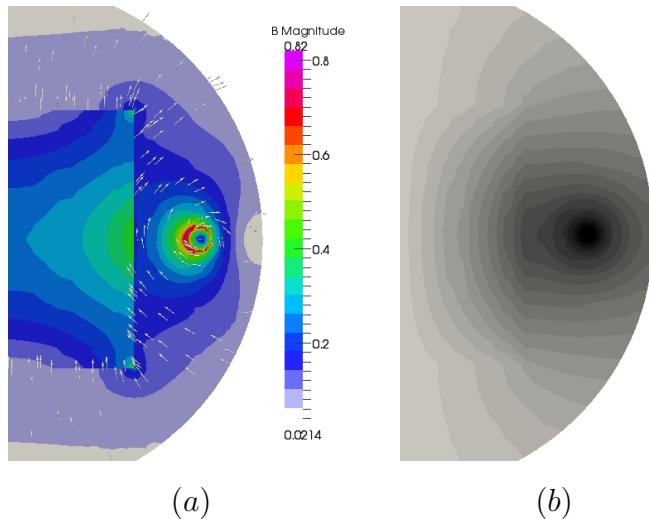
Otrais tests sastāv no gredzenveida vada, kurā plūst līdzstrāva, un feromagnētiska cilindra, proti, vienkāršota spole (skat. 4.17. att.). `multiRegionPlasmaTest` rezultāti (4.18. att.) salīdzināti ar FEMM aksiālsimetrisku aprēķinu (4.19. att.). Feromagnētiskā cilindra iekšpusē **B** intensitāte ir aptuveni divreiz mazāka kā FEMM aprēķinā, turpretim ārpus cilindra neveidojas apgabals ar mazu **B** intensitāti.

Analizējot šo, kā arī citus kvalitatīvi labākus risinājumus, tika konstatēts, ka potenciālās problēmas rodas, kad magnētiskās indukcijas vektors **B** ir paralēls robežvirsmai. Iespējams, šie vienkāršotie robežnosacījumi neņem vērā kādu no **A** komponentēm vai gradientiem. Ekvivalenti piemērs tika sarēķināts ar programmu COMSOL (4.17. (c). att.).

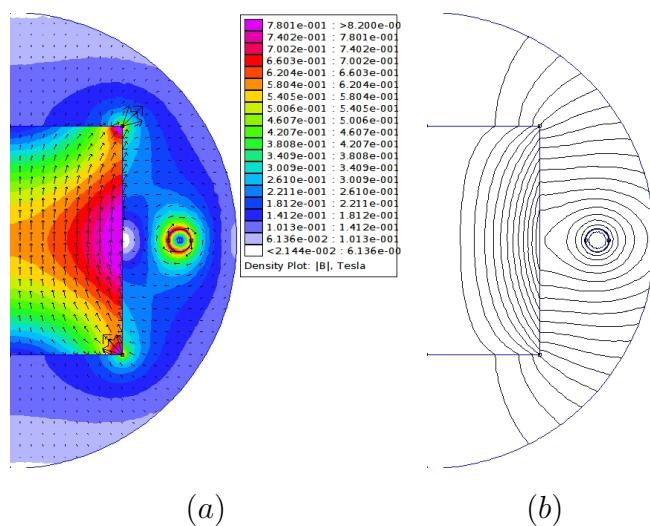
Magnetronu uzputināšanas sistēmās primāri magnētiskā lauka avoti ir magnēti. Magnētiskais lauks, kas iziet no magnētiem, ir perpendikulārs virsmai, tāpēc šo robežnosacījumu magnētiskā lauka kropļojumi magnētu polu apgabalos nepastāv. Kā tālāk rādis aprēķini 4.15. un 5.12. attēlos, šie vienkāršie robežnosacījumi labi tiek galā ar vienkāršām magnetronu konstrukcijām, problēmas sagādā feromagnētiski mērķi.



Att. 4.17: (a) Gredzenveida vads ap feromagnētisku serdi, (b) režģis, (c) COMSOL magnētiskā lauka indukcijas \mathbf{B} aprēķins.



Att. 4.18: `multiRegionPlasmaTest` aprēķins gredzenveida vadam ap feromagnētisku serdi: (a) \mathbf{B} amplitūda un nenormēti vektori, (b) \mathbf{A} amplitūdas sadalījums.



Att. 4.19: FEMM aksiālsimetrisks aprēķins gredzenveida vadam ap feromagnētisku serdi:
 (a) \mathbf{B} amplitūda un normēti vektori, (b) magnētiskās plūsmas līnijas

4.2.3. Magnētiskās caurlaidības nelinearitāte

Teorētiskais apraksts

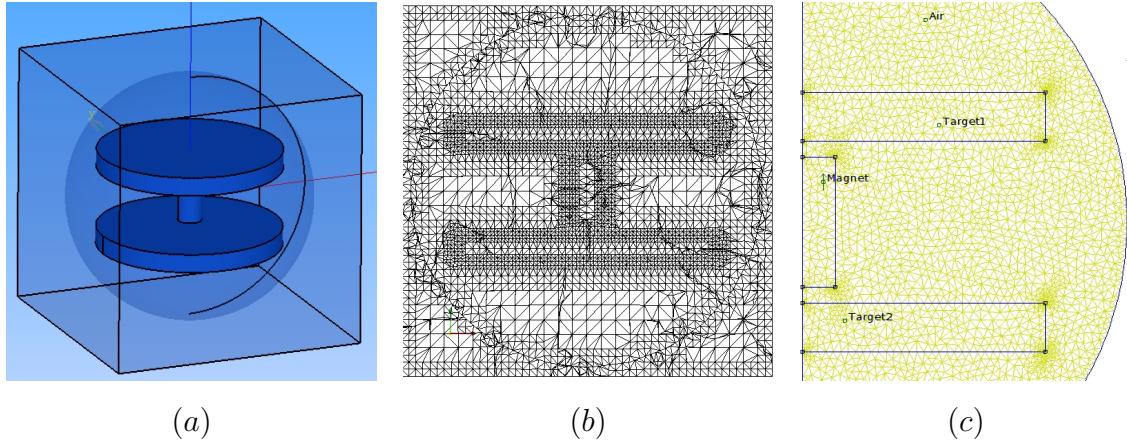
Feromagnētiķa magnētiskās īpašības magnētiskajā laukā ir atkarīgas no magnētiskās histerēzes, proti magnētiskā lauka intensitātes \mathbf{H} atkarības no indukcijas \mathbf{B} . Aprēķinu programmas parasti ielasa tabulu ar \mathbf{H} atkarību no \mathbf{B} , tad interpolē to. Taču skaitliskos aprēķinos uzskatāmāk lietot relatīvo magnētisko caurlaidību μ , jo tā tieši ietilpst vienādojumos. μ dažādiem materiāliem atšķiras, tam nevar piekārtot universālu formulu, tāpēc μ atkarība no \mathbf{B} arī jāielasa tabulas veidā. Tabulu ielasīšana un interpolēšana ietilpst OpenFOAM standarta rīkos, parametru ielasīšanu no teksta failiem OpenFOAM aprakstīja Anja Miehe [22]. μ atkarība no \mathbf{B} tiek ielasīta programmas sākumā tabulas veidā ar OpenFOAM funkcijas *graph* palīdzību. Pēc katras \mathbf{B} iterācijas tiek veikta *graph* tabulas interpolācija ar funkciju *interpolateXY* un jaunu μ vērtību noteikšana.

Rezultāti

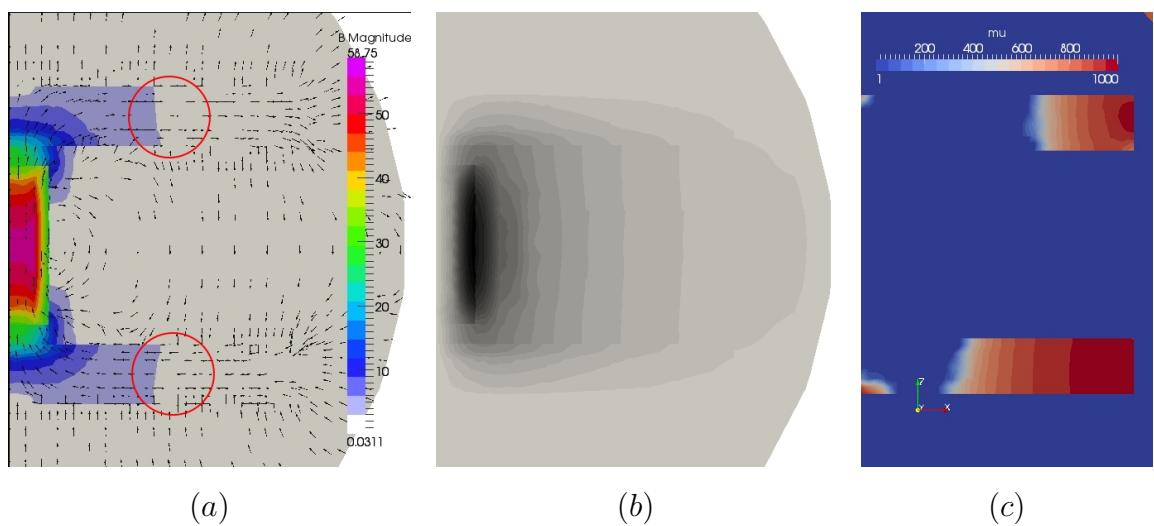
Magnētiskās caurlaidības nelinearitātes pārbaudei tika veikts viens tests.

Cilindrisks magnēts atrodas aprēķinu apgabala centrā, virs un zem tā atrodas divas feromagnētiskas plāksnes ar dažādām μ atkarībām no \mathbf{B} (4.20. att.). `multiRegionPlasmaTest` ielasīto failu saturu jeb $\mu(\mathbf{B})$ vērtības skat. 8.4.. pielikumā. Abiem mērķiem ir vienādas sākotnējās vērtības $\mu = 1000$. Augšējais mērķis “piesātinās” pie 2 T, apakšējais pie 4 T. Šīs vērtības ir ģenerētas ar *web* rīku [23], kā arī patstāvīgi papildinātas līdz lielākai \mathbf{B} vērtībai.

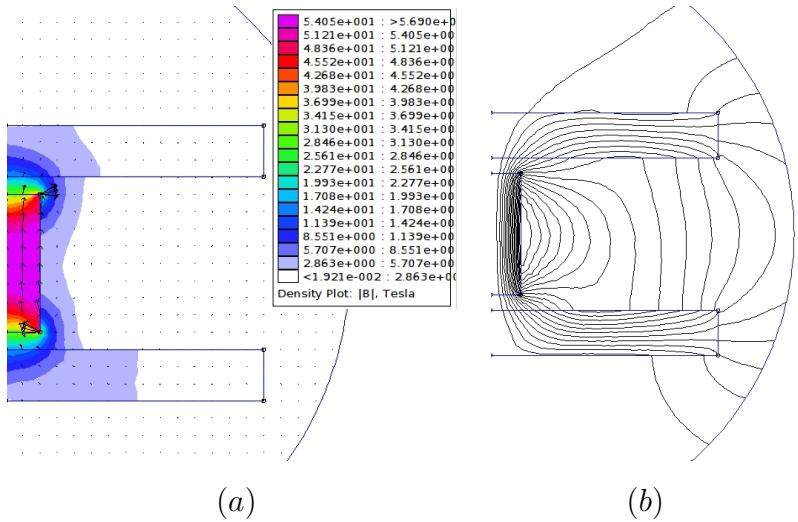
`multiRegionPlasmaTest` rezultāts attēlots 4.21. attēlā, atbilstoša FEMM aprēķina rezultāts (4.22. attēlā). Attēlā 4.21. (a) ar sarkaniem riņķiem atzīmēti divu dažādu feromagnētiķu (atšķiras magnetizācijas piesātinājuma robeža) apgabali, kuros B ir vienāds. Šis rezultāts ir līdzīgs tam, kādu iegūst, ja abiem feromagnētiķiem uzdod konstantu $\mu = 1000$, šāda aprēķina rezultātu skat. 4.23. (a) attēlā. Šīs problēmas cēlonis ir pieņēmums, ka diamagnētiķa un feromagnētiķa μ attiecība ir 0 (skat. 4.2.2. nodaļu). Nākamajā nodaļā šī problēma tiek atrisināta.



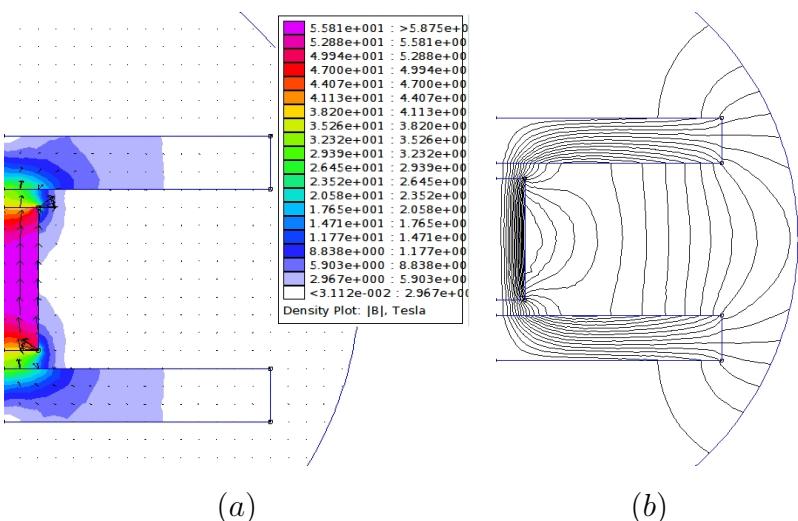
Att. 4.20: (a) Cilindrisks magnēts un divi feromagnētiski diskī, (b) režģis (176672 režģa šūnas), (c) FEMM piemērs (3541 režģa šūnas).



Att. 4.21: `multiRegionPlasmaTest` aprēķins cilindriskam magnētam un diviem feromagnētiskiem diskiem: (a) \mathbf{B} amplitūda un nenormēti vektori, (b) \mathbf{A} amplitūdas sadalījums, (c) μ sadalījums. Ar sarkaniem riņķiem atzīmēti feromagnētiskā diska apgabali, kuros \mathbf{B} nav atbilstošs ekvivalentam FEMM aprēķinam 4.22. (a) attelā.



Att. 4.22: FEMM aprēķins cilindriskam magnētam un diviem feromagnētiskiem diskiem ar μ atkarību no \mathbf{B} : (a) \mathbf{B} amplitūda un normēti vektori, (b) magnētiskās plūsmas līnijas.



Att. 4.23: FEMM aprēķins cilindriskam magnētam un diviem feromagnētiskiem diskiem ar konstantu μ (a) \mathbf{B} amplitūda un normēti vektori, (b) magnētiskās plūsmas līnijas.

4.2.4. Jauktie robežnosacījumi uz apgabalu robežvirsmām

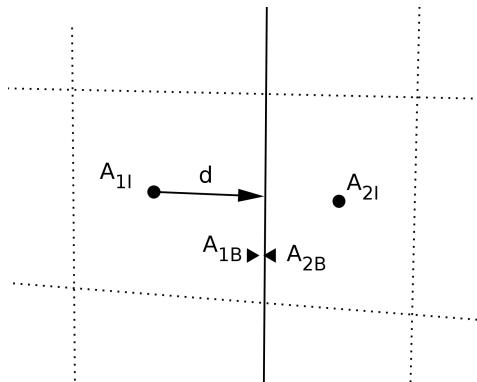
Teorētiskais apraksts

Lietojot vienkāršotos robežnosacījumus, kas aprakstīti 4.2.2. apaksnodaļā, potenciālas problēmas rodas, rēķinot laukus feromagnētiskiem materiāliem ar dažādiem μ . Iepriekš aplūkotie robežnosacījumi ir speciālgadījums, kad magnētiskā caurlaidība feromagnētikim ir bezgalība. Tai pat laikā programmai jāspēj rēķināt laukus dažādiem feromagnētikiem ar kopīgu virsmu.

Par jaukto robežnosacījumu pamatu tiek izmantoti OpenFOAM robežnosacījumi "mixed". Tajos tiek uzdotas trīs vērtības: *refValue*, *refGrad* un *valueFraction*.

- **refValue** tiek uzdoti Diriħlē robežnosacījumi. Modeļa gadījumā raksturīgā vērtība ir \mathbf{A}_{2I} jeb $\mathbf{A}_{1B} = \mathbf{A}_{2I}$ (4.24. att.);
- **refGrad** tiek uzdoti Neimana robežnosacījumi. Modeļa gadījumā raksturīgā gradiента vērtība ir 0 jeb $\frac{\partial \mathbf{A}_2}{\partial \mathbf{n}} = 0$;
- **valueFraction** tiek uzdota proporcija starp *refValue* un *refGrad*, kuru rēķina kā μ_1/μ_2 , ierobežojot maksimālo vērtību līdz 1, proti,

$$\text{valueFraction} = \begin{cases} \mu_1/\mu_2 & \text{ja } \mu_1/\mu_2 < 1 \\ 1 & \text{ja } \mu_1/\mu_2 \geq 1 \end{cases}$$



Att. 4.24: Jauktu robežnosacījumu uzdošana vektorpotenciālam. Vērtības režģa šūnu centros (\mathbf{A}_{1I} , \mathbf{A}_{2I}), vērtības uz robežvirsmas (\mathbf{A}_{1B} , \mathbf{A}_{2B}), attālums no šūnas centra līdz robežvirsmai d .

`multiRegionPlasmaFoam` “mixed” robežnosacījumi tiek diskretizēti sekojoši:

```
valueFraction_*refValue_+(1.0 - valueFraction_)*  
(this->patchInternalField() + refGrad_/this->patch().deltaCoeffs())
```

Šeit `patchInternalField` ir \mathbf{A}_{2I} un $1/this->patch().deltaCoeffs()$ ir d - attālums no režģa šūnas centra līdz robežvirsmai (4.24. att.). Šos robežnosacījumus var vienkāršot:

```
K()*nbrIntFld + (1.0 - K())*this->patchInternalField()
```

$nbrIntFld$ ir \mathbf{A}_{2I} , `this->patchInternalField()` ir \mathbf{A}_{1I} un $K()$ ir μ_1/μ_2 , ierobežots līdz 1.

Rezultāti

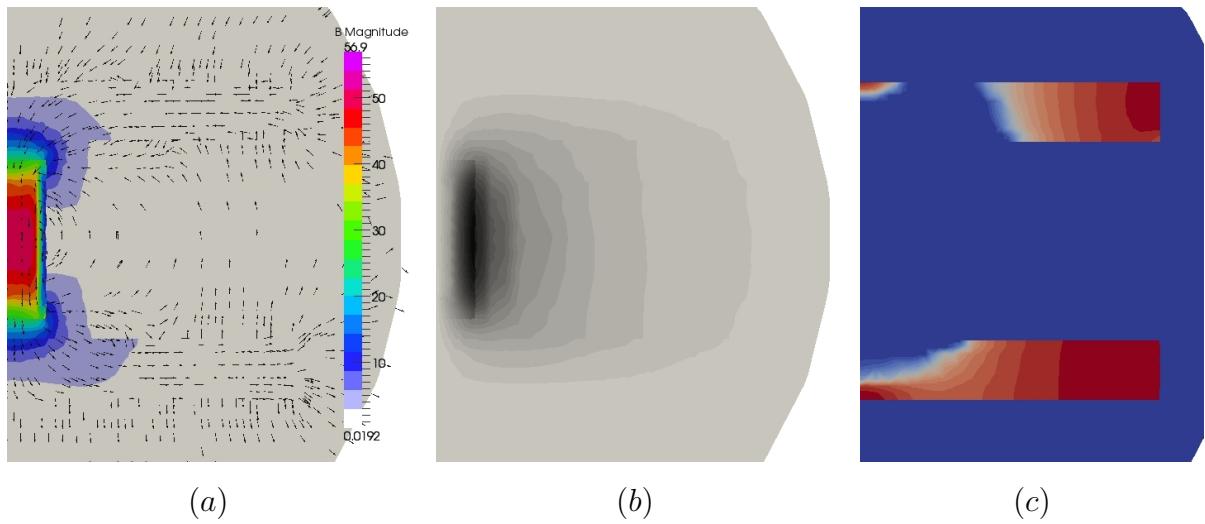
Jaukti robežnosacījumu pārbaudei tika veikti divi testi.

Pirmajā testā tiek atkārtots iepriekšējā apakšnodalā veiktais aprēķins cilindriskam magnētam un diviem diskveida feromagnētiem (4.20. att.). `multiRegionPlasmaTest` aprēķins (4.25. att.) un atbilstošs FEMM aprēķins (4.26. att.). `multiRegionPlasmaTest` rezultāts kvalitatīvi atbilst FEMM rezultātam.

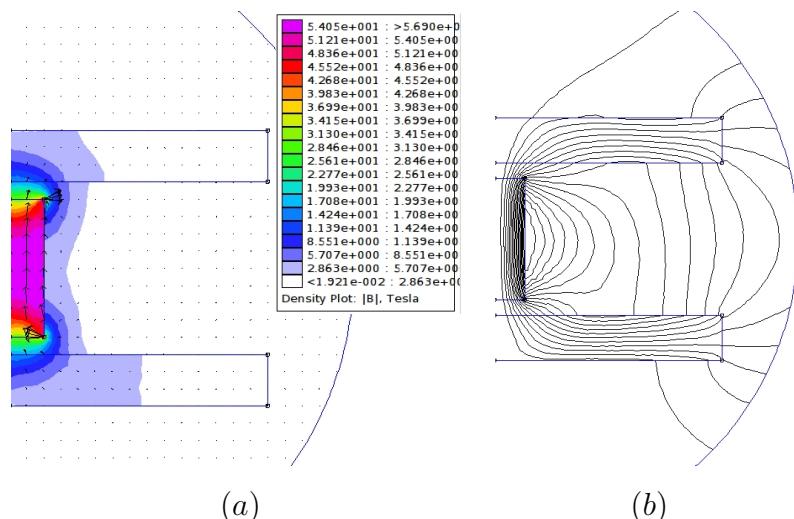
Otrajā testā tika veikti divi aprēķini kvadratūras magnetrona modelim (4.27. att.), tas ir magnetrons feromagnētisku materiālu uzputināšanai, sastāv no feromagnētiskas plāksnes (apakšā) un dažādos virzienos vērstiem magnētiem (magnētu magnetizācijas virzieni ar bultiņām parādīti FEMM piemērā 4.28. attēlā) [10].

Pirmajā aprēķinā tika uzdoti magnēti ar tik lielu magnetizāciju, lai magnētiskā lauka līnijas “izietu” no feromagnētiem. `multiRegionPlasmaTest` rezultāts (4.29. att.) un atbilstošs FEMM rezultāts (4.30. att.). Novērojamas nelielas \mathbf{B} atšķirības apakšējā magnētiskajā vadā.

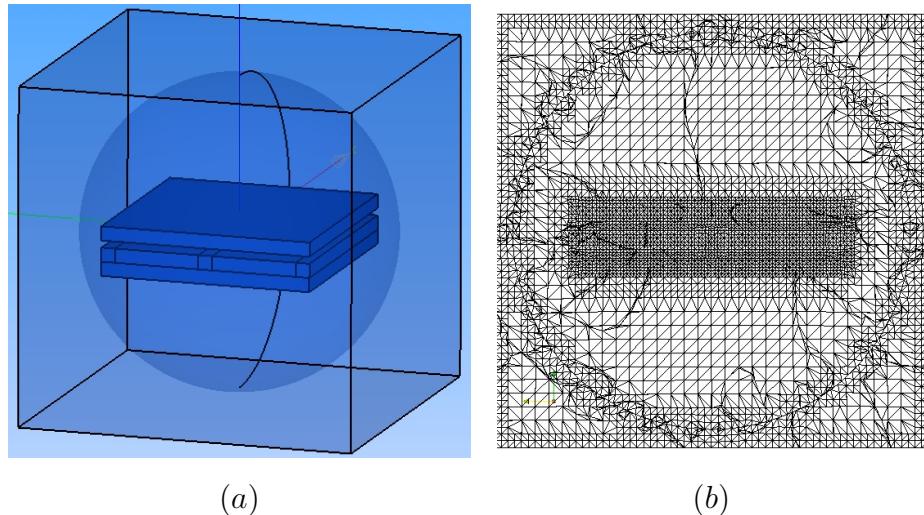
Otrajā aprēķinā tika uzdoti magnēti ar 100 reizes mazāku magnetizāciju, tāpēc magnētiskā lauka līnijas “neiziet” no feromagnētiem. `multiRegionPlasmaTest` rezultāts (4.31. att.), atbilstošs FEMM (4.32. att.) un COMSOL (4.33. att.) rezultāts. Jāatzīmē, ka FEMM aprēķins ir 2D, turpretim COMSOL aprēķins ir 3D. Kvalitatīvi FEMM un COMSOL rezultāti ir līdzīgi - magnētiskā plūsma iet caur augšējo mērķi un magnētiem, turpretim `multiRegionPlasmaTest` aprēķinā liela tās daļa iet caur apakšējo magnētisko vadu. Visticamāk tas ir saistīts ar robežnosacījumiem.



Att. 4.25: `multiRegionPlasmaTest` aprēķins cilindriskam magnētam un diviem feromagnētiskiem diskiem ar μ atkarību no \mathbf{B} : (a) \mathbf{B} amplitūda un nenormēti vektori, (b) \mathbf{A} amplitūdas sadalījums, (c) μ sadalījums.



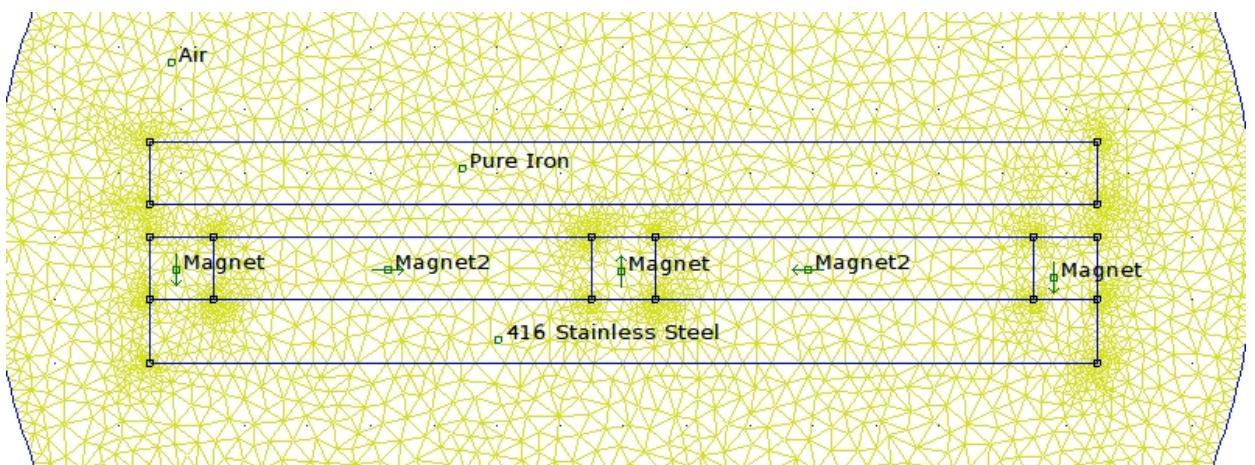
Att. 4.26: FEMM aprēķins cilindriskam magnētam un diviem feromagnētiskiem diskiem ar μ atkarību no \mathbf{B} : (a) \mathbf{B} amplitūda un normēti vektori, (b) magnētiskās plūsmas līnijas.



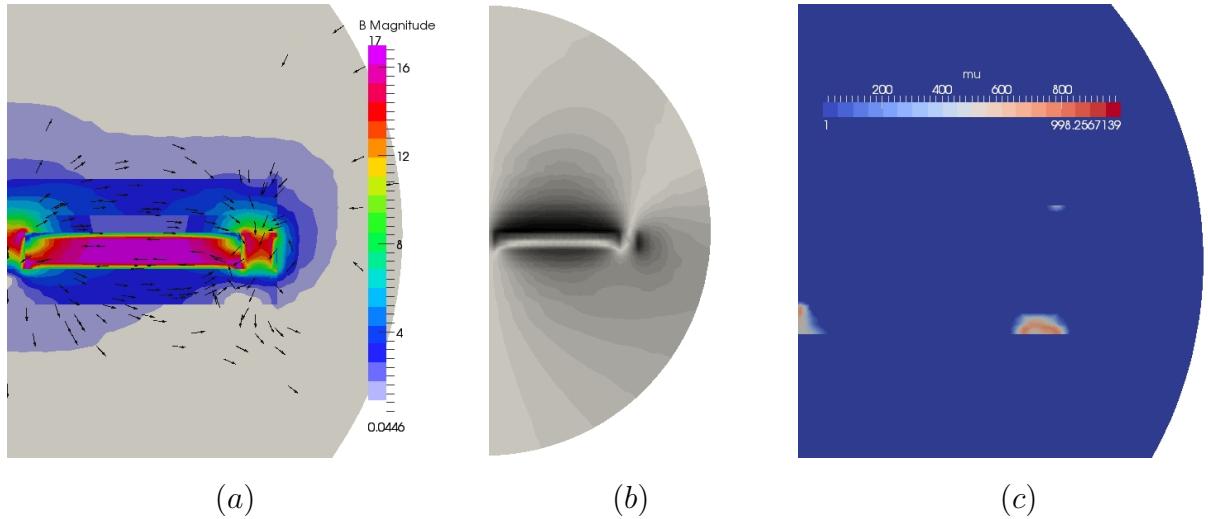
(a)

(b)

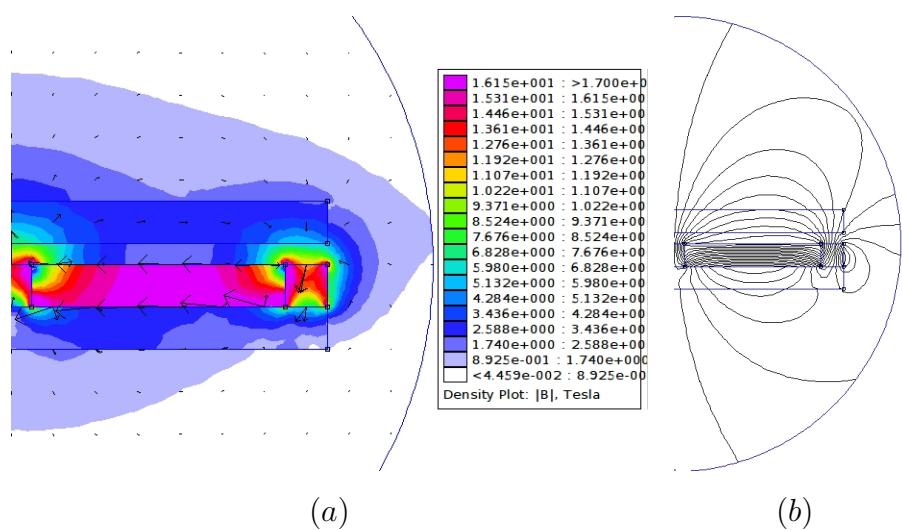
Att. 4.27: (a) Kvadratūras magnetrons (5 magnēti, magnētiskais vads apakšā, feromagnētisks mērķis augšā), (b) režģis (986287 režģa šūnas).



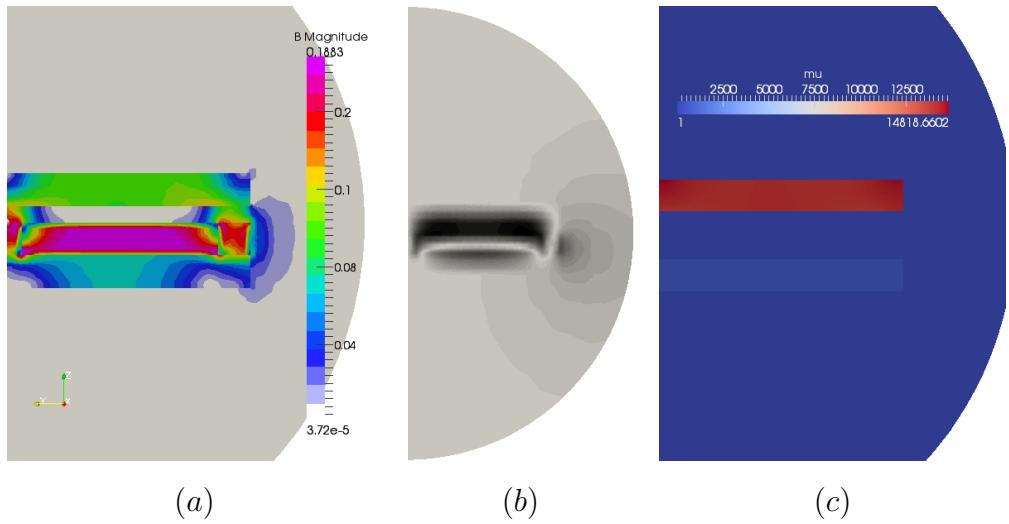
Att. 4.28: FEMM 2D piemērs kvadratūras magnetronam. Augšējam mērķim un apakšējam magnētiskajam vadam uzdota μ atkarība no \mathbf{B} (4198 režģa šūnas).



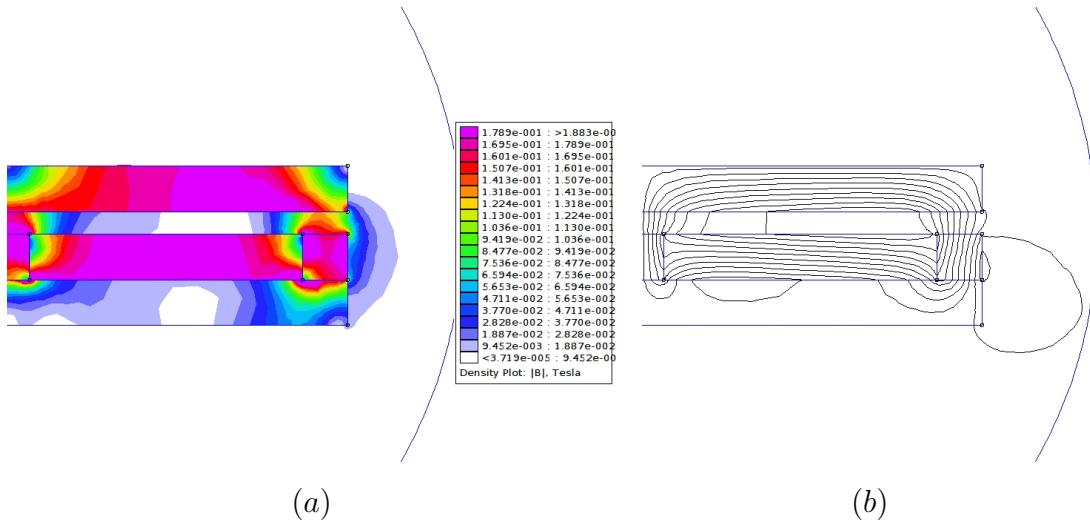
Att. 4.29: `multiRegionPlasmaTest` aprēķins kvadratūras magnetronam ar specīgiem magnētiem un μ atkarību no \mathbf{B} augšējam mērķim un apakšējam magnētiskajam vadam:
 (a) \mathbf{B} amplitūda un nenormēti vektori, (b) \mathbf{A} amplitūdas sadalījums, (c) μ sadalījums.



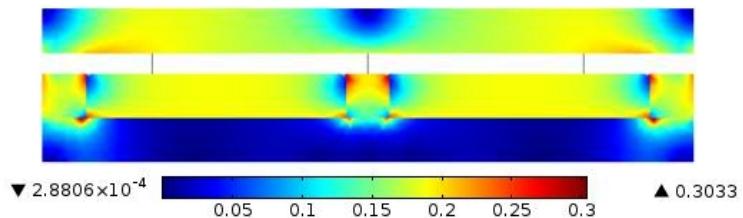
Att. 4.30: FEMM aprēķins kvadratūras magnetronam ar specīgiem magnētiem un μ atkarību no \mathbf{B} augšējam mērķim un apakšējam magnētiskajam vadam: (a) \mathbf{B} amplitūda un normēti vektori, (b) magnētiskās plūsmas līnijas.



Att. 4.31: `multiRegionPlasmaTest` aprēķins kvadratūras magnetronam ar vājiem magnētiem un μ atkarību no \mathbf{B} augšējam mērķim un apakšējam magnētiskajam vadam: (a) \mathbf{B} amplitūda un nenormēti vektori, (b) \mathbf{A} amplitūdas sadalījums, (c) μ sadalījums.



Att. 4.32: FEMM aprēķins kvadratūras magnetronam ar vājiem magnētiem un μ atkarību no \mathbf{B} augšējam mērķim un apakšējam magnētiskajam vadam: (a) \mathbf{B} amplitūda un normēti vektori, (b) magnētiskās plūsmas līnijas.



Att. 4.33: COMSOL aprēķins \mathbf{B} amplitūdai kvadratūras magnetronam ar vājiem magnētiem.

4.2.5. A aprēķins materiāliem ar nehomogēnu μ

Teorētiskais apraksts

Konstantam μ vienādojumu (4.26) var vienkāršot līdz $-\nu\Delta\mathbf{A} = \mathbf{J}_S$, jo $\nabla \times \nu\nabla \times \mathbf{A} = 0$. Taču gadījumos ar nehomogēnu μ , atmestā komponente nav vienāda ar 0 un $\nabla\nu\nabla \cdot \mathbf{A} \neq \nu\Delta\mathbf{A}$.

`multiRegionPlasmaFoam` jāuzdod šādi vienādojumi:

```
solve( fvm::laplacian(phi) );
J=-sigma*fvc::grad(phi);
solve( fvc::curl(fvc::curl(A)/mu) - fvm::laplacian(1/mu, A) ==
(J+fvc::curl(Hc))*constant::electromagnetic::mu0 );
B=fvc::curl(A);
```

Secinājumi

Vienādojumi tika ieviesti, taču iepriekš darbā aplūkotie risinājumi diverģēja. Iespējams, nepieciešams ieviest aprēķina relaksāciju. Turpmākie šo vienādojumu pētījumi netika veikti, pamatojoties uz to, ka tie izpaužas gadījumos, kad μ sadalījums feromagnētiķos ir nehomogēns, proti, magnetronu sistēmās ar feromagnētiskiem mērķiem un ļoti specīgiem magnētiskajiem laukiem, taču šo sistēmu korekta modelēšana ir atsevišķa darba vērtā.

4.3. Citi formulējumi

Formulējumu izvedumi aizgūti no *Testoni* darba [13].

4.3.1. A, V- Ψ (skalārā potenciāla) formulējums

Teorētiskais apraksts

Skalārais magnētiskais potenciāls Ψ var tikt lietots apgabalā bez virpuļstrāvām Ω_2 , virpuļstrāvu apgabalā Ω_1 aprēķiniem tiek lietots magnētiskais vektorpotenciāls. Šādu formulējumu sauc par $\mathbf{A}, V\text{-}\Psi$.

Apgabalā bez virpuļstrāvām Ω_2 magnētiskā lauka intensitāti varam dalīt divās dalās:

$$\mathbf{H} = \mathbf{H}_S + \mathbf{H}_m, \quad (4.39)$$

kur \mathbf{H}_S ir magnētiskās intensitātes vektors, kuru rada avota strāvas \mathbf{J}_S (vienādojums (4.1)), turpretim \mathbf{H}_m ir tāds vektors, kuram izpildās

$$\nabla \times \mathbf{H}_m = 0. \quad (4.40)$$

\mathbf{H}_m var būt izteikts kā reducētā magnētiskā skalārā potenciāla ϕ gradients:

$$\mathbf{H}_m = -\nabla\phi. \quad (4.41)$$

Rezultātā, vienādojumu (4.39) var pārrakstīt formā

$$\mathbf{H} = \mathbf{H}_s - \nabla\phi. \quad (4.42)$$

Lauks \mathbf{H}_s var tikt aprēķināts, izmantojot Bio-Savāra likumu:

$$\mathbf{H}_s = \frac{1}{4\pi} \oint_S \frac{\mathbf{J}_s \times \mathbf{r}}{|\mathbf{r}^3|} dS, \quad (4.43)$$

kur dS ir bezgalīgi mazs noslēgtas virsmas S apgabals. Vienādojums (4.4) bija

$$\mathbf{B} = \mu\mu_0 \cdot \mathbf{H}. \quad (4.4)$$

Ievietojot (4.43) iekš (4.4) un izsakot diverģenci, iegūstam

$$\nabla \cdot \mu\nabla\phi = \nabla \cdot \mu\mathbf{H}_s. \quad (4.44)$$

Gadījumos, kad μ ir ļoti liels, \mathbf{H}_s un \mathbf{H}_m ir ar kārtu līdzīgi un pretēji vērsti un tas ir potenciāls kļūdu cēlonis. Šo problēmu var atrisināt, lietojot feromagnētiskiem apgabaliem bez avotu strāvām (tātad $\mathbf{J}_s = 0$) skalāro magnētisko potenciālu Ψ . Līdz ar to vienādojumu (4.1) var uzrakstīt formā

$$\nabla \times \mathbf{H} = 0. \quad (4.45)$$

No tā izriet, ka

$$\mathbf{H} = -\nabla\Psi. \quad (4.46)$$

Ievietojot (4.46) iekš (4.4) un izsakot diverģenci, iegūstam

$$\nabla \cdot \mu\nabla\Psi = 0 \quad \text{apgabalam } \Omega_2. \quad (4.47)$$

Virpulstrāvu apgabalos joprojām paliek spēkā vektorpotenciāla \mathbf{A} vienādojums (4.20):

$$\nabla \times \nu\nabla \times \mathbf{A} + \sigma \left\{ \frac{\partial \mathbf{A}}{\partial t} + \nabla V \right\} = 0 \quad \text{apgabalam } \Omega_1.$$

Ievērojot robežnosacījumus (4.12) un (4.13), iegūstam \mathbf{A}, V - \mathbf{A} - Ψ formulējuma robežnosacījumus uz robežvirsmas S_{12} :

$$\mathbf{n}_1 \cdot \nabla \times \mathbf{A}_1 - \mathbf{n}_2 \cdot \mu_2(\nabla\Psi) = 0, \quad (4.48)$$

$$\nu\nabla \times \mathbf{A}_1 \times \mathbf{n}_1 - \nabla\Psi \times \mathbf{n}_2 = 0. \quad (4.49)$$

Uz aprēķinu apgabala ārējās virsmas magnētiskajam vektorpotenciālam \mathbf{A} lieto tos pašus robežnosacījumus kā \mathbf{A}, V - \mathbf{A} formulējumā. Lielumam Ψ uz ārējās robežvirsmas uzdod $\Psi = 0$.

Apgabalam Ω_1 spēkā paliek vienādojums (4.27):

$$\nabla \cdot \left\{ -\sigma \frac{\partial \mathbf{A}}{\partial t} - \sigma \nabla V \right\} = 0 \quad \text{apgabalam } \Omega_1. \quad (4.50)$$

Secinājumi

Šis formulējums, salīdzinot ar $\mathbf{A}, V\text{-}\mathbf{A}$, ļauj taupīt atmiņu un aprēķinu laiku [13]. Manuprāt, tas arī ļauj uzdot precīzus robežnosacījumus dielektriskajiem apgabaliem (gaiss utt.), kuros lieto skalāro potenciālu, jo nav nepieciešams ar dažādām robežnosacījumu aproksimācijām aizstāt inverso rotoru, bet gan var tieši lietot skalārā potenciāla gradientu $\nabla\Psi$. Šo formulējumu nedrīkst lietot vairākkārt savienotiem apgabaliem. *Testoni* darbā [13] aplūko šī formulējuma modifikāciju - $\mathbf{A}, V\text{-}\mathbf{A}\text{-}\Psi$, kas paredz vairākkārt savienota apgabala Ω_2 dalīšanu, lai iegūtu vienu vai vairākas nesavienotas apgabala daļas, kurās lieto Ψ , bet pārējās apgabala daļas - \mathbf{A} . Šis formulējums netika realizēts, jo tas sarežģī geometriju.

4.3.2. $\mathbf{T}\text{-}\Omega, \Omega$ formulējums

Teorētiskais apraksts

Līdzīgi kā $\mathbf{A}, V\text{-}\mathbf{A}\text{-}\Psi$, arī šajā formulējumā bez-virpuļstrāvu apgabalos tiek reķināts skalārs magnētiskais potenciāls Ω , kas samazina aprēķinu laiku un atmiņas patēriņu.

Formulējuma izvedumu sāksim ar vienādojumu (4.8):

$$\nabla \cdot \mathbf{J} = 0.$$

Ieviesīsim brīvi izvēlētu vektoru \mathbf{T} , kuru turpmāk sauksim par elektrisko vektoriālo potenciālu:

$$\mathbf{J} = \nabla \times \mathbf{T}. \quad (4.51)$$

Atzīmēsim, ka $\nabla \cdot \nabla \times \mathbf{T} = 0$, kur \mathbf{T} ir pietiekami daudz reižu diferencējama vektoriāla funkcija. No vienādojuma (4.1) $\nabla \times \mathbf{H} = \mathbf{J}$ izriet, ka \mathbf{T} un \mathbf{H} atšķiras par skalāras funkcijas gradientu

$$\mathbf{H} = \mathbf{T} - \nabla\Omega \quad \text{apgabalam } \Omega_1, \quad (4.52)$$

kur Ω ir magnētiskais skalārais potenciāls. Apvienojot vienādojumus (4.2), (4.4) un (4.5), iegūstam

$$\nabla \times \frac{1}{\sigma} \nabla \times \mathbf{T} + \mu \frac{\partial}{\partial t} (\mathbf{T} - \nabla\Omega) = 0 \quad \text{apgabalam } \Omega_1 \quad (4.53)$$

un no vienādojuma (4.3) $\nabla \cdot \mathbf{B} = 0$ izriet, ka

$$\nabla \cdot \mu (\mathbf{T} - \nabla\Omega) = 0 \quad \text{apgabalam } \Omega_1. \quad (4.54)$$

Bez-virpuļstrāvu apgabalos magnētisko lauku reķina caur skalāro potenciālu Ω

$$\mathbf{H} = -\nabla\Omega \quad \text{apgabalam } \Omega_2, \quad (4.55)$$

kur Ω atrod no vienādojuma

$$-\nabla \cdot \mu \nabla \Omega = 0 \quad \text{apgabalam } \Omega_2. \quad (4.56)$$

Lai pilnībā definētu vektoriālo lauku \mathbf{T} , nepieciešams definēt tā rotoru, diverģenci un atbilstošus robežnosacījumus. Izmantosim Kulona nosacījumu un pieņemsim, ka

$$\nabla \cdot \mathbf{T} = 0. \quad (4.57)$$

Tādā gadījumā vienādojumam (4.53) trūkst loceklis $\nabla_{\sigma}^{\frac{1}{\sigma}} \nabla \cdot \mathbf{T}$. Pilnais vienādojums izskaitās tā:

$$\nabla \times \frac{1}{\sigma} \nabla \times \mathbf{T} - \nabla \frac{1}{\sigma} \nabla \cdot \mathbf{T} + \mu \frac{\partial}{\partial t} (\mathbf{T} - \nabla \Omega) = 0 \quad \text{apgabalam } \Omega_1. \quad (4.58)$$

No vienādojuma (4.14) $\mathbf{J}_1 \cdot \mathbf{n}_1 = 0$ varam izteikt \mathbf{T} robežnosacījumus:

$$\mathbf{n}_1 \times \mathbf{T} = 0. \quad (4.59)$$

Ievietojot vienādojumus (4.4) $\mathbf{B} = \mu \cdot \mathbf{H}$, (4.3) $\mathbf{H} = \mathbf{T} - \nabla \Omega$ un (4.55) $\mathbf{H} = -\nabla \Omega$ iekš vienādojumiem (4.12) un (4.13), iegūstam robežnosacījumus uz robežvirsmas S_{12} :

$$\mu_1 \cdot (\mathbf{T}_1 - \nabla \Omega_1) \cdot \mathbf{n}_1 + \mu_2 \cdot (-\nabla \Omega_1) \cdot \mathbf{n}_2 = 0, \quad (4.60)$$

$$(\mathbf{T}_1 - \nabla \Omega_1) \times \mathbf{n}_1 + (-\nabla \Omega_1) \times \mathbf{n}_2 = 0. \quad (4.61)$$

Uz aprēķinu apgabala ārējās virsmas izpildās $\mathbf{B} \cdot \mathbf{n} = 0$ un $\mathbf{H} \times \mathbf{n} = 0$. Izmantojot vienādojumus (4.52) un (4.55), iegūstam ārējos robežnosacījumus:

$$\mu \cdot (-\nabla \Omega) \cdot \mathbf{n} = 0, \quad \text{ja } \sigma = 0; \quad (4.62)$$

$$(-\nabla \Omega) \times \mathbf{n} = 0, \quad \text{ja } \sigma = 0; \quad (4.63)$$

$$\mu \cdot (\mathbf{T} - \nabla \Omega) \cdot \mathbf{n} = 0, \quad \text{ja } \sigma \neq 0; \quad (4.64)$$

$$(\mathbf{T} - \nabla \Omega) \times \mathbf{n} = 0, \quad \text{ja } \sigma \neq 0. \quad (4.65)$$

Secinājumi

Tika mēģināts ieviest šo formulējumu OpenFOAM, taču rezultāts diverģē. Vienādoju mu diskretizācijas piemērs apgabalam Ω_1 :

```

1:   solve( fvm::laplacian(phi) );
2:   J=-sigma*fvc::grad(phi);
3:   solve( fvm::laplacian(T)==fvc::curl(J-fvc::curl(Hc)*mu) );
4:   solve( fvm::laplacian(Omega)==fvc::div(T) );
5:   B=(T-fvc::grad(Omega))*constant::electromagnetic::mu0*mu;

```

Iespējamie iemesli, kāpēc neizdevās ieviest šo formulējumu OpenFOAM:

- Kļūda vienādojumos un robežnosacījumos. Iespējams, jāizmanto citas vienādojumu diskretizācijas shēmas.
- 3. un 4. vienādojumu jārēķina kā vienādojumu sistēmu. *Testoni* [13] lieto FEM jeb galīgo elementu metodi, diskretizē vienādojumus lineārā vienādojumu sistēmā, tad to risina ar programmu ANSYS.
- Apgabals Ω_2 ir vairākkārt savienots. Līdzīgi kā $\mathbf{A}, V\text{-}\mathbf{A}\text{-}\Psi$, arī šajā formulējumā bez-virpulšstrāvu apgabalos tiek rēķināts skalārs magnētiskais potenciāls, kuru apzīmē ar Ω . Iespējams, lietojot šo formulējumu vairākkārt savienotiem apgabaliem, nepieciešams tos dalīt daļās.
- Šo formulējumu nevar izmantot galīgo tilpumu metodei (FVM). Daudzas elektromagnētisma problēmu risināšanas programmas lieto galīgo elementu metodi (FEM), piemēram, FEMM, ANSYS un COMSOL.

5. Retinātas plazmas modelēšana

5.1. apakšnodaļā tiek aplūkoti divi formulejumi neitrālas gāzes dinamikas modelešanai - CFD (Computational Fluid Dynamics - skaitliskā hidrodinamika) un DSMC (Direct Simulation Monte Carlo - tiešās simulācijas Montekarlo (metode)). 5.2. apakšnodaļā aplūkota lādētu daļiņu savstarpējās un ārējā elektromagnētiskā lauka mijiedarbības modelešana ar PIC (Particle In Cell - "daļiņas šūnā") metodi, kā arī tās vienkāršota realizācija programmā `multiRegionPlasmaFoam`.

5.1. Elektriski neitrālas gāzes modelešana

Gāzu dinamikas modelešanai pamatā tiek izmantoti divi formulejumi: CFD un DSMC metode [15].

Metodes izvēle nav patvalīga, izvēli nosaka gāzes retinājuma pakāpe (kritēriji tiks precizēti 5.1.1. un 5.1.2. nodaļā), ko novērtē ar *Knudsena skaitli* (Kn):

$$(Kn) = \lambda/L,$$

kur λ - vidējais brīvā ceļa garums, L - plūsmas raksturīgais izmērs. λ norāda uz vidējo ceļa garumu, ko veic molekula starp sadursmēm:

$$\lambda = \frac{1}{\sqrt{2\pi} d^2 n},$$

kur d - molekulas efektīvais diametrs, n - daļiņu koncentrācija. Plūsmas raksturīgo izmēru L definē kā makroskopisko lielumu (piem., blīvums ρ) amplitūdas gradientu:

$$L = \frac{\rho}{(d\rho/dx)}. \quad (5.1)$$

5.1.1. Skaitliskās hidrodinamikas (CFD) metode

CFD (Computational Fluid Dynamics) izmanto Navjē-Stoksa vienādojumu:

$$\rho \left(\frac{\partial \mathbf{v}}{\partial t} + \mathbf{v} \cdot \nabla \mathbf{v} \right) = -\nabla p + \nabla \cdot \mathbf{T} + \mathbf{f},$$

kur ρ - gāzes blīvums, \mathbf{v} - plūsmas ātruma vektors, p - spiediens, \mathbf{T} - sprieguma deviators, \mathbf{f} - ārējo spēku blīvums [24].

Apvienojot šo metodi ar Maksvela vienādojumiem, iegūst magnētiskās hidrodinamikas (MHD) formulejumu. `OpenFOAM` ir ieklauta standarta programma MHD problēmu risināšanai `mhdFoam`.

CFD lieto gāzēm, kurām Kn ir mazāks par 0.1. Pie Kn lielākiem par 0.1, uzsaka, ka gāzei nepastāv kvazistacionārs stāvoklis, sekojoši, nevar definēt tādus statistiskus, makroskopiskus lielumus kā spiediens, blīvums, temperatūra.

Argona gāzes un elektronu koncentrācija uzputināšanas procesā ir robežās no 10^{12} m^{-3} līdz 10^{18} m^{-3} (atmosfēras spiedienā 10^{25} m^{-3}), argona molekulas efektīvā diametra novērtējums $4.2 \cdot 10^{-10} \text{ m}$. Attiecīgi λ ir robežās no 1.3 m līdz $1.3 \cdot 10^6 \text{ m}$. Nemot L kā aprēķinu apgabala izmēru 1 m , izriet, ka CFD metodi izmantot nevar.

5.1.2. Tiešās simulācijas Montekarlo (DSMC) metode

DSMC (Direct Simulation Monte Carlo) metode aplūko molekulas kā diskrētas daļīnas vai daļīnu kopas - superdaļīnas. Katrai daļīnai ir piešķirtas fizikālas īpašības (masa, diametrs utt.), koordinātas un ātrums. DSMC balstās uz Boltzmana vienādojumu, kurš vienatomu gāzei ir

$$\frac{\partial f}{\partial t} + \mathbf{c} \cdot \frac{\partial f}{\partial \mathbf{r}} + \mathbf{F} \cdot \frac{\partial f}{\partial \mathbf{c}} = \left(\frac{\partial f}{\partial t} \right)_C,$$

kur $f(\mathbf{c}, \mathbf{r}, t)$ - sadalījuma funkcija, \mathbf{c} - molekulu ātrums, \mathbf{r} - molekulu koordinātas, \mathbf{F} - ārējs spēks. Vienādojuma pirmais saskaitāmais norāda uz sadalījuma funkcijas izmaiņu laikā, otrs saskaitāmais norāda uz f izmaiņu sakarā ar molekulu pārvietošanos caur telpas apgabala $d\mathbf{r}$ robežām, trešais saskaitāmais saistīts ar molekulu ātruma izmaiņām ārēja spēka \mathbf{F} rezultātā, vienādojuma labajā pusē esošais loceklis norāda uz f izmaiņu molekulu savstarpējo sadursmju rezultātā [6].

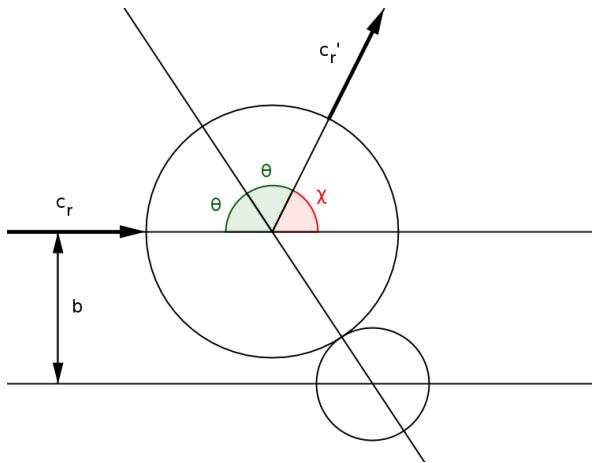
DSMC nodrošina daļīnu sadursmes, kuras balstās uz nejauši ģenerētiem skaitļiem un no klasiskās mehānikas atvasinātiem daļīnu sadursmju modeļiem. Pirms aplūkot DSMC darbību, vispirms iepazīsimies ar vienkāršu daļīnu sadursmju modeli, kas izriet no klasiskās mehānikas.

Daļīnu sadursmes no klasiskās mehānikas viedokļa

No klasiskās mehānikas viedokļa daļīnas saduras, kad

$$d_{12} = \frac{1}{2}(d_1 + d_2), \quad (5.2)$$

kur d_{12} ir attālums starp daļīnu centriem, d_1 un d_2 ir daļīnu diametri. No impulta un enerģijas saglabāšanās likuma, zinot daļīnu masas, attālumu b (skat. 5.1. att.), kā arī relatīvo ātrumu pirms sadursmes $\mathbf{c}_r = \mathbf{c}_1 - \mathbf{c}_2$, kur \mathbf{c}_1 un \mathbf{c}_2 ir daļīnu ātrumi pirms sadursmes, iespējams noteikt relatīvo ātrumu pēc sadursmes $\mathbf{c}'_r = \mathbf{c}'_1 - \mathbf{c}'_2$, kur \mathbf{c}'_1 un \mathbf{c}'_2 ir daļīnu ātrumi pēc sadursmes [6].



Att. 5.1: Divu daļiņu sadursmes shematisks attēlojums. \mathbf{c}_r un \mathbf{c}'_r - relatīvais ātrums pirms un pēc sadursmes.

Vienkāršākais daļiņu sadursmes modelis ir cieto sfēru modelis (HS), kuram daļiņu ātrumus pēc sadursmes iegūst ar formulām:

$$\mathbf{c}'_1 = \mathbf{c}_m + \frac{m_1}{m_1 + m_2} \mathbf{c}'_r, \quad (5.3)$$

$$\mathbf{c}'_2 = \mathbf{c}_m + \frac{m_2}{m_1 + m_2} \mathbf{c}'_r, \quad (5.4)$$

kur m_1 un m_2 ir daļiņu masas, \mathbf{c}_m ir abu daļiņu masas centra ātrums [5]. Jāatzīmē, ka $|\mathbf{c}_r| = |\mathbf{c}'_r|$, līdz ar to relatīvais ātrums sadursmes brīdī pagriežas par leņķi χ (skat. 5.1. attēlu) [6]. χ ar attālumu b saista vienādojums

$$b = d_{12} \cos\left(\frac{\chi}{2}\right).$$

Taču šim cieto sfēru modelim un klasiskās mehānikas piegājenam daļiņu sadursmju noteikšanā ir vairāki trūkumi:

- Molekulas atgrūžas *van der Waals* spēka rezultātā, līdz ar to tās nevar uzskatīt par cietiem ķermeniem. Molekulu efektīvais sadursmes šķērsgriezums σ ir apgriezti proporcionāls \mathbf{c}_r jeb $\sigma \sim 1/\mathbf{c}_r$.
- Vairākatomu molekulām ir papildus brīvības pakāpes un ne-sfēriska forma, tāpēc no mehānikas un ģeometrijas atvasinātie sfērisko daļiņu sadursmju modeli nav korekti.
- Lai tiešā veidā konstatētu divu daļiņu sadursmi, nepieciešams rēķināt attālumus starp daļiņām. Šāda “attāluma rēķināšanas” piegājiena trūkums - nepieciešams ļoti mazs laika solis, lai konstatētu visas sadursmes, jo pie liela laika soļa daļiņas izlidos viena otrai cauri, un sadursme netiks konstatēta.
- Cieto sfēru modelim $\mu \sim T^{1/2}$, kur μ ir gāzes viskozitāte un T - temperatūra. Taču reālām gāzem spēkā sakarība $\mu \sim T^{3/4}$, tāpēc cieto sfēru modeli nav vēlams izmantot gāzes temperatūrām ar kārtu 1000 K un vairāk [6].

Daļīņu sadursmes no DSMC viedokļa

DSMC apiet visus iepriekš minētos klasiskās mehānikas trūkumus un daļīņu sadursmes aplūko statistiski. Turpmāk aplūkotais modeļa apraksts aizgūts no [5].

Aplūkosim vienu režģa šūnu ar superdaļīņu skaitu N , tilpumu V_C . Maksimālā varbūtība, ka šajā šūnā laikā Δt notiks superdaļīņu sadursme, ir

$$P_{max} = F_N (\sigma_T c_r)_{max} \Delta t / V_C, \quad (5.5)$$

kur F_N raksturo vienas superdaļīņas patieso daļīņu skaitu, σ_T ir pilnais daļīņu sadursmes šķērsgriezuma laukums, r - relatīvais ātrums un $(\sigma_T c_r)_{max}$ ir maksimālā vērtība, kas atbilst kādai no režģa šūnas daļīņām. Tad pāru skaits, kuru nepieciešams pārbaudīt uz sadursmes varbūtību ir

$$\frac{1}{2} N \bar{N} F_N (\sigma_T c_r)_{max} \Delta t / V_C, \quad (5.6)$$

kur \bar{N} ir vidējais superdaļīņu skaits šūnā laikā. Taču iegūtais skaitlis nebūt nav superdaļīņu pāru sadursmju skaits. Vispirms tiek brīvi izvēlēts kāds superdaļīņu pāris, tad šī pāra sadursmes varbūtība ir

$$P = \frac{\sigma_T c_r}{(\sigma_T c_r)_{max}}. \quad (5.7)$$

Šo sadursmes varbūtību salīdzina ar nejausi ģenerētu skaitli (intervālā no 0 līdz 1) R . Ja $P > R$, tad notiek superdaļīņu pāra sadursme.

Aplūkotais superdaļīņu sadursmes detektēšanas algoritms ir *No Time Counter* (NTC), tas ir realizēts OpenFOAM bibliotēkas `dsmc` failā `DsmcCloud.C` funkcijā `collisions()`. Citus algoritmus sadursmju detektēšanai, piemēram, *Randomly Sampled Frequency* (RSF) skatīt [6].

Pēc superdaļīņu sadursmes pāra noskaidrošanas nepieciešams modelēt pašu superdaļīņu sadursmi. Atzīmēsim, ka no iepriekš aplūkotā NTC sadursmju noteikšanas algoritma nebūt neizriet, ka attālums starp superdaļīņām ir mazāks vai vienāds ar to rādiusu summu, superdaļīņas faktiski var atrasties dažādās šūnas pusēs. Iepriekš tika piesaukti vairāki *cieto sfēru* modeļa (HS) trūkumi attiecībā uz gāzes viskozitāti un sadursmes šķērsgriezumu, tāpec *G. A. Bird* piedāvā *mainīgu cieto sfēru* modeli (VHS). Par šī modeļa pamatu kalpo pieņēmums, ka sfēras diametrs d ir funkcija no relatīvā ātruma c_r un parasti šo funkcionālo sakarību pieraksta formā

$$d = d_{ref} (c_{r,ref}/c_r)^\nu, \quad (5.8)$$

kur d_{ref} un $c_{r,ref}$ ir attiecīgi atskaites diametrs un relatīvais ātrums, ν ir relatīvā ātruma eksponente (sīkāk skat. [5], [6]). Tad lenķis, par kādu pēc sadursmes mainīsies relatīvā ātruma vektors \mathbf{c}_r (5.1. att.) ir

$$\chi = 2 \cos^{-1} (b/d_{12}). \quad (5.9)$$

Taču šo formulu var lietot tikai gadījumā, ja daļīnas saduras no klasiskās mehānikas vie-dokļa, proti, izpildās nosacījums par attālumu starp daļīnām (vienādojums (5.2)), taču pie izvēlētā sadursmju noteikšanas algoritma tas neizpildās, tāpēc tiek pieņemts, ka χ vien-līdz varbūtīgi var būt vērstīs visos virzienos, tāpēc relatīvā ātruma vektoru pēc sadursmes \mathbf{c}'_r izsakot kā

$$\mathbf{c}'_r = <\cos \theta, \sin \theta \cos \phi, \sin \theta \sin \phi> |\mathbf{c}_r|, \quad (5.10)$$

kur $\cos \theta = 2R_1 - 1$ un $\phi = 2\pi R_2$, kur savukārt R_1 un R_2 ir nejauši ģenerēti skaitli no 0 līdz 1 [25].

Mainīgo sfēru modelis der sfēriskām daļīnām, piemēram, vienatomu molekulām un elektroniem. Divu un vairāk atomu molekulām papildus translācijas brīvības pakāpēm ir arī rotācijas un svārstību kustības brīvības pakāpes. Gadījumos, kad retināta gāze sastāv arī no daudzatomu molekulām, kopā ar DSMC bieži lieto *Larsen-Borgnakke* metodi, kura papildus brīvības pakāpju enerģiju summu aplūko kā molekulu iekšējo energiju, tāpēc mehāniskas rotācijas un svārstības, netiek modelētas. Šis modelis ir ieviests OpenFOAM, par tā teorētiskajiem aspektiem skat. [5] un [6].

5.2. Lādētu daļīnu un elektromagnētiskā lauka mijiedarbības modelešana

Elektriski lādētu daļīnu mijiedarbību vistiešākā veidā raksturo Kulona spēks

$$\mathbf{F}_{12} = \frac{1}{4\pi\epsilon_0} \frac{q_1 q_2}{|\mathbf{r}_{12}|^3} \mathbf{r}_{12},$$

kur \mathbf{F}_{12} ir Kulona spēks, ϵ_0 - elektriskā konstante, q_1 un q_2 - daļīnu lādiņi un \mathbf{r}_{12} - attāluma vektors starp daļīnām. Līdz ar to vienkāršākais lādētu daļīnu dinamikas modelešanas paņēmiens ir sekojošs:

1. Aprēķināt visu uz daļīnu darbojošos spēku summu, no Nūtona 2. likuma izteikt paātrinājumu.
2. Ar kādu no diferenciālvienādojumu integrēšanas paņēmiem (piem., Runge-Kuttas 4. kārtas metodi) atrast daļīnas ātrumu un koordinātes.

Aprēķinu skaits un laiks atkarībā no daļīnu skaita n mainās kvadrātiski jeb $O(n^2)$. Pie 10^4 laika soļiem un 10^4 daļīnām uz viena procesora aprēķins ilgst vairāk kā stundu, taču ticamu plazmas aprēķinu veikšanai nepieciešams vismaz par 3 kārtām vairāk daļīnu un laika soļu, līdz ar to šī tiešā lādētu daļīnu dinamikas modelešana var kalpot tikai vienkāršu rezultātu pārbaudei un sarežģītos modeļos parasti izmanto citas metodes, piemēram, Particle-In-Cell (PIC) metodi, kurai aprēķinu skaits un laiks ir lineāri atkarīgs no daļīnu skaita jeb $O(n)$.

5.2.1. Particle-In-Cell (PIC) metode

PIC metodē lādētas daļīnas mijiedarbojas ar elektrisko lauku, kurš definēts uz režģa virsotnēm. Turpmākais *PIC* metodes izklāsts aizgūts no [25].

PIC metodes cikls sastāv no šādiem soliem:

1. Lādiņa blīvuma sadalījuma aprēķins uz režģa virsotnēm.
2. Elektriskā potenciāla ϕ un elektriskā lauka intensitātes \mathbf{E} aprēķins.
3. Elektriskā lauka intensitātes \mathbf{E} interpolācija.
4. Daļīnu ātruma un koordinātas aprēķins.
5. Jaunu daļīnu pievienošana.
6. Datu saglabāšana.

1. solis: Lādiņa blīvuma sadalījuma aprēķins uz režģa virsotnēm

Aplūkosim vienkāršu gadījumu - dots regulārs 2D režģis un daļiņa režģa šūnā (5.2.att.). Ja lādiņa sadalījums starp 4 režģa punktiem A , B , C un D ir vienāds ar laukumu S_A , S_B , S_C un S_D attiecību, tad to sauc par 1. kārtas jeb lineāro sadalījumu. Ja režģa šūnas malas garums ir 1 vienību garš, tad matemātiski lādiņa sadalījuma svarus w_A , w_B , w_C un w_D izsaka kā

$$w_A = (1 - h_x)(h_y),$$

$$w_B = (h_x)(h_y),$$

$$w_C = (1 - h_x)(1 - h_y),$$

$$w_D = (h_x)(1 - h_y),$$

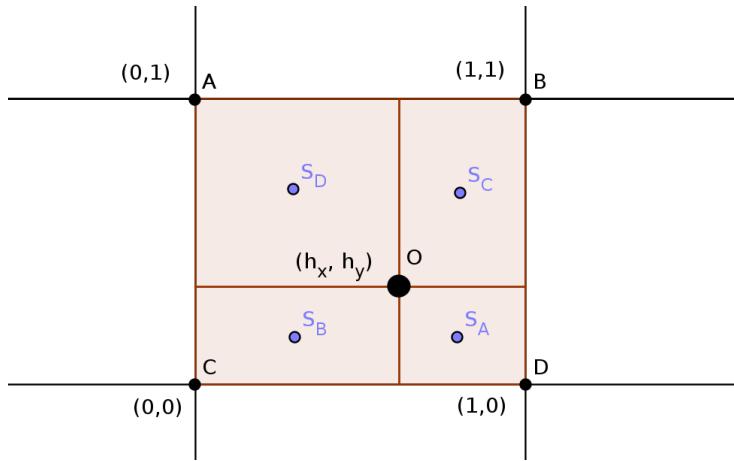
kur h_x un h_y ir daļīnas koordinātas. Blogā [25] aplūkoti gadījumi, kad režģis nav regulārs. Tad režģa šūnas apgabalam un daļīnas koordinātai veic transformāciju uz režģa šūnas iekšējo atskaites sistēmu, kurā režģa šūna ir regulāra un normēta.

2. solis: Elektriskā potenciāla un elektriskā lauka intensitātes aprēķins

Tā kā elektriskā lādiņa blīvums definēts uz režģa mezglu punktiem, tad elektriskā potenciāla ϕ un elektriskā lauka intensitātes vektora \mathbf{E} aprēķinu visā aprēķinu apgabalā veic ar FDM vai FEM metodēm.

3. solis: Elektriskā lauka intensitātes interpolācija

Šajā solī, izmantojot 1. solī aprēķinātos daļīnas lādiņa sadalījuma proporcijas, sarēķina \mathbf{E} vērtību daļīnas atrašanās vietā.



Att. 5.2: Shematisks lādiņa “O” sadalījuma attēlojums 2D gadījumā. S_A, S_B, S_C, S_D - laukumi, kas nosaka lādiņa daļas proporciju starp režģa punktiem A, B, C, D .

4. solis: Daļinu ātruma un koordinātas aprēķins

Uz lādētu daļinu elektromagnētiskajā laukā darbojas Lorenca spēks

$$\mathbf{F} = q(\mathbf{E} + \mathbf{v} \times \mathbf{B}), \quad (5.11)$$

kur q - daļinās lādinš, \mathbf{v} - daļinās ātrums. Zinot lādiņa masu m šo vienādojumu iespējams uzrakstīt galīgajās diferencēs:

$$\Delta \mathbf{v} = \frac{q}{m} (\mathbf{E} + \mathbf{v} \times \mathbf{B}) h, \quad (5.12)$$

kur $\Delta \mathbf{v}$ ir ātruma izmaiņa mazā laika solī h . Ja daļinās ātrums laikā “n” bija \mathbf{v}_n , tad laika solī “n+1” tas būs $\mathbf{v}_{n+1} = \mathbf{v}_n + \Delta \mathbf{v}$. Daļinās koordinātes izmaiņu $\Delta \mathbf{x}$ laika solī h nosaka kā:

$$\Delta \mathbf{x} = \mathbf{v}_{n+1} h, \quad (5.13)$$

un analogiski koordināte laikā “n+1” būs $\mathbf{x}_{n+1} = \mathbf{x}_n + \Delta \mathbf{x}$.

Tā kā skaitliskajos aprēķinos h nevar ņemt bezgalīgi mazu, tad vienādojumos (5.12) un (5.13) rodas skaitliskās kļūdas, kuru nevēlams blakusefekts ir daļinu pilnās enerģijas izmaiņa (visbiežāk palielināšanās). Precīzākas metodes bieži ir lēnākas, tāpēc lādētu daļinu dinamikas modelešanā parasti izmanto metodes - kompromisus starp precizitāti un ātrdarbību. Plaši lietotas ir *Runge-Kutas 4. kārtas* metode un *Borisa* metode (autors Jay P. Boris).

Ar *Runge-Kutas 4. kārtas* metodi [26] ātrumu laika solī “n+1” no Lorenca spēka vienādojuma (5.11) izsaka kā

$$\begin{aligned} \mathbf{v}_{n+1} &= \mathbf{v}_n + \frac{1}{6}(\mathbf{k}_1 + 2\mathbf{k}_2 + 2\mathbf{k}_3 + \mathbf{k}_4) h, \\ \mathbf{k}_1 &= \frac{q}{m} (\mathbf{E} + \mathbf{v}_n \times \mathbf{B}), \end{aligned}$$

$$\begin{aligned}\mathbf{k}_2 &= \frac{q}{m} \left(\mathbf{E} + \left(\mathbf{v}_n + \frac{k_1 h}{2} \right) \times \mathbf{B} \right), \\ \mathbf{k}_3 &= \frac{q}{m} \left(\mathbf{E} + \left(\mathbf{v}_n + \frac{k_2 h}{2} \right) \times \mathbf{B} \right), \\ \mathbf{k}_4 &= \frac{q}{m} \left(\mathbf{E} + \left(\mathbf{v}_n + k_3 h \right) \times \mathbf{B} \right).\end{aligned}$$

Savukārt ar *Borisa* metodi [25] ātrumu laika solī “n+1/2” izsaka kā

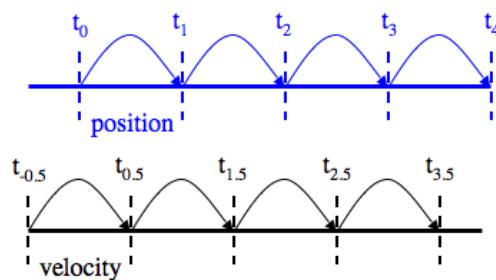
$$\begin{aligned}\mathbf{v}_{n+1/2} &= \mathbf{v}^+ + \left(\frac{q\mathbf{B}}{m} \right) \frac{h}{2}, \\ \mathbf{v}^+ &= \mathbf{v}^- + \mathbf{v}' \times \mathbf{s}, \\ \mathbf{v}' &= \mathbf{v}^- + \mathbf{v}^- \times \mathbf{t}, \\ \mathbf{v}^- &= \mathbf{v}_{n+1/2} + \left(\frac{q\mathbf{E}}{m} \right) \frac{h}{2}, \\ \mathbf{s} &= \frac{2\mathbf{t}}{1 + |\mathbf{t}|^2}, \\ \mathbf{t} &= \left(\frac{q\mathbf{B}}{m} \right) \frac{h}{2},\end{aligned}$$

kur \mathbf{s} , \mathbf{t} , \mathbf{v}' , \mathbf{v}^+ un \mathbf{v}^- ir papildus ieviesti vektori. Ātruma laika soli “n-1/2” un “n+1/2” saistīti ar secību, kādā tiek integrēts daļīgas ātrums un koordināta. To sauc par *leapfrog* jeb pārlēkšanas metodi (5.3. att.). Ātrumu laika solī “-1/2” iegūst tā:

$$\mathbf{v}_{-1/2} = \mathbf{v}_0 - \left(\frac{q\mathbf{E}}{m} \right) \frac{h}{2}.$$

Un atbilstošs koordinātas aprēķins:

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \mathbf{v}_{n+1/2} h.$$



Att. 5.3: Shematisks *leapfrog* jeb pārlēkšanas metodes attēlojums (*position* - pozīcija, *velocity* - ātrums). Bilde aizgūta no [25].

5. solis: Jaunu daļīnu pievienošana

Šajā solī tiek integrēta DSMC metode, tā gan pievieno jaunas daļīnas, gan izdzēš tās, kuras ir izlidojušas no aprēķinu apgabala. DSMC metode arī nodrošina daļīnu sadursmes un ķīmiskās reakcijas.

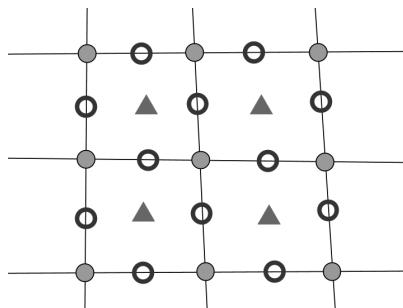
6. solis: Datu saglabāšana

Katrā laika solī tiek saglabātas daļiņu koordinātas un ātrums. Ja daļiņas mijiedarbojas ar apgabala ārējo virsmu (atsitas vai izlido no apgabala), tad pilnais leņķiskais moments un pilnā enerģija mainās, taču šo lielumu saglabāšana un analizēšana ilgākā laika periodā ļauj pārliecināties par šiem fundamentāliem saglabāšanas likumiem.

5.2.2. Vienkāršota PIC realizācija programmā `multiRegionPlasmaFoam`

`OpenFOAM` rēķina diferenciālvienādojumus ar galīgo tilpumu metodi. `OpenFOAM` ir trīs *C++* datu glabāšanas struktūras jeb klasses, kurās glabā apgabalu iekšējos laukus (skalāros un vektoriālos) (skat. 5.4. att.):

- *volField* definēts režģa šūnu centros (trijstūri).
- *surfaceField* definēts uz režģa šūnu malām (riņķi).
- *pointField* definēts režģa mezglu punktos (aplī).



Att. 5.4: `OpenFOAM` klasses aprēķinu apgabala vērtību glabāšanai.

Lai varētu izmantot lādiņa lineāro sadalījumu, kurš aprakstīts iepriekšējā nodalā (1. solis), der režģa mezglos definētie punkti (*pointField*). Sliktākajā gadījumā lādiņa lineāro sadalījumu var veikt starp režģa malām definētajiem punktiem (*surfaceField*), piemēram, sadalot pusi no katras daļiņas lādiņa starp punktiem dažādās režģa šūnas pusēs. Taču `OpenFOAM` rēķina un glabā laukus *volFields* klasē, līdz ar to daļiņu lineārais sadalījums nav realizējams un viss elektriskais potenciāls, ko rada lādiņu šūnā, pieder vienam režģa punktam šūnas centrā. Rezultātā tas novērt pie aprēķinu klūdām.

Daļiņu ātrumu un koordināšu aprēķinam tika izmantota *Runge-Kutas 4. kārtas* metode, kura aprakstīta iepriekšējā nodalā (4.solis). Šī metode uzrādīja apmierinošu precizitāti, tāpēc *Borisa* metode pagaidām netiek aplūkota. [25] norādīts, ka *Borisa* metode bieži tiek izmantota ar PIC, iespējams, tā ir ātrāka kā *Runge-Kutas* metode.

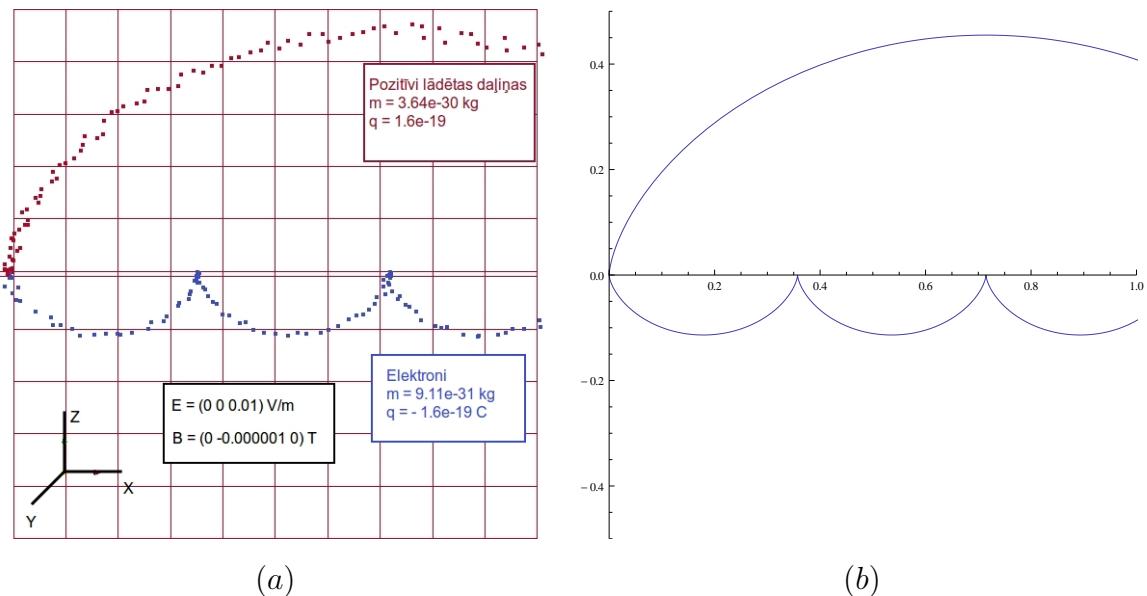
Rēķinot superdaļiņas paātrinājumu, superdaļiņas masa un lādiņš tiek nemta kā vienās daļiņas masa un lādiņš, turpretim, rēķinot elektriskā lauka intensitāti \mathbf{E} , tiek nemta

vērā visu superdalīņā “ietilpstoto” daļiņu lādiņu summa. Superdalīnu lietojums var būtiski ietaupīt aprēķinu laiku un atmiņu, reālos aprēķinos superdalīnas var aizvietot miljonus parasto daļiņu.

5.3. Rezultāti

5.3.1. Lādētu daļiņu dinamika ārējā elektromagnētiskajā laukā

Daļiņu mijiedarbības ar ārējo elektromagnētisko lauku pārbaudei tika veikts viens tests. Aprēķinu apgabalā, kurā ir homogēni, perpendikulāri vērsti elektriskais (“z” virzienā) un magnētiskais lauks (“-y” virzienā), tiek pievienoti elektroni un pozitīvas daļiņas (sākotnējais ātrums ir nulle), kuras elektriskā lauka rezultātā sāk paātrināties (5.5. (a) att.). Programmas skaitliskie aprēķini salīdzināti ar atrisinājumu (5.5. (b). att.), kas iegūts ar *Mathematica* programmu. Rezultāti sakrīt, skaitliskā atrisinājuma klūda ir vienāda ar *Runge-Kutas* metodes klūdu. Gadījumā, ja elektriskais un magnētiskais lauks nav homogēni, tad papildus klūda rodas šo lauku vērtību maiņu starp divām režģa blakus šūnām. Risinājums - smalcināt režģi.



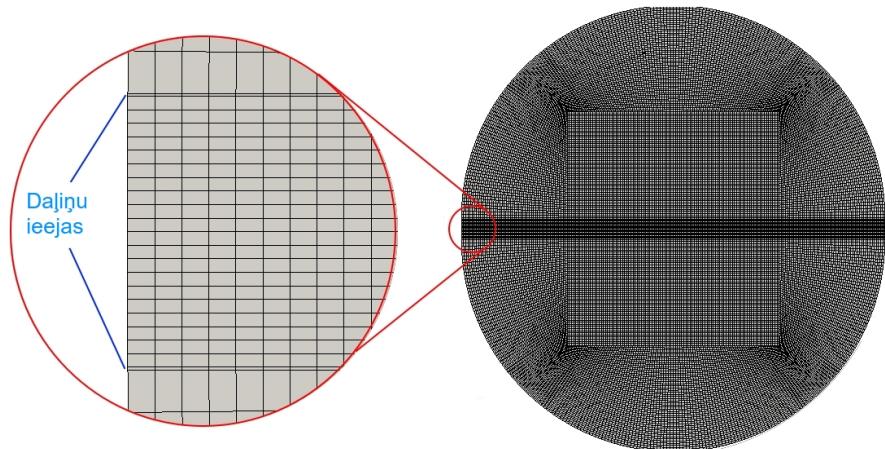
Att. 5.5: Homogēns elektriskais un magnētiskais lauks, bez daļiņu savstarpējās mijiedarbības: (a) `multiRegionPlasmaTest` lādētu daļiņu trajektorijas (2D režģis, 110 režģa šūnas), (b) *Mathematica* daļiņu dinamikas diferenciālvienādojumu risinājums.

5.3.2. Lādētu daļiņu savstarpēja mijiedarbība

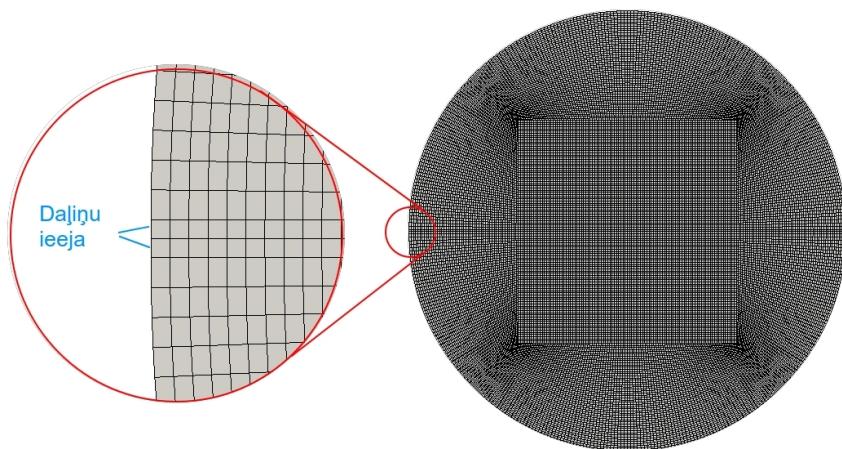
Lai pārbaudītu, kā programma modelē elektronu (turpmāk tekstā - daļiņu) savstarpējo mijiedarbību, tika veikti divi testi.

Pirmajā testā tika modelēts gadījums, kad pa divām ieejām laiž lādētas daļīnas ar ātrumu $v = 10^5 \text{ m/s}$ (režģi skat. 5.6. att.). Laika solis $h = 4 \cdot 10^{10} \text{ s}$. Tika veikti divi aprēķini, kuri modelē ekvivalentu gadījumu, taču pirmajā aprēķinā (5.8. (a) att.) apgabalā ikkatrau laika soli vidēji tiek ielaistas 2 daļīnas, bet otrajā aprēķinā (5.8. (b) att.) apgabalā ikkatrau laika soli vidēji tiek ielaistas 0.2 superdaļīnas, kuras atbilst 10 daļīnām.

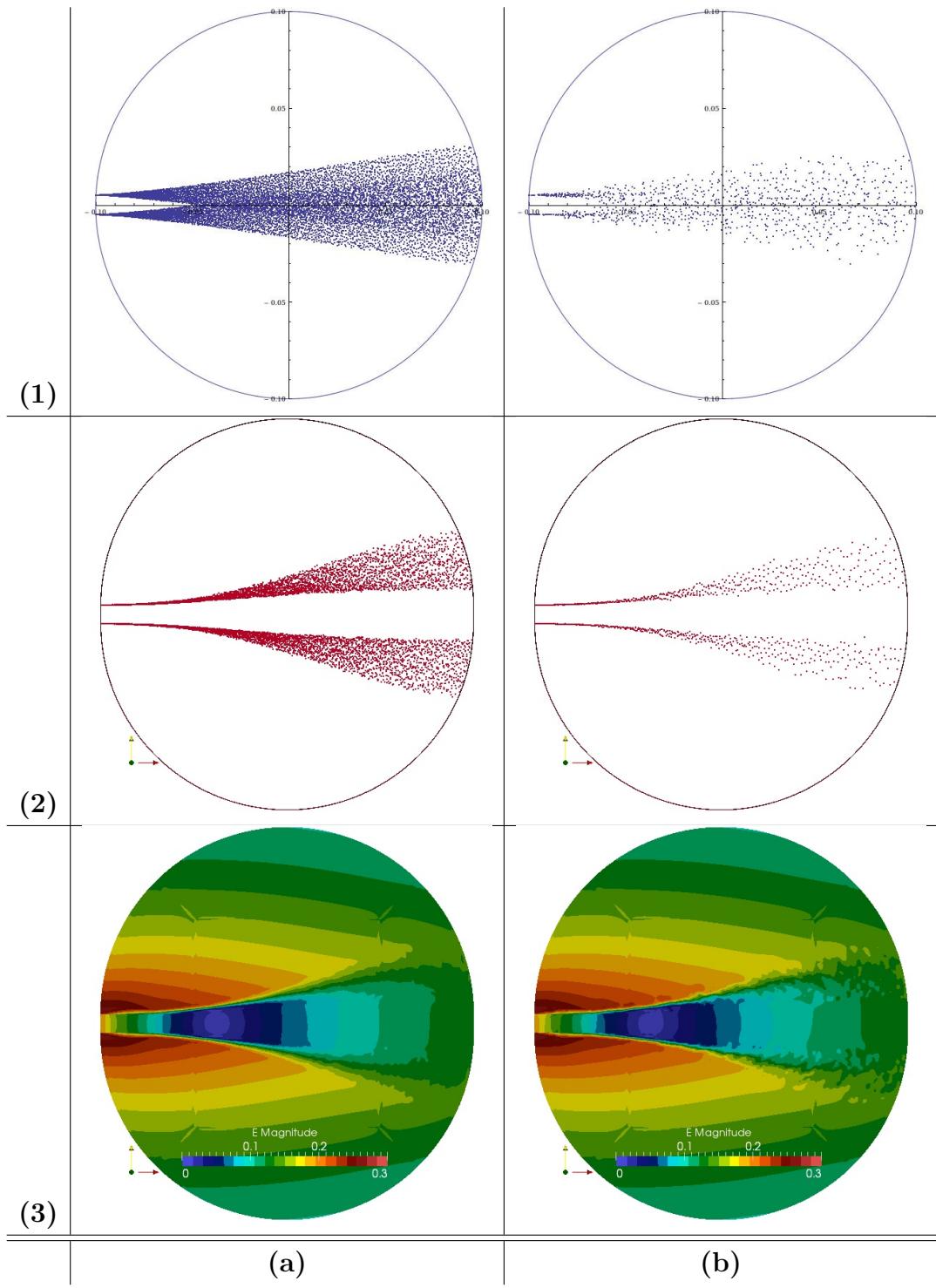
Otrajā testā tika modelēts identisks gadījums kā pirmajā testā ar vienīgo izņēmumu, ka daļīnas apgabalā tiek ielaistas pa 2 režģa šūnas platu ieeju. Pirmajā aprēķinā (5.9. (a) att.) apgabalā ikkatrau laika soli vidēji tiek ielaistas 2 daļīnas, bet otrajā aprēķinā (5.9. (b) att.) apgabalā ikkatrau laika soli vidēji tiek ielaistas 0.2 superdaļīnas, kuras atbilst 10 daļīnām.



Att. 5.6: Aprēķinam, kurā pa divām sprauslām ielido elektroni, atbilstoš 2D režģis. 34400 režģa šūnas.



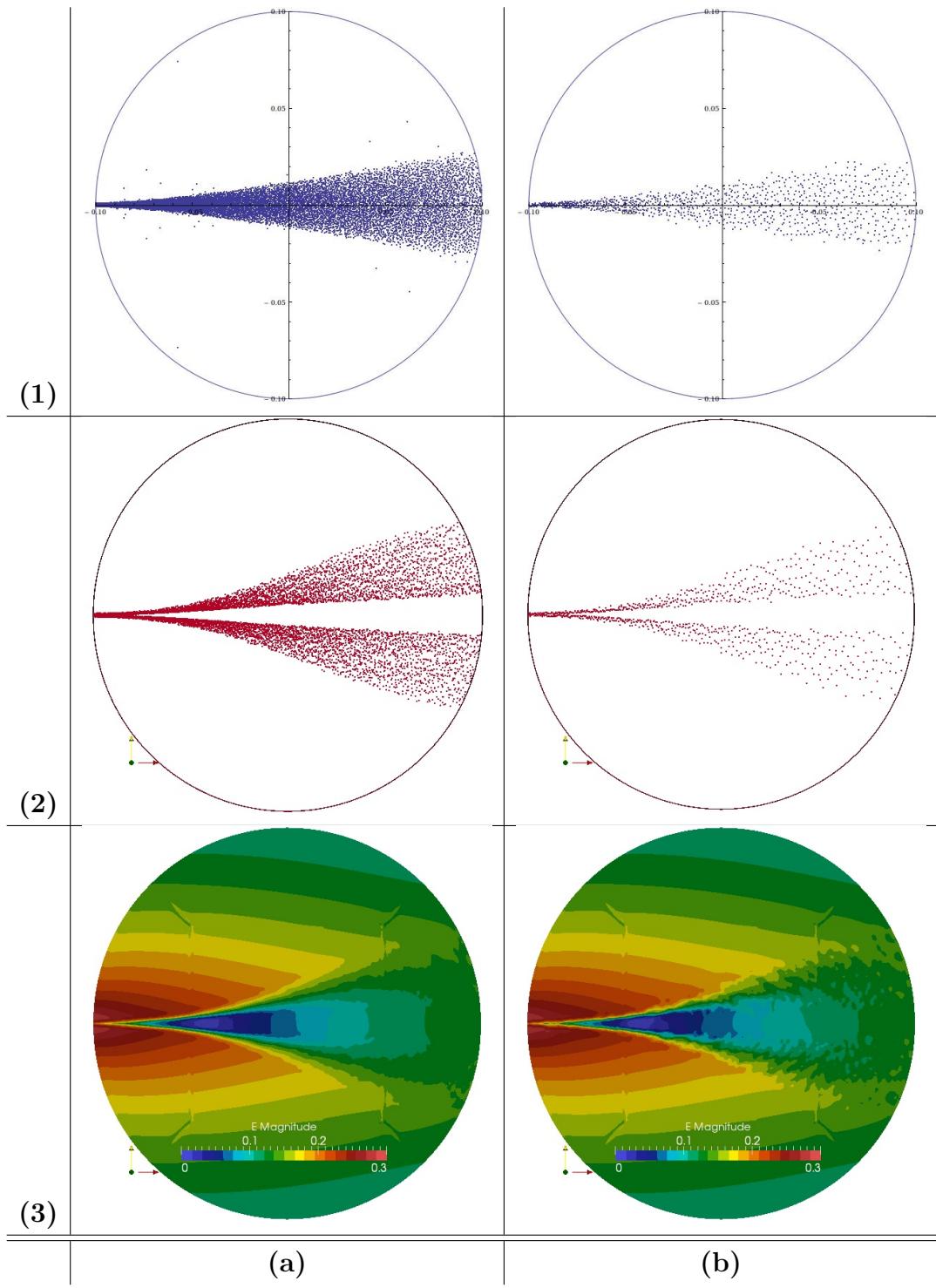
Att. 5.7: Aprēķinam, kurā pa vienu sprauslu ielido elektroni, atbilstoš 2D režģis. 30400 režģa šūnas.



Att. 5.8: Aprēķins divām sprauslām, pa kurām ielido elektroni.

$v_0 = 10^5 \text{ m/s}$; $h = 4 \cdot 10^{-10} \text{ s}$; $t = 2 \cdot 10^{-6} \text{ s}$:

- (1) Aprēķins pārbaudei (realizēts C++), starp daļinām darbojas Kulona spēks.
 - (2) OpenFOAM aprēķins. Daļinas mijiedarbojas ar elektriskā lauka intensitāti \mathbf{E} .
 - (3) OpenFOAM aprēķinam (2) atbilstošs \mathbf{E} .
- (a) daļiņu skaits $\sim 10^4$, (b) superdaļiņu skaits $\sim 10^3$, katra superdaļiņa = 10 daļinās.



Att. 5.9: Aprēķins vienai sprauslai, pa kuru ielido elektroni.

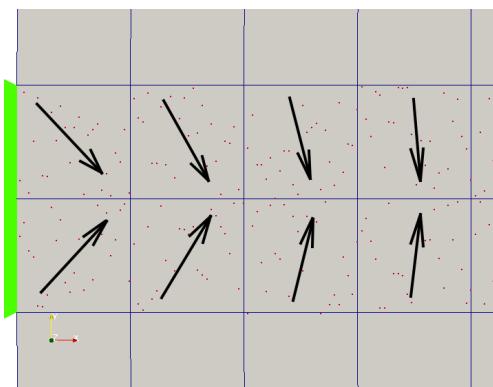
$$v_0 = 10^5 \text{ m/s}; h = 4 \cdot 10^{-10} \text{ s}; t = 2 \cdot 10^{-6} \text{ s};$$

- (1) Aprēķins pārbaudei (realizēts C++), starp daļinām darbojas Kulona spēks.
 - (2) OpenFOAM aprēķins. Daļinas mijiedarbojas ar elektriskā lauka intensitāti \mathbf{E} .
 - (3) OpenFOAM aprēķinam (2) atbilstošs \mathbf{E} .
- (a) daļinu skaits $\sim 10^4$, (b) superdaļinu skaits $\sim 10^3$, katras superdaļinā $= 10$ daļinas.

Secinājumi

No rezultātiem 5.8. un 5.9. attēlos redzams, ka abos gadījumos ar superdaļinām veiktais aprēķins (2)(a) kvalitatīvi atbilst ar vienkāršām daļinām veiktajam aprēķinam (2)(b). Tas ļaus ar superdaļinām aizstāt miljonus reālu daļiņu, kas ir nepieciešams, bet ne pietiekams nosacījums, lai varētu modelēt lādētu daļiņu dinamiku lielām sistēmām.

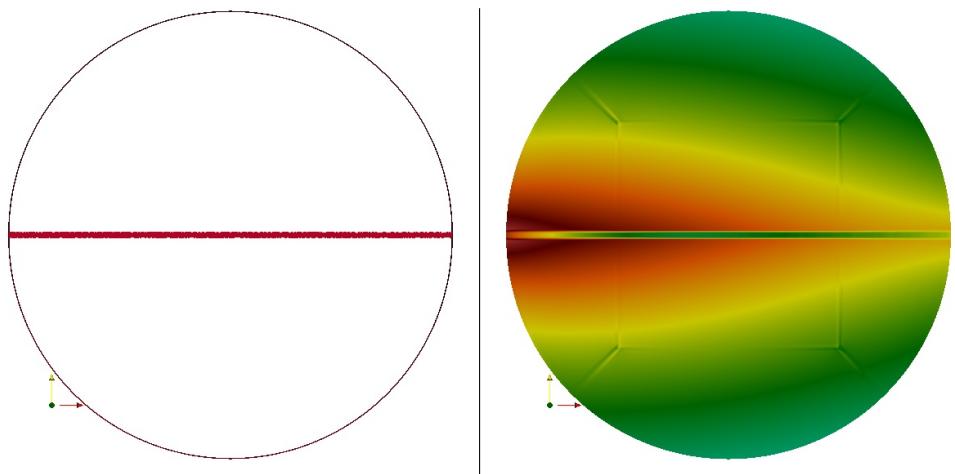
Abos attēlos redzams, ka (2) aprēķini neatbilst uz Kulona spēku balstītajiem aprēķiniem (1). Atšķirība ir viena kūla sašķelšanās divos kūlos. Iemesls tam ir daļiņu lādiņa sadalījums. Tā kā viss vienas šūnas lādiņš pieder punktam šūnas centrā, tad nav iespējams modelēt mijiedarbību starp atsevišķām vienas šūnas daļinām, līdz ar to uz šūnā esošām daļinām iedarbojas \mathbf{E} , ko rada blakus šūnās esošās daļiņas (5.10. att.).



Att. 5.10: Fragments no aprēķina 5.9. (2)(a) attēlā. Elektroni - sarkani punkti, ieja - zaļa trapece, melnie vektori ir elektriskās intensitātes \mathbf{E} vektori (aptuvens virziena novērtējums), kas ir definēti šūnu centros un darbojas uz visiem šūnā esošiem elektroniem.

Attēlā 5.11. ir attēlots aprēķinam 5.9. (2)(a) un (3)(a) identisks aprēķins, tikai šoreiz ieejas divas šūnas ir sapludinātas vienā. Rezultāts parāda - realizēto daļiņu mijiedarbības modeli nedrīkst izmantot liela lādiņa blīvuma gradiента gadījumā (lādiņam vienmērīgi jāaizpilda aprēķinu apgabals), kā arī aprēķinu precizitāte uzlabojas, ja katras lādēta daļiņa aizņem savu režģa šūnu, proti, būtiski jāsmalcina režģis (to var secināt no 5.8. un 5.9. attēliem). Turpretim, lai DSMC varētu veikt daļiņu sadursmes un kīmiskās reakcijas, optimāls daļiņu skaits šūnā ir vismaz 20 [6]. Līdz ar to elektriski lādētu daļiņu savstarpējo mijiedarbību ieteicams modelēt gadījumos, kad lādiņa blīvuma sadalījums telpā ir vienmērīgs.

Magnetronu sistēmās lādiņa blīvuma gradients ir nesalīdzināmi mazāks par pieņēmos aplūkoto, jo plazma sastāv gan no pozitīvajiem argona joniem, gan negatīvajiem elektroniem. Acīmredzot, veicot vairākus šādus vienkāršotus testus, iespējams atrast likumsakarību starp daļiņu savstarpējās mijiedarbības klūdu (šobrīd nedefinēts lielums), kura atkarīga no lādiņa blīvuma gradiента un šūnu izmēra.

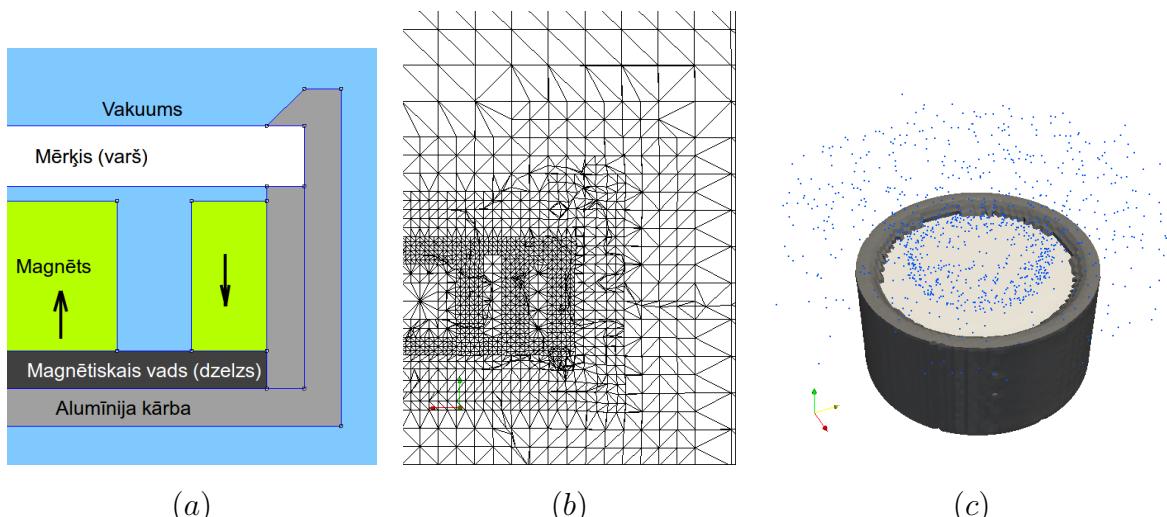


Att. 5.11: Attēlā 5.9. (2)(a) un (3)(a) attēlotajam aprēķinam līdzīgs gadījums, taču daļas tiek ielaistas pa vienu režģa šūnu lielu spraugu.

5.3.3. Elektronu dinamika magnetronu sistēmā

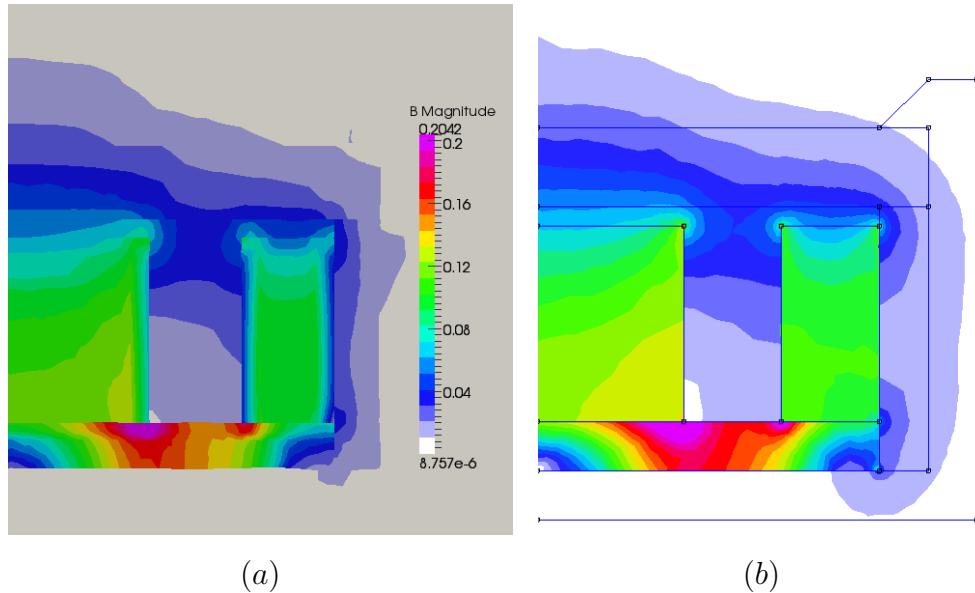
Tika veikts viens aprēķins vienkāršotai cirkulāra magnetrona sistēmai. Aprēķina mērķis ir tehnisks, proti, pārbaudīt izstrādātās programmas iespējas. Šobrīd ir iespējams definēt atvērtos daļu robežnosacījumus, pa kuriem var ieplūst un izplūst daļas, kā arī sienas, no kurām daļas var atsisties gan no klasiskās mehānikas, gan statistiskās termodinamikas viedokļa.

Elekroni tiek izšauti no mērķa augšup ar ātrumu 10^5 m/s . Mērķa virsmas elektriskais potenciāls $\phi = -10 \text{ V}$, magnetrona korpuiss ir zemēts $\phi = 0 \text{ V}$ (5.12. (a) att.). Attēlā 5.12. (b) attēlotas daļas, īpaša uzmanība jāpievērš tam, ka virs mērķa veidojas riņķis ar lielāku elektronu blīvumu, tā iemesls ir paralēli mērķa virsmai vērstais magnētiskais lauks, kurš satver vieglos elektronus.

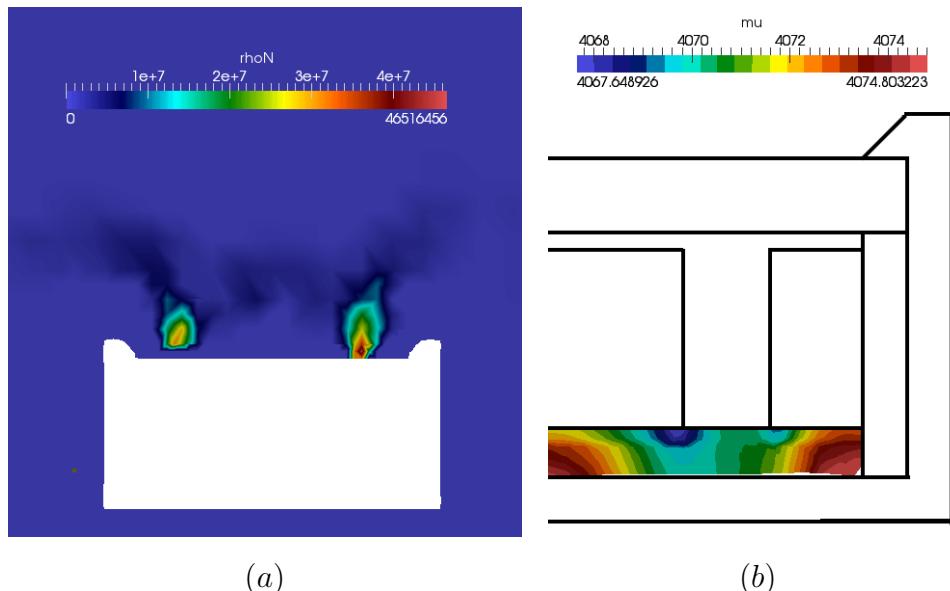


Att. 5.12: Aprēķins cirkulāram magnetronam ar ne-feromagnētisku mērķi: (a) konstrukcija (ar bultām attēloti magnētu magnetizācijas virzieni), (b) aprēķinu režģa fragments, (c) magnetronu sistēma - mērķis (balts) un elektroni (zili), magnetrona korpuiss (peleks).

Attēlā 5.13. (a) un (b) salīdzināts magnētiskā lauka aprēķins ar programmu `multiRegionPotencFoam` un FEMM aksiālsimetriskam gadījumam. Šajā aprēķinā netiek ņemts vērā lādētu daļiņu radītais magnētiskais lauks, jo daļiņu koncentrācija, kura attēlotā 5.14. (a) attēlā ir ļoti maza. Magnētiskā vada magnētiskās caurlaidības μ sadalījums attēlots 5.14. (b) attēlā.



Att. 5.13: Aprēķins cirkulāram magnetronam ar ne-feromagnētisku mērķi: (a) OpenFOAM aprēķinātā magnētiskā lauka indukcija \mathbf{B} , (b) FEMM aprēķinātā magnētiskā lauka indukcija \mathbf{B} . Abu risinājumu skalas ir vienādas.



Att. 5.14: Aprēķins cirkulāram magnetronam ar ne-feromagnētisku mērķi: (a) daļiņu blīvums, (b) magnētiskā vada magnētiskās caurlaidības sadalījums.

6. Secinājumi

Bakalaura darbā tika turpināta magnētiskā lauka aprēķinu programmas izstrāde, kas varētu tikt izmantota magnetronu konstrukcijas optimizēšanai. Tika apgūta aprēķinu bibliotēka `OpenFOAM`, dažādi modelešanas piemēri, apzinātas bibliotēkas iespējas un trūkumi. Darba uzsākšanas brīdī izvirzītais mērķis tika sasniegts daļēji. Saskaņā ar izvirzītajiem uzdevumiem var secināt:

- Darba gaitā izstrādātā programma `multiRegionPlasmaFoam` var skaitliski modelēt kvazistacionāru magnētisko un elektrisko lauku sarežģītās magnetronu uzputināšanas sistēmās. Rezultāti tika salīdzināti ar analogiskiem risinājumiem, kuri veikti ar 2D programmu `FEMM` aksiālsimetriskiem gadījumiem, kā arī ar analītiskiem novērtējumiem. Tika apzināti daži speciālgadījumi, kuros programma rēķina neprecīzi. Aprēķini liecina, ka neprecizitātes iemesls ir saistīts ar robežnosacījumu uzdošanu uz apgabalu ar dažādu magnētisko caurlaidību robežām. Iespējams, šo problēmu var novērst, lietojot citu formulējumu, piemēram, " T, Ω, Ω ". Šis formulējums tika realizēts programmā, taču darba gaitā neizdevās atrisināt konvergences problēmas.
- Darba gaitā tika izstrādāta metodika, kas ļauj dažādas sarežģītas ģeometrijas ievadīt datorā, sadalīt nepieciešamajos apgabalos, aizpildīt to ar 3D režīgi, uzdot nepieciešamās materiālu īpašības un robežnosacījumus. Uzdevuma realizēšanā izmantotas gan `OpenFOAM` iebūvētās komponentes - `snappyHexMesh`, `blockMesh`, gan citas programmas, piemēram, `Salome`. Aprēķinu piemēra uzdošanas vienkāršošanai tika uzrakstīti skripti `Python` un `Bash` valodās.
- Tika realizēta feromagnētisku materiālu magnētiskās caurlaidības atkarības no magnētiskā lauka indukcijas ķēmšana vērā aprēķinu laikā.
- Tika atrasti piemērotākie aprēķinu apgabala ārējie robežnosacījumi skalārajiem un vektoriālajiem potenciāliem, kas imitē bezgalīgi lielu aprēķinu apgabalu.
- Izmantojot `OpenFOAM` izstrādāto aprēķinu programmu `dsmcFoam`, programmā `multiRegionPlasmaFoam` tika iestrādāta iespēja modelēt retinātu gāzu dinamiku. Programma veiksmīgi var modelēt lādētu daļiņu uzvedību ārējā elektromagnētiskajā laukā. Rezultāti tika salīdzināti ar analītisku risinājumu un sakritība bija laba. Darbā netika pilnībā atrisināts uzdevums modelēt lādētu daļiņu savstarpējo mijiedarbību. `OpenFOAM` ir realizēta galīgo tilpumu metode, kas ne visos gadījumos ļauj realizēt kāda fizikāla procesa matemātisku formulējumu. Daļiņu savstarpējo mijiedarbību kulona spēka dēļ tipiski modelē ar *Particle-In-Cell* metodi, kura daļiņas elektrisko lādiņu "sadala" pa režīga virsotnēm atkarībā no to atrašanās vietas šūnā, tad elektromagnētiskā lauka aprēķinu veic ar galīgo diferenču vai elementu metodi. `OpenFOAM` vecākajās versijās, piemēram, 1.6 bija pieejams papildinājums aprēķiniem

galīgajos elementos. Jaunākajās OpenFOAM versijās papildinājums aprēķiniem galīgajos elementos nav pieejams. Acīmredzot, OpenFOAM primāri paredzēts šķidrumu, gāzu, siltuma plūsmas modelēšanai.

Turpmākais darbs saistīts ar matemātiski korektu robežnosacījumu, ķīmisko reakciju, pilnīgas Particle-In-Cell metodes un atrisinājumu konverģences kritēriju ieviešanu.

7. Pateicība

Bakalaura darba izstrāde uzsākta septembrī. Šajā laika posmā esmu uzkrājis plašu zināšanu, prasmju un pieredzes kapitālu, vēl esmu apzinājis savas intereses un izlēmis par turpmākām studiju gaitām, visticamāk, skaitliskā modelēšana un elektrofizika klūs par manu hobiju, darbu un neatņemamu dzīves sastāvdaļu. Es apzinos, cik svarīga ir baka- laura darba izvēle studentam kā indivīdam. Esmu vadījies pēc savām interesēm, mērķiem un kompetences, rezultātā nonācu Latvijas augsto tehnoloģiju uzņēmumā **SIDRABE**, kur mani ļoti labi uzņēma. Ne reizi neesmu šaubījies par izvēli izstrādāt darbu šajā uzņēmu- mā, bija prieks būt citādā, mazāk akadēmiskā, neformālā vidē, savām acīm redzēt kā un kur gūtās zināšanas cilvēki patiešām liek lietā, materializē.

Līdz ar to vēlos pateikties uzņēmuma **SIDRABE** zinātniskajam personālam par atsaucību un palīdzību manai kompetencei un interesēm piemērota darba atrašanā.

Taču īpaši vēlos pateikties manam darba vadītājam **Kaspāram Lācim**, viņš ir pie- dāvājis ļoti interesantu tematu bakalaura darbam, un vēl var piedāvāt citus, perspektīvus darbus. Faktiski tiek likti pamati Latvijā jaunam kompetences centram - retinātu gāzu un plazmas modelēšana. Jaunas idejas un projekta uzsākšana ir liela atbildība, risks. Iespējams, nāksies vēl neskaitāmas reizes mainīt metodes, sākt šo-to no sākuma, taču būtiskākais šajā procesā ir vienmēr meklēt, krāt pieredzi un dalīties ar to. Paldies par laiku, ko veltījāt darba vadīšanai, konstruktīvām diskusijām un pieredzes apmaiņai!

Literatūra

Žurnāli

- [1] **A. Bogaerts, E. Bultinck, I. Kolev, L. Schwaeder, K. Van Aeken, G. Buyle, D. Depla.** Computer modelling of magnetron discharges. *Journal of Physics D: Applied Physics*, 42(2009), p.12.
- [2] **R. Acevedo, R. Rodriguez.** Analysis of the A,V-A- ψ potential formulation for the eddy current problem in a bounded domain. *Electronic Transactions on Numerical Analysis*. Volume 26(2007), p.270-284.

Grāmatas

- [3] **D. M. Mattox.** Handbook of Physical Vapor Deposition (PVD) processing. Noyes Publications, 1998. p. 288-406
- [4] **Y. Pauleau.** Materials Surface Processing by Directed Energy Techniques. 2006, Elsevier. p. 67-106
- [5] **G. A. Bird.** Molecular Gas Dynamics and the Direct Simulation of Gas Flows. Oxford: Clarendon Press, 1994.
- [6] **C. Shen.** Rarefied Gas Dynamics. Fundamentals, Simulations and Micro Flows. Springer, 2005.

Konferenču tēzes

- [7] **M. Siemers, A. Pflug, B. Szyska.** Particle-in-cell Monte Carlo Analysis of Anomalous Target Erosion in Magnetron Sputtering. Society of Vacuum Coaters, 51st Annual Technical Conference Proceedings, Chicago, 2008. (p. 43-48).
- [8] **M. Siemers, A. Pflug, B. Szyska.** Three Dimensional Plasma Simulations with a Parallelized Particle-in-Cell Monte Carlo Approach. Society of Vacuum Coaters, 51st Annual Technical Conference Proceedings, Chicago, 2006. (p. 216-219).
- [9] **T. Hauser, J. Allen.** foamDSMC - A DSMC solver for rarefied flow applications based on OpenFOAM. OpenFOAM International Conference 2007, 2007.

Disertācijas, maģistru un bakalauru darbi

- [10] **M. Sarama.** Magnētiskā lauka 3D modelešana magnetronu izputināšanas sistēmās. Bakalaura darbs. Latvijas Universitāte, Rīga, 2012.
- [11] **Z. Huang.** OpenFOAM Simulation for Electromagnetic Problems. Master of Science Thesis. Chalmers University of Technology, Goteborg, 2010.
- [12] **C. S. MacLachlan.** Numerical modelling of low temperature plasma. Ph.D. Thesis. University of Glasgow, Glasgow, 2009.

Citi darbi

- [13] **P. Testoni.** Implementation in the ANSYS finite element code of the electric vector potential $T-\Omega,\Omega$ formulation and its validation with the magnetic vector potential $A-V,A$ formulation. University of Cagliari, Gennaio, 2003.
- [14] **D. Meeker.** FEMM 4.2 Magnetostatic Tutorial. (Based on the "Introduction to FEA with FEMM" tutorial by Ian Stokes-Rees, TSS (UK) Ltd.), 2006.
- [15] **W. Gulbinski.** Physical Vapour Deposition of Thin Film Coatings. Prezentācija, Koszalin University of Technology, Koszalin. (2008).
- [16] **A. Muižnieks, K. Lācis.** Elektromagnētisms. Lekciju konspekts kursam vispārīgajā fizikā, Latvijas Universitāte, Rīga. (2003).
- [17] **R. Bargallo.** Finite Elements for Electrical Engineering. Konspekts, Universitat Politècnica de Catalunya. (2006).
- [18] **COMSOL 4.3.** AC/DC Module. User's Guide, COMSOL. (2012).

Interneta resursi

- [19] <http://www.openfoam.org/docs/user/snappyHexMesh.php>
OpenFOAM Foundation. Mesh generation with the snappyHexMesh utility, 19.03.2013.
- [20] <http://www.salome-platform.org/user-section/salome-tutorials>
OPEN CASCADE. SALOME Tutorials.
- [21] <http://salome-platform.org/user-section/documentation/current-release>
OPEN CASCADE. SALOME documentation.
- [22] <http://www.cfd-online.com/Forums/openfoam-programming-development/>
Anja Miehe. Read temperature dependent thermophysical properties from a file.
- [23] <http://www.integratedsoft.com/Tools/Magnetization-Curve>
INTEGRATED Engineering Software. Generated Magnetization Curve.
- [24] http://en.wikipedia.org/wiki/Navier-Stokes_equations
Wikipedia. Navier–Stokes equations.
- [25] <http://www.particleincell.com/>
Particle In Cell Consulting. Plasma Physics and Scientific Computing Blog.
- [26] http://en.wikipedia.org/wiki/Runge-Kutta_methods
Wikipedia. Runge–Kutta methods.

OpenFOAM standarta programmas

- [27] `run/tutorials/heatTransfer/chtMultiRegionFoam/snappyMultiRegionHeater`
OpenFOAM 2.1.1. snappyMultiRegionHeater. Programmas inicializācijas piemērs.
- [28] `/opt/openfoam211/applications/solvers/heatTransfer/chtMultiRegionFoam`
OpenFOAM 2.1.1. chtMultiRegionFoam. Programmas piemērs.
- [29] `/opt/openfoam211/applications/solvers/electromagnetics/magneticFoam`
OpenFOAM 2.1.1. magneticFoam. Programmas piemērs.
- [30] `/opt/openfoam211/applications/solvers/discreteMethods/dsmc/dsmcFoam`
OpenFOAM 2.1.1. dsmcFoam. Programmas piemērs.
- [31] `/opt/openfoam211/src/lagrangian/dsmc`
OpenFOAM 2.1.1. dsmc. Bibliotēka.
- [32] `run/tutorials/discreteMethods/dsmcFoam/freeSpaceStream`
OpenFOAM 2.1.1. freeSpaceStream. Programmas inicializācijas piemērs.

8. Pielikums

8.1. Python programma ģeometrijas apstrādei un virsmas režģu ķēršanas programmā Salome

```
import geompy
import salome
import smesh

gg = salome.ImportComponentGUI("GEOM")

# 'box_tmp' ir vesels kubs, 'air_tmp' ir vesela sfera
box = salome.myStudy.FindObjectByPath("/Geometry/box_tmp").GetObject()
air = salome.myStudy.FindObjectByPath("/Geometry/air_tmp").GetObject()

box = geompy.MakeCut(box, air)
id_box = geompy.addToStudy(box, "box")
gg.createAndDisplayGO(id_box)
gg.setDisplayMode(id_box,2)

solids = ['pipe', 'electrode1', 'electrode2'] # aprekinu apgabali
for solid in solids:
    object0 = salome.myStudy.FindObjectByPath("/Geometry/" + solid).GetObject()
    air = geompy.MakeCut(air, object0)

id_air = geompy.addToStudy(air, "air")
gg.createAndDisplayGO(id_air)
gg.setDisplayMode(id_air,2)

objectBox = salome.myStudy.FindObjectByPath("/Geometry/box").GetObject()
listBox = geompy.ExtractShapes(objectBox, geompy.ShapeType["SHELL"], True)
Box = geompy.ExtractShapes(listBox[1], geompy.ShapeType["FACE"], True)
boxNames = ['minX', 'minY', 'minZ', 'maxZ', 'maxY', 'maxX']
for i, v in enumerate(Box):
    id_temp_common = geompy.addToStudy(v, boxNames[i])
    gg.createAndDisplayGO(id_temp_common)
    gg.setDisplayMode(id_temp_common,2)
    triaN = smesh.Mesh(v, boxNames[i])
    algo2D = triaN.Triangle(smesh.NETGEN_1D2D)
    n12_params = algo2D.Parameters()
        # 'box' arejo skaldnu virsmas rezga smalcinasanas parametrs
    n12_params.SetMaxSize(0.03)
    triaN.Compute()
        # virsmas rezga saglabasanas adrese
    triaN.ExportSTL("/home/juris/piemers/" + boxNames[i] + ".stl", 1)

solids[0:0] = ['box', 'air']
for solid1 in solids:
    for solid2 in solids:
        if solid1 != solid2:
            object1 = salome.myStudy.FindObjectByPath("/Geometry/" + solid1).GetObject()
            list1 = geompy.ExtractShapes(object1, geompy.ShapeType["SHELL"])
            object2 = salome.myStudy.FindObjectByPath("/Geometry/" + solid2).GetObject()
            list2 = geompy.ExtractShapes(object2, geompy.ShapeType["SHELL"])
            for i in list1:
                for j in list2:
```

```

temp_common = geompy.MakeCommon(i, j)
prop = geompy.BasicProperties(temp_common)
if prop[1]>0:
    id_temp_common = geompyaddToStudy(temp_common, solid1+'_to_'+solid2)
    gg.createAndDisplayGO(id_temp_common)
    gg.setDisplayMode(id_temp_common,2)
    if (solid1=='box' and solid2=='air') or (solid1=='air' and solid2=='box'):
        meshBox = smesh.Mesh(temp_common, solid1+'_to_'+solid2)
        algoBox = meshBox.Segment()
        # 'box' un 'air' kopigas virsmas rezga smalcinasanas parametrs
        algoBox.MaxValue(0.01)
        algoBox = meshBox.Triangle(smesh.MEFISTO)
        meshBox.Compute()
        # virsmas rezga saglabasanas adrese
        meshBox.ExportSTL("/home/juris/piemers/"+solid1+'_to_'+solid2+".stl", 1)
    else:
        meshOther = smesh.Mesh(temp_common, solid1+'_to_'+solid2)
        algoOther = meshOther.Segment()
        # parejo virsmas rezgu smalcinasanas parametrs
        algoOther.MaxValue(0.001)
        algoOther = meshOther.Triangle(smesh.MEFISTO)
        meshOther.Compute()
        # virsmas rezga saglabasanas adrese
        meshOther.ExportSTL("/home/juris/piemers/"+solid1+'_to_'+solid2+".stl", 1)

```

8.2. Bash skripts .stl failu apvienošanai

```

#!/bin/sh
for n in air pipe electrode1 electrode2 box
do
    for i in $n*
    do
        if [ "${i##*.}" = "stl" ] ; then
            sed s/'solid'/'solid '${i%.*}'/ <$i >> $n.stl
        fi
    done
done

for n in min max
do
    for i in $n*
    do
        sed s/'solid'/'solid '${i%.*}'/ <$i >> box.stl
    done
done

```

8.3. multiRegionPlasmaFoam.C faila saturs

```
#include "fvCFD.H"
#include "dsmcCloud.H"
#include "regionProperties.H"
#include "PtrList.H"
#include "volFields.H"
#include "typeInfo.H"
#include "IOdictionary.H"
#include "autoPtr.H"
#include "runTimeSelectionTables.H"
#include "electromagneticConstants.H"
#include "AMixedFvPatchVectorField.H"
#include "graph.H"
#include "interpolateXY.H"

int main(int argc, char *argv[])
{
    #include "setRootCase.H"
    #include "createTime.H"

    regionProperties rp(runTime);

    #include "createEmptyMeshes.H"
    #include "createAirMeshes.H"
    #include "createLinearMeshes.H"
    #include "createNonlinearMeshes.H"

    #include "createAirFields.H"
    #include "createLinearFields.H"
    #include "createNonlinearFields.H"

    while (runTime.loop())
    {
        Info<< "Time = " << runTime.timeName() << nl << endl;

        forAll(linearRegions, i)
        {
            Info<< "\nSolving for linear region "
                << linearRegions[i].name() << endl;
            #include "setRegionLinearFields.H"
            #include "solveLinear.H"
        }

        forAll(nonlinearRegions, i)
        {
            Info<< "\nSolving for nonlinear region "
                << nonlinearRegions[i].name() << endl;
            #include "setRegionNonlinearFields.H"
            #include "interpolateProperties.H"
            #include "solveNonlinear.H"
        }

        forAll(airRegions, i)
        {
            Info<< "\nSolving for air region "
                << airRegions[i].name() << endl;
        }
    }
}
```

```

#include "setRegionAirFields.H"
#include "solveAir.H"
}

runTime.write();

Info<< "ExecutionTime= " << runTime.elapsedCpuTime() << "s"
<< "ClockTime= " << runTime.elapsedClockTime() << "s"
<< nl << endl;
}

Info<< "End\n" << endl;

return 0;

```

8.4. OpenFOAM failu saturs μ atkarības no B uzdošanai

(a) Augšējam mērķim

```

(
0 1000
0.0607187659 1000.6933030986
0.0894434019 1000.3342962083
0.1317017027 999.5557726323
0.1937490669 997.8702011393
0.2844691084 994.233375151
0.415911267 986.4446094062
0.6026881248 970.02615607
0.8574679867 936.5425585477
1.1773348014 872.6264579119
1.5201843897 764.6161302522
1.8001248359 614.4246106878
1.9516775283 452.0566194677
1.9994705688 314.281327875
2.0090359403 214.2940248078
2.0138112529 145.767233394
2.0203582272 99.2403575845
2.05 16.3133816674
2.1 8.3556345126
2.5 3.9788735774
3 2.3873241464
4 1.061032954
5 1.0202239942
10 1.0073097664
)

```

(b) Apakšējam mērķim

```

(
0 1000
0.0792829997 1000.8692891671
0.1199682131 1000.7006815338
0.1814921951 1000.3147531983
0.2744314507 999.4321542189
0.4144927103 997.4176489677
0.6244223613 992.8401106889
0.9352122002 982.5434077524
1.3827500095 959.8999475567
1.9893113519 912.4850844126
2.7141087508 822.6042972773
3.396272024 680.1534601373
3.8287958466 506.6492097667
3.9852466962 348.4504945712
4.0159146163 232.012327479
4.0261794675 153.6952463486
4.0396453068 101.8946203402
4.05 32.228875977
4.07 6.477606184
4.1 3.2626763335
4.5 1.1936620732
5 1.0753712371
10 1.0073097664
)
```

Bakalaura darbs “Lādētu daļiņu kustības magnētiskā laukā 3D modelēšana OpenFOAM vidē” izstrādāts uzņēmumā AS “Sidrabe”.

Ar savu parakstu apliecinu, ka pētījums veikts patstāvīgi, izmantoti tikai tajā norādītie informācijas avoti un iesniegtā elektroniskā kopija atbilst izdrukai.

Autors: *Juris Venčels* _____
(paraksts) (datums)

Rekomendēju darbu aizstāvēšanai.

Vadītājs: *Dr. phys. Kaspars Lācis* _____
(paraksts) (datums)

Recenzents: *asoc. prof. Andris Jakovičs*

Darbs iesniegts Fizikas nodalā _____
(datums)

Dekāna pilnvarotā persona: vecākā metodiķe Dzintra Holsta _____
(paraksts)

Darbs aizstāvēts bakalaura gala pārbaudījumu komisijas sēdē

(datums) (protokola nr.) (vērtējums)

Komisijas sekretārs(-e): _____
(vārds, uzvārds) (paraksts)