



Fakulta elektrotechnická
Katedra aplikované elektroniky a telekomunikací

BAKALÁŘSKÁ PRÁCE

Využití programu Agros2D pro analýzu fyzikálních polí

Autor práce: Petr Pedál
Vedoucí práce: Ing. Roman Hamar, Ph.D.

Plzeň 2013

ZÁPADOČESKÁ UNIVERZITA V PLZNI

Fakulta elektrotechnická

Akademický rok: 2012/2013

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Petr PEDÁL**
Osobní číslo: **E10B0333P**
Studijní program: **B2612 Elektrotechnika a informatika**
Studijní obor: **Elektronika a telekomunikace**
Název tématu: **Využití programu Agros2D pro analýzu fyzikálních polí**
Zadávající katedra: **Katedra aplikované elektroniky a telekomunikací**

Zásady pro vypracování:

1. Uveďte vlastnosti programu Agros2D a popište základní ovládání a princip 2D modelování v tomto programu.
2. Alespoň pro tři typy fyzikálních polí vytvořte ilustrativní příklady s podrobným popisem postupu modelování úlohy. V rámci každého vybraného fyzikálního pole vytvořte příklad pro kartézský i osově symetrický typ problému.
3. Vytvořte alespoň dva ilustrativní příklady zaměřené na použití Python skriptů.
4. Zhodnoťte Vaše dosažené zkušenosti s modelováním v programu Agros2D a navrhněte případná další vylepšení tohoto programu.

Rozsah grafických prací: **podle doporučení vedoucího**

Rozsah pracovní zprávy: **20 - 30 stran**

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam odborné literatury:

1. Mayer, D.: **Teorie elektromagnetického pole. Skripta ZČU, Plzeň, 2004.**
2. Benešová, Z., Mayer, D.: **Základní příklady z teorie elektromagnetického pole. Skripta ZČU, Plzeň, 2008.**
3. **www.agros2d.org**
4. **www.python.org**

Vedoucí bakalářské práce: **Ing. Roman Hamar, Ph.D.**

Katedra teoretické elektrotechniky

Konzultant bakalářské práce: **Ing. Roman Hamar, Ph.D.**

Katedra teoretické elektrotechniky

Datum zadání bakalářské práce: **15. října 2012**

Termín odevzdání bakalářské práce: **7. června 2013**



L.S.

Doc. Ing. Jiří Hammerbauer, Ph.D.

děkan

Doc. Dr. Ing. Vjačeslav Georgiev

vedoucí katedry

V Plzni dne 15. října 2012

Abstrakt

Bakalářská práce je zaměřena na využití simulačního programu Agros2D. Jejím cílem je uvedení vlastností programu, popis ovládání prostředí programu a vytvoření několika ilustrativních příkladů s podrobným popisem postupu modelování úlohy. Bakalářská práce obsahuje vytvořené ilustrativní příklady pro různé druhy fyzikálních polí v kartézských i v osově symetrických souřadnicích. Sedm příkladů je vytvořeno pomocí grafického uživatelského prostředí. Poslední dva příklady jsou zaměřeny na využití Python skriptů.

Klíčová slova

Agros2D, simulace, Python skript, magnetické pole, elektrické pole

Abstract

Pedál, Petr. *Use of the Agros2D for the Analysis of Physical Fields [Využití programu Agros2D pro analýzu fyzikálních polí]*. Pilsen, 2013. Bachelor thesis (in Czech). University of West Bohemia. Faculty of Electrical Engineering. Department of Applied Electronics and Telecommunications. Supervisor: Roman Hamar

The thesis is focused on the use of simulation software Agros2D. Its aim is to introduce the software to describe the user interface of the program and to create a few illustrative examples detailing the process of modeling. The thesis contains illustrative examples specially created for different types of physical field in Cartesian coordinates and in axially symmetric coordinates. Seven examples have been created by using the graphical user interface. The last two examples show the use of the Python scripting.

Keywords

Agros2D, simulation, Python script, electric field, magnetic field

Prohlášení

Předkládám tímto k posouzení a obhajobě bakalářskou práci, zpracovanou na závěr studia na Fakultě elektrotechnické Západočeské univerzity v Plzni.

Prohlašuji, že jsem svou závěrečnou práci vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce. Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této závěrečné práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 270 trestního zákona č. 40/2009 Sb.

Také prohlašuji, že veškerý software, použitý při řešení této bakalářské práce, je legální.

V Plzni dne 6. června 2013

Petr Pedál

.....

Podpis

Poděkování

Tímto bych chtěl poděkovat vedoucímu mé bakalářské práce Ing. Romanu Hamarovi, Ph.D. za cenné rady, připomínky a metodické vedení práce. Také bych rád poděkoval svým blízkým, kteří mě ve studiu podporovali, hlavně svým rodičům.

Obsah

Seznam obrázků	xii
Seznam tabulek	xiv
Seznam symbolů a zkratk	xv
Úvod	1
1 Vlastnosti simulačního programu	2
1.1 Program Agros2D	2
1.1.1 Hlavní autoři	2
1.1.2 Oficiální webové stránky	3
1.2 Knihovna Hermes2D	3
1.2.1 Schopnosti knihovny z matematického pohledu	3
1.2.2 Schopnosti knihovny z pohledu realizace	3
1.2.3 Metoda hp-FEM	4
1.3 Řešitelná fyzikální pole	4
1.4 Moduly programu	5
1.4.1 Modul preprocesor	5
1.4.2 Modul procesor	5
1.4.3 Modul postprocesor	5
1.5 Porovnání s konkurencí	6
2 Ovládání uživatelského prostředí	7
2.1 Základní rozvržení prostředí	7
2.2 Horní lišta	8
2.3 Nastavení problému	10
2.4 Graf	10
2.5 Editor skriptů	10
3 Princip 2D modelování	11
3.1 Definice modelu	11
3.2 Nastavení problému	11

3.3	Vytvoření geometrie	11
3.4	Doplnění fyzikálních podmínek	12
3.5	Diskretizace a řešení	12
4	Model feromagnetického aktuátoru	13
4.1	Formulace problému	13
4.2	Definice modelu	14
4.2.1	Rozdělení oblastí	14
4.2.2	Okrajové podmínky	14
4.3	Vytvoření modelu	14
4.3.1	Nastavení problému	14
4.3.2	Vytvoření konstant	14
4.3.3	Vytvoření geometrie	15
4.3.4	Doplnění fyzikálních podmínek	16
4.3.5	Diskretizace a řešení	17
4.4	Diskuze výsledků	18
4.4.1	Rozložení výsledného pole	18
4.4.2	Zobrazení výsledků do grafu	21
4.4.3	Výpočet síly působící na kotvu	22
5	Python skript	24
5.1	Příkazy a klíčová slova	24
5.2	Ostatní příkazy	24
6	Zkušenosti a vylepšení programu	25
6.1	Dosažené zkušenosti	25
6.2	Návrh na vylepšení programu	26
	Závěr	27
	Literatura	28
	Přílohy	29
A	Indukční ohřev ocelového válce	29
A.1	Formulace problému	29
A.2	Definice modelu	30
A.2.1	Rozdělení oblastí	30
A.2.2	Okrajové podmínky	30
A.3	Vytvoření modelu	31
A.3.1	Nastavení problému	31
A.3.2	Vytvoření konstant	31

A.3.3	Vytvoření geometrie	32
A.3.4	Doplnění fyzikálních podmínek	32
A.3.5	Diskretizace a řešení	33
A.4	Diskuze výsledků	34
A.4.1	Rozložení výsledného pole	34
A.4.2	Výpočet Jouleových ztrát	35
A.4.3	Hloubka vniku	37
B	Deskový kondenzátor s respektováním okrajového jevu	38
B.1	Formulace problému	38
B.2	Definice modelu	39
B.2.1	Rozdělení oblastí	39
B.2.2	Okrajové podmínky	39
B.3	Vytvoření modelu	39
B.3.1	Nastavení problému	39
B.3.2	Vytvoření konstant	39
B.3.3	Vytvoření geometrie	40
B.3.4	Doplnění fyzikálních podmínek	41
B.4	Diskuze výsledků	41
B.4.1	Diskretizace a řešení	41
B.4.2	Rozložení výsledného pole	42
B.5	Výpočet kapacity kondenzátoru	45
C	Model elektromagnetu	46
C.1	Formulace problému	46
C.2	Definice modelu	46
C.2.1	Rozdělení oblastí	46
C.2.2	Okrajové podmínky	47
C.3	Vytvoření modelu	47
C.3.1	Nastavení problému	48
C.3.2	Vytvoření konstant	48
C.3.3	Vytvoření geometrie	48
C.3.4	Doplnění fyzikálních podmínek	50
C.3.5	Diskretizace a řešení	50
C.4	Diskuze výsledků	52
C.4.1	Rozložení výsledného pole	52
C.4.2	Zobrazení výsledků do grafu	55
D	Koaxiální kabel s nedokonalou izolací	56
D.1	Formulace problému	56
D.2	Definice modelu	57

D.2.1	Rozdělení oblastí	57
D.2.2	Okrajové podmínky	57
D.3	Vytvoření modelu	57
D.3.1	Nastavení problému	57
D.3.2	Vytvoření konstant	57
D.3.3	Vytvoření geometrie	58
D.3.4	Doplnění fyzikálních podmínek	59
D.3.5	Diskretizace a řešení	59
D.4	Diskuze výsledků	60
D.4.1	Rozložení výsledného pole	60
D.4.2	Zobrazení výsledku pomocí grafu	62
D.4.3	Ztráty v izolaci	62
E	Dvouodičové pásové vedení	64
E.1	Formulace problému	64
E.2	Definice modelu	65
E.2.1	Rozdělení oblastí	65
E.2.2	Okrajové podmínky	65
E.3	Vytvoření modelu	65
E.3.1	Nastavení problému	65
E.3.2	Vytvoření konstant	66
E.3.3	Vytvoření geometrie	66
E.3.4	Doplnění fyzikálních podmínek	67
E.3.5	Diskretizace a řešení	67
E.4	Diskuze výsledků	68
E.4.1	Rozložení výsledného pole	68
E.4.2	Zobrazení výsledků do grafu	71
E.4.3	Jev blízkosti a skin efekt	72
F	Kulové jiskřiště	74
F.1	Formulace problému	74
F.2	Definice modelu	75
F.2.1	Rozdělení oblastí	75
F.2.2	Okrajové podmínky	75
F.3	Vytvoření modelu	75
F.3.1	Nastavení problému	75
F.3.2	Vytvoření konstant	75
F.3.3	Vytvoření geometrie	76
F.3.4	Doplnění fyzikálních podmínek	76
F.3.5	Diskretizace a řešení	77
F.4	Diskuze výsledků	77

F.4.1	Rozložení výsledného pole	77
F.4.2	Zobrazení výsledku pomocí grafu	79
G	Model elektromagnetu pomocí Python skriptu	81
G.1	Python skript	81
G.2	Formulace problému	81
G.3	Popis využitých příkazů	82
G.3.1	Vygenerované výsledky	84
G.4	Kompletní script pro model elektromagnetu	87
H	Model feromagnetického aktuátoru pomocí Python skriptu	92
H.1	Python skript	92
H.2	Formulace problému	92
H.3	Popis využitých příkazů	92
H.3.1	Vygenerované výsledky	94
H.3.2	Výpočet síly působící na kotvu	96
H.4	Kompletní script pro model feromagnetického aktuátoru	97

Seznam obrázků

1.1	Rychlost konvergence pro různé druhy adaptivity[5]	4
2.1	Výchozí rozvržení uživatelského prostředí	7
2.2	Horní lišta	9
4.1	3D model cívky (modrá) s částečně vsunutým jádrem (červená)	13
4.2	Nastavení problému	15
4.3	2D model cívky s jádrem (rozměry v cm)	16
4.4	Diskretizovaná definiční oblast	17
4.5	Závislost chyby na počtu kroků hp-adaptivity	18
4.6	Zobrazení siločár	19
4.7	Zobrazení vektorů	19
4.8	Rozložení magnetické indukce	20
4.9	3D skalární rozložení magnetické indukce	20
4.10	Rozložení vektorového magnetického potenciálu	21
4.11	Magnetická indukce v horizontální ose aktuátoru pro $z = 0$ m	22
4.12	Závislost Maxwellovské síly na posunu kotvy vůči cívce	23
A.1	3D model : ocelový válec (červená), cívka (modrá) a podložka (šedivá)	29
A.2	Rozměry modelu (cívka zvýrazněna modře a jádro červeně)	30
A.3	2D model ocelového válce a cívky	32
A.4	Diskretizovaná definiční oblast pro frekvenci 10 Hz	33
A.5	Závislost chyby na počtu stupňů volnosti pro frekvenci 10 Hz	34
A.6	Rozložení magnetické indukce pro frekvenci 10 Hz	34
A.7	Zobrazení siločár a vektorů magnetické indukce pro frekvenci 10 Hz	35
A.8	Zobrazení měrných Jouleových ztrát pro frekvenci 10 Hz	35
A.9	Závislost chyby na počtu kroků hp-adaptivity pro frekvenci 1000 Hz	36
A.10	Zobrazení měrných Jouleových ztrát pro frekvenci 100 Hz	36
B.1	2D model kondenzátoru	38
B.2	2D model kondenzátoru s kruhovou fiktivní hranicí	40
B.3	Závislost chyby na počtu kroků hp-adaptivity	41
B.4	Zobrazení počáteční (světlá) a řešené sítě (tmavá)	42

B.5	Závislost stupňů volnosti na počtu kroků hp-adaptivity	42
B.6	Rozložení ekvipotenciál a vektorů intenzity elektrického pole ve vyšetřované oblasti	43
B.7	Rozložení skalárního elektrického potenciálu ve vyšetřované oblasti	43
B.8	Skalární elektrický potenciál v ose kondenzátoru	44
B.9	Hustota energie v ose kondenzátoru	44
B.10	Intenzita elektromagnetického pole v ose kondenzátoru	45
C.1	Rozměry elektromagnetu v cm	47
C.2	Model elektromagnetu s označením jednotlivých bodů	49
C.3	Řešená a počáteční síť (počáteční světlo barvou)	51
C.4	Závislost stupňů volnosti na počtu kroků hp-adaptivity	51
C.5	Rozložení vektorové magnetického potenciálu pro magnety nesouhlasně	52
C.6	Rozložení vektorové magnetického potenciálu pro magnety souhlasně	52
C.7	Zobrazení siločár pro magnety nesouhlasně	53
C.8	Zobrazení siločár pro magnety souhlasně	53
C.9	Vektory magnetického pole pro magnety nesouhlasně	54
C.10	Vektory magnetického pole pro magnety souhlasně	54
C.11	Magnetická indukce podél zadané přímky v tab. C.7	55
D.1	2D model koaxiálního kabelu	56
D.2	2D model koaxiálního kabelu s body pro vytvoření geometrie	58
D.3	Řešená a počáteční síť	59
D.4	Závislost relativní chyby na počtu stupňů volnosti	60
D.5	Zobrazení ekvipotenciál a vektorů pro proudovou hustotu	60
D.6	Skalární rozložení elektrického potenciálu	61
D.7	Rozložení proudové hustoty v izolaci koaxiálního kabelu	61
D.8	Graf skalárního elektrického potenciálu podél osy r	62
D.9	Graf intenzity elektrického pole podél osy r	63
D.10	Graf proudové hustoty podél osy r	63
E.1	2D model dvouvodičového pásového vedení	64
E.2	2D model dvouvodičového pásového vedení (rozměry v cm)	66
E.3	Diskretizovaná definiční oblast (počáteční síť světlo barvou a řešená síť tmavší barvou)	68
E.4	Závislost stupňů volnosti na počtu kroků hp-adaptivity	68
E.5	Rozložení skalárního pole magnetické indukce	69
E.6	Zobrazení siločár a vektorů	69
E.7	Rozložení skalárního pole vektorového magnetického potenciálu	70
E.8	Zobrazení siločár a vektorů	70
E.9	Vektorový magnetický potenciál pro rozdílný směr proudu tekoucí vodiči	71

E.10	Vektorový magnetický potenciál pro stejný směr proudu tekoucí vodiči . . .	72
E.11	Celková proudová hustota pro stejný směr proudu tekoucí vodiči	73
E.12	Celková proudová hustota pro rozdílný směr proudu tekoucí vodiči	73
F.1	3D model kulového jiskřiště (l není v měřítku)	74
F.2	2D model jiskřiště (rozměry v centimetrech)	76
F.3	Závislost stupňů volnosti na počtu kroků výpočtu	77
F.4	Zobrazení ekvipotenciál a vektorů	78
F.5	Skalární rozložení elektrického potenciálu	78
F.6	Graf intenzity elektrického pole podél osy jiskřiště	79
F.7	Skalární zobrazení intenzity elektrického pole kolem spodní elektrody . . .	80
G.1	Rozměry elektromagnetu v cm	82
G.2	Zobrazení siločár pro magnety uložené souhlasně	85
G.3	Zobrazení siločár pro magnety uložené nesouhlasně	85
G.4	Rozložení skalárního pole magnetické indukce pro magnety uložené souhlasně	86
G.5	Rozložení skalárního pole magnetické indukce pro magnety uložené nesou-	
	hlasně	86
G.6	Rozložení vektorového magnetického potenciálu pro magnety uložené sou-	
	hlasně	86
H.1	3D model cívky (modrá) s částečně vsunutým jádrem (červená)	93
H.2	Vykreslení počáteční sítě	94
H.3	Zobrazení siločár magnetického pole	94
H.4	Zobrazení vektorů magnetického pole	95
H.5	Rozložení skalárního pole magnetické indukce	95
H.6	Velikost síly působící na kotvu získané z energie magnetického pole	96
H.7	Velikost síly působící na kotvu získané z Maxwellovské síly	97

Seznam tabulek

1.1	Druhy řešení pro různá fyzikální pole	5
1.2	Porovnání simulačních programů [1]	6
1.3	Porovnání fyzikálních vlastností simulačních programů [1]	6
2.1	Popis ikon v horní liště	9
4.1	Hodnoty konstant	15
4.2	Souřadnice uzlů feromagnetického aktuátoru	15
4.3	Zadané hodnoty pro různé materiály	17
4.4	Souřadnice geometrie pro vykreslení grafu	21
4.5	Hodnoty Maxwellovské síly působící na kotvu při vsunování do cívky	23
A.1	Parametry problému	31
A.2	Hodnoty konstant	31
A.3	Souřadnice jednotlivých uzlů	32
A.4	Zadané hodnoty pro různé materiály	33
A.5	Hodnoty souřadnic	37
B.1	Nastavení parametrů problému	39
B.2	Hodnoty konstant	40
B.3	Souřadnice jednotlivých uzlů	40
B.4	Zadané hodnoty pro různé materiály	41
C.1	Parametry nastavení problému	48
C.2	Hodnoty konstant	48
C.3	Souřadnice uzlů elektromagnetu	49
C.4	Souřadnice uzlů fiktivní hranice kolem elektromagnetu	49
C.5	Zadané hodnoty pro cívky, vzduch a železo	50
C.6	Zadané hodnoty pro pravý a levý magnet	50
C.7	Souřadnice geometrie pro vykreslení grafu	55
D.1	Parametry nastavení problému koaxiálního kabelu	57
D.2	Hodnoty konstant	58
D.3	Souřadnice jednotlivých uzlů	58

D.4	Hodnoty souřadnic	62
E.1	Parametry nastavení problému	65
E.2	Hodnoty konstant	66
E.3	Souřadnice uzlů dvou vodičového pásového vedení	66
E.4	Zadané hodnoty materiálů pro vodiče a vzduch	67
E.5	Souřadnice geometrie pro vykreslení grafu	71
F.1	Parametry nastavení problému	75
F.2	Hodnoty konstant	76
F.3	Souřadnice jednotlivých uzlů	76
F.4	Hodnoty souřadnic	79
G.1	Parametry nastavení problému	83

Seznam symbolů a zkratek

API	Application Programming Interface. Rozhraní pro programování aplikací.
CAD	Computer Aided Design. Počítačem podpořený návrh.
CSV	Comma Separated Values. Hodnoty oddělené čárkami.
DOFs	Degrees of Freedom. Počet stupňů volnosti.
DXF	Data eXchange Format. Formát výměny dat.
FEM	Finite Element Method. Metoda konečných prvků.
GPL	General Public License. Všeobecná veřejná licence.
OpenGL	Open Graphic Library. Otevřená grafická knihovna.
UMFPACK	Unsymmetric Multifrontal Sparse LU Factorization Package.
VTK	The Visualization Toolkit. Nástrojová sada pro vizualizaci.
XML	Extensible Markup Language. Rozšiřitelný značkovací jazyk.

Úvod

Bakalářská práce je zaměřena na využití simulačního programu Agros2D. Jejím cílem je vytvoření několika ilustrativních příkladů s podrobným popisem postupu modelování úlohy, pro různé druhy fyzikálních polí v kartézských souřadnicích i v radiálních, které jsou využity pro osově symetrický typ problému.

Začíná popisem vlastností programu. Následně je představeno uživatelské prostředí a je vysvětleno, jak se ovládá. Poté je nastíněn princip dvou dimenzionálního modelování. Dále je uveden jeden ilustrativní příklad, ve kterém je velice podrobně vysvětleno, jak se postupuje při vytváření modelu v jednotlivých krocích. V závěru ilustrativního příkladu je popsáno, jaké druhy výsledků simulační program poskytuje a je ukázáno jejich využití pro případné další výpočty. Následuje nastínění využití Python skriptů při práci s programem, které mohou velice zjednodušit opakovanou manipulaci s modelem. Dále jsou zhodnoceny dosažené zkušenosti a navrhnutá případná vylepšení simulačního programu.

Dalších osm ilustrativních příkladů je umístěno v příloze bakalářské práce. Prvních sedm příkladů, včetně příkladu uvedeného v kapitole 4 bakalářské práce, obsahuje popis postupu modelování při využití grafického uživatelského prostředí programu a následnou manipulaci s výsledky opět s využitím grafického uživatelského prostředí. Poslední dva ilustrativní příklady jsou zaměřeny na využití Python skriptů při modelování a při práci s výsledky. Pomocí skriptu je velice snadná opakovaná manipulace s parametry modelu, jako jsou úprava materiálů a hlavně geometrie při opakujících se výpočtech stejného modelu. Skript také dokáže automaticky vygenerovat výsledky a například nám může vykreslit graf závislosti určité veličiny na měnících se parametrech modelu.

1

Vlastnosti simulačního programu

Agros2D je univerzální multiplatformní počítačový program určený pro řešení fyzikálních dvoudimenzionálních polí buďto v kartézském nebo osově symetrickém uspořádání. Je vytvořen v programovacím jazyce C++. Program je vyvíjen na Katedře teoretické elektrotechniky FEL ZČU v Plzni a je distribuován pod GPL licenci.[1]

Simulační program pro řešení příslušných parciálních diferenciálních rovnic využívá knihovnu Hermes2D. Tato knihovna je založena na adaptivní metodě konečných prvků vyššího řádu přesnosti. Je vyvíjena skupinou pod vedením Pavla Šolína, který v roce 2010 působil na univerzitě v Renu v USA a je v této oblasti uznávaným odborníkem. Grafické prostředí je vytvořeno pomocí knihovny Qt a je plně multiplatformní. Díky této knihovně je aplikace také lokalizovaná do českého jazyka. Pro vykreslování geometrie a vypočtených dat je použita knihovna OpenGL. Program je vyvíjen s důrazem na jednoduchost a snadnost použití a přitom na co největší komplexnost.[1]

1.1 Program Agros2D

1.1.1 Hlavní autoři

Hlavní autoři simulačního programu Agros2D jsou:

- Pavel Karban - hlavní vývojář
- František Mach - vývojář, dokumentace
- Pavel Kůs - vývojář
- Lukáš Koudela - vývojář RF modulu
- Bartosz Sawicki - překlad do polštiny
- Václav Kotlan - překlad do němčiny
- Petr Kropík - vývojář

1.1.2 Oficiální webové stránky

Odkazy na oficiální stránky programu:

- Webové - <http://agros2d.org/>
- Facebook - <https://www.facebook.com/pages/Agros2D/132524130149770?sk=info>
- Diskuzní - <http://groups.google.com/group/agros2d/>

1.2 Knihovna Hermes2D

Hermes2D je knihovna pro programovací jazyce C++. Její využití je pro rychlý vývoj adaptivních řešitelů hp-FEM a hp-DG. Tyto nové hp-adaptivní algoritmy pomáhají řešit širokou škálu problémů systémů od obyčejných diferenciálních rovnic a stacionárních lineárních parciálních diferenciálních rovnic až po složité časově nelineární multifyzikální diferenciální parciální rovnice.[2]

1.2.1 Schopnosti knihovny z matematického pohledu

Mezi schopnosti knihovny Hermes2D z matematického pohledu patří:

- Aproximace prostoru až k desátému řádu polynomu.
- Závěsné uzly v libovolném pořadí.
- Multimesh vytvořený přes komponent specifických sítí bez výčnělků nebo interpolací v multifyzikálních problémech.
- Časově adaptivní schopnosti s velkým počtem předem realizovaných metod numerického řešení obyčejných diferenciálních rovnic metodou Runge-Kutta.
- Sdružené problémy řešené společně s využitím hp-DG a hp-FEM.
- Adaptivita založená na základě referenčního řešení a místní projekci.
- Výpočty s fyzikálními veličinami definovanými v různých oblastech.

1.2.2 Schopnosti knihovny z pohledu realizace

Mezi schopnosti knihovny Hermes2D z pohledu realizace patří:

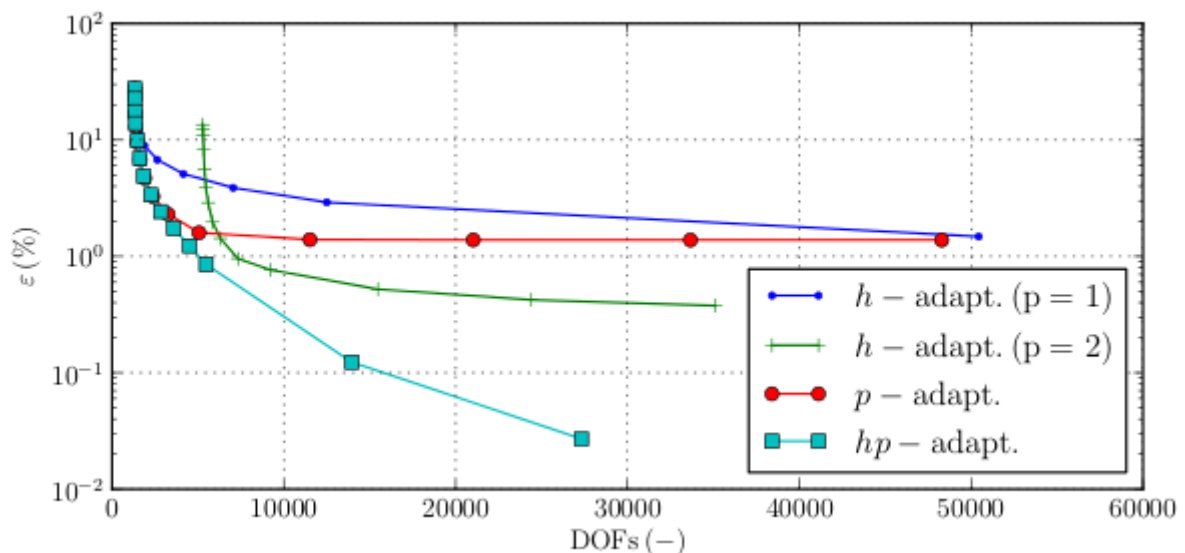
- Možnost multiplatformní paralelní sdílené paměti.
- Jednotná knihovna pro manipulaci s reálnými a komplexními problémy s využitím pokročilých C++ šablon.

- Vlastní vizualizace založená na OpenGL a VTK výstupy sítě, mezer a řešení.
- Oddělení kódu knihovny do samostatných snadno spravovatelných úseků.
- Možnosti uložit a načíst nejdůležitější třídy pomocí XML schémat.
- Přehledná dokumentace generovaná univerzálním nástrojem pro tvorbu dokumentace ze zdrojového kódu C++ (Doxygen).
- Využití výjimek v API.

1.2.3 Metoda hp-FEM

Metoda hp-FEM kombinuje změnu velikosti elementů (zjemnění řešené sítě) a změnu řádu polynomu, kterým je každý element proložen. Těmito změnami je zajištěna exponenciální konvergence výsledků. Pro provádění těchto změn se používá automatická hp-adaptivita, která se provádí na základě největších gradientů řešené veličiny.[3]

Na obr. 1.1 je znázorněn průběh řešení pro různé druhy adaptivity. Vidíme, že nejrychlejší konvergenci k požadované relativní chybě má na obrázku hp-adaptivita. Ale přesto ne pro každý model je nejvhodnější, a proto simulační program umožňuje výběr a nebo deaktivaci adaptivity. Pokud uživatel deaktivuje adaptivitu nabízí mu program volbu řádu polynomu a počtu zjemnění elementů.[5]



Obr. 1.1: Rychlost konvergence pro různé druhy adaptivity[5]

1.3 Řešitelná fyzikální pole

Simulační program zvládá řešení pro několik typů fyzikálních polí a u některých umožňuje volbu druhu analýzy - viz tab. 1.1.

Fyzikální pole	Druh analýzy		
	Ustálený stav	Harmonická analýza	Přechodný děj
Obecné pole	ano	ne	ne
Elektrostatické pole	ano	ne	ne
Magnetické pole	ano	ano	ano
Elektrické proudové pole	ano	ne	ne
Teplotní pole	ano	ne	ano
Strukturální mechanika	ano	ne	ne
Akustické pole	ne	ano	ne

Tab. 1.1: Druhy řešení pro různá fyzikální pole

1.4 Moduly programu

Simulačním program je rozdělen do těchto tří fází: preprocessor, procesor a postprocessor.

1.4.1 Modul preprocessor

První modul obsahuje definice parametrů problému a vytvoření geometrie modelu. Model je vytvořen z uzlů, hran a ze značek oblastí. Definován je pomocí okrajových podmínek a materiálů. Modul preprocessoru umožňuje základní operace pro práci s uzly, hranami a značkami oblastí, jako jsou například geometrické transformace (posun, otočení a zvětšení). Geometrie modelu může být nakreslena předem v CAD programu a do simulačního programu může být importována ve formátu DXF.[1]

1.4.2 Modul procesor

K diskretizaci geometrie řešeného problému je využíván program Triangle založený na Delaunay triangulaci s jeho pomocí lze vytvořit vysoce kvalitní nestrukturované trojúhelníkové sítě. Modul procesor využívá při výpočtu elementy vyššího řádu přesnosti a automatickou hp-adaptivitu (automatické zjemnění sítě a řádu polynomu na elementu dle potřeby). Pro řešení získané soustavy algebraických rovnic je použit UMFPACK z balíku SuiteSparse.[1]

1.4.3 Modul postprocessor

V poslední fázi se jedná o zpracování výsledků do požadovaného tvaru. Například zjištění lokálních veličin ve vyšetřované oblasti nebo zobrazení rozložení výsledného pole pomocí dvoudimenzionálního skalárního zobrazení vybrané veličiny.

1.5 Porovnání s konkurencí

V následujících tabulkách je porovnání programu Agros2D s několika ostatními simulačními programy. Mezi klíčové vlastnosti, které nejsou dostupné u jiných programů, patří díky knihovně Hermes2D automatická hp-adaptivita a také použití Pythonu jako skriptovacího jazyka aplikace. Díky němu získává Agros2D možnost využití obrovského množství knihoven pro vědecké výpočty.[1]

Produkt	ANSYS	Comsol	QuickField	FEMM	Agros2D
Licence	komerční	komerční	komerční	Aladdin	GPL v2
Geometrie	2D/3D	2D/3D	2D	2D	2D
Max. řád polynomu	2	5	1	1	10
Adaptivita	h, p	h	-	-	h, p, hp
Skriptovací jazyk	Ansys	Matlab Script	ActiveX	LUA	Python

Tab. 1.2: Porovnání simulačních programů [1]

Porovnání z hlediska, zda simulační program zvládne řešit jen lineární nebo i nelineární problémy je uvedeno v tab. 1.3, kde BH znamená závislost magnetické indukce na magnetické intenzitě.[1]

Produkt	Fyzikální vlastnosti	
	Lineární	Nelineární
ANSYS	ano	ano
Comsol	ano	ano
QuickField	ano	ano
FEMM	ano	nelineární pouze BH charakteristika
Agros2D + Hermes2D	ano	Hermes2D nelineární

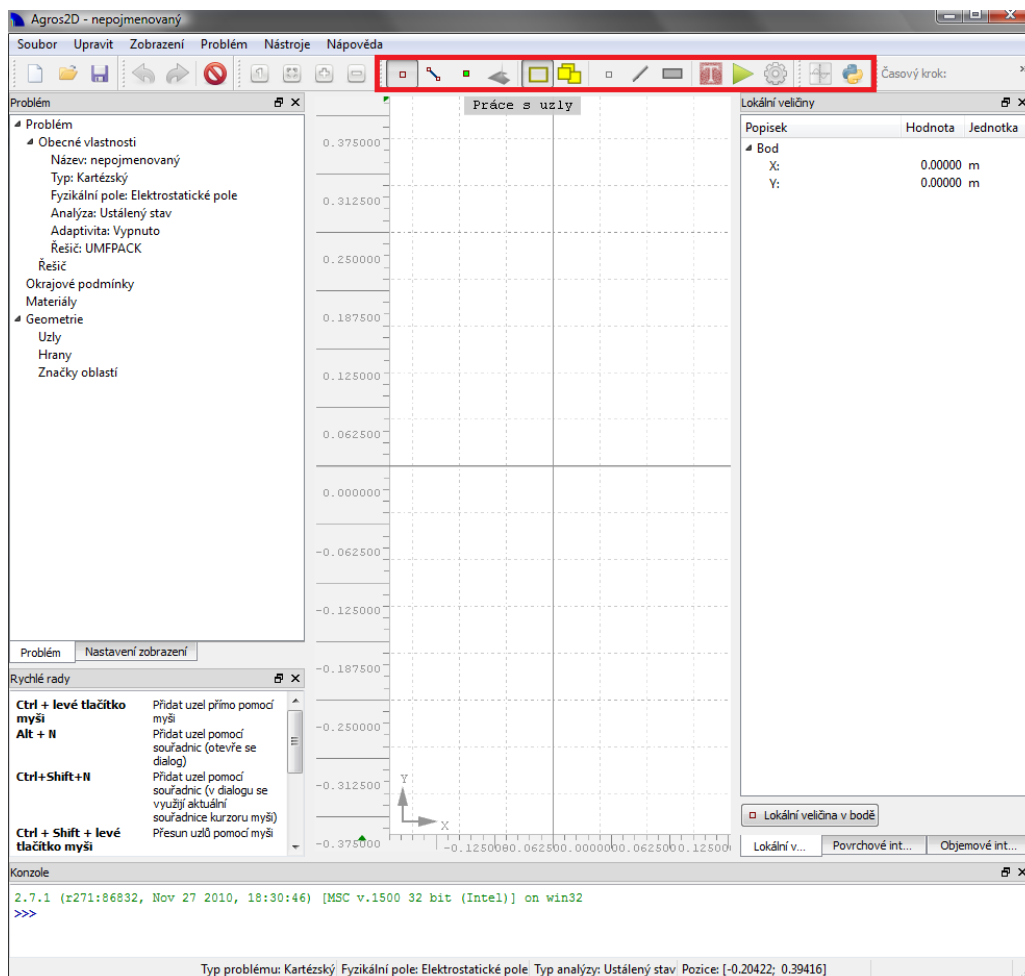
Tab. 1.3: Porovnání fyzikálních vlastností simulačních programů [1]

2

Ovládání uživatelského prostředí

2.1 Základní rozvržení prostředí

Po prvním otevření programu Agros2D, nás uvítá výchozí rozvržení uživatelského prostředí - viz obr. 2.1. Na obrázku je červeným obdélníkem zvýrazněna oblast, která bude nazývána *horní lišta*. Ikony v této oblasti budou při modelování často používány.



Obr. 2.1: Výchozí rozvržení uživatelského prostředí

Prostředí je rozvrženo tak, že v levé části se nachází okno *Nastavení zobrazení* obsahující dvě záložky, které se přepínají vespodu okna.

První je *Problém*, kde jsou uvedeny obecné vlastnosti problému. Dále pak vytvořené okrajové podmínky a materiály. Posledním údajem je geometrie, která obsahuje tři elementy programu uzly, hrany a značky oblastí.

Druhá je *Nastavení zobrazení*, která sama obsahuje tři další záložky. První s názvem *Základní* umožňuje nastavení zobrazení a kromě geometrie je využívána až po vyřešení problému k zobrazení různých druhů výsledků. Druhá záložka *Postprocesor* slouží k nastavení parametrů zobrazovaných výsledků. A nakonec třetí záložka *Pracovní plocha* obsahuje nastavení, které ovládá, jak bude vypadat pracovní plocha a co v ní bude zobrazeno.

Pod tímto oknem je okno *Rychlé rady*, které vždy obsahuje nápovědu a klávesové zkratky pro modelování. To, které rady se zobrazí, závisí na tom, v jakém režimu jsme aktivní. Program je rozdělen do tří režimů při vytváření geometrie. První je *Práce s uzly*, druhým režimem je *Práce s hranami* a nakonec poslední režim se nazývá *Práce se značkami oblasti*. Toto rozdělení je výhodné pro manipulaci s jednotlivými prvky prostředí, aniž bychom přitom ovlivňovali prvky z jiného režimu.

Vpravo se nachází poslední velké okno pro zobrazení vypočtených výsledků, v obr. 2.1 je aktivní první záložka *Lokální veličiny*. V tomto okně se zobrazují v jednotlivých záložkách vypočítané výsledky v bodě, v povrchu a nakonec v objemu.

Uživatelské rozhraní ještě obsahuje v dolní části okno *Konzole*, kam se dají zapisovat příkazy v programovacím jazyku Python a kterým program rozumí a vykoná. Další využití konzole je jako rychlá kalkulačka, například pokud chceme vypočítat veličinu z několika vyřešených hodnot, snadno si ji do konzole přeneseme pomocí schránky a v konzoli již zapíšeme do požadovaného vzorce. Po potvrzení klávesou enter bude zobrazen výsledek, musíme však pamatovat na správnou syntaxi výpočtu, která se řídí syntaxí v programovacím jazyku Python.

Hned vedle horní lišty vlevo jsou umístěny čtyři ikonky, které určují velikost zobrazení modelu v pracovní ploše, která se nachází uprostřed uživatelského rozhraní a je mnohem větší než zobrazená pracovní plocha na obr. 2.1, protože při vytvoření obrázku bylo okno programu co nejvíce zminimalizováno.

Nakonec prostředí ještě obsahuje ikonky pro manipulaci se soubory, horní nabídku s rozbalovacími okny, kde se dá uživatelské prostředí přizpůsobit a také ovládat.

2.2 Horní lišta

Horní lišta, jak bylo zobrazeno na obr. 2.1, obsahuje ovládací ikony programu. Které se dají rozdělit do částí podle toho, v jaké fázi simulace se nacházíme. První je preprocesor a nacházíme se v ní při vytváření geometrie a zadávání parametrů problému. Druhá, která řeší daný model, je procesor. Poslední je postprocesor a slouží k manipulaci s výsledky.



Obr. 2.2: Horní lišta

Na obr. 2.2 jsou jednotlivé ikony v horní liště označeny alfanumerickou značkou a jejich význam spolu s tím, do kterého režimu spadají, je znázorněn v tab. 2.1.

Značka	Název	Režim
1	Práce s uzly	Preprocesor
2	Práce s hranami	Preprocesor
3	Práce se značkami oblastí	Preprocesor
4	Postprocesor	Postprocesor
5	Výběr oblasti	Preprocesor
6	Transformace	Preprocesor
7	Lokální veličiny	Postprocesor
8	Povrchové integrály	Postprocesor
9	Objemové integrály	Postprocesor
A	Diskretizace oblasti	Procesor
B	Vyřešit problém	Procesor
C	Nastavení problému	Preprocesor
D	Graf	Postprocesor
E	Editor skriptů	Preprocesor

Tab. 2.1: Popis ikon v horní liště

Všechny ikony s číselným označením program pouze přepnou do vybraného režimu, kromě ikony šest, která otevře nové okno *Transformace*. V tomto okně si můžeme zvolit posun, rotaci nebo zvětšení (zmenšení) předtím vybraných elementů v režimu *Výběr oblasti*. Pro každou z těchto možností lze vybrat, zda se vybrané elementy transformují nebo se vytvoří jejich kopie zaškrtnutím políčka *Vytvořit kopii*.

A naopak všechny ikony se znakovým označením otvírají vlastní okna. Pro ikonku A se provede diskretizace oblasti a okno se samo uzavře. Ikonka B spustí řešení problému se zadanými parametry, pokud se pracuje bez adaptivity, tak se okno automaticky po vyřešení uzavře. Při použití adaptivity se okno nezavře a obsahuje tři záložky, ve kterých je znázorněn průběh výpočtu pro jednotlivé kroky adaptivity. První je závislost relativní chyby na stoupajícím počtu kroků výpočtu. Druhý graf je závislost stupňů volnosti opět na stoupajícím počtu kroků výpočtu a na posledním je zobrazena závislost relativní chyby na počtu stupňů volnosti.

2.3 Nastavení problému

Ikona C v horní liště (viz obr. 2.2) otevře okno *Nastavení problému*, kde se vybírají parametry modelu a definují proměnné. V tomto okně je i záložka pro popis řešeného modelu. Je zde zobrazena rovnice podle vybraného fyzikálního pole a druhu analýzy, s kterým se mění do požadovaného tvaru. Dále se podle druhu analýzy dá nastavovat frekvence nebo parametry přechodného děje. Pro ustálený stav ani jedna z těchto možností není k dispozici. Můžeme si zvolit typ problému, zda budeme řešit v kartézských souřadnicích nebo v osově symetrických.

Další volba je, zda se použije adaptivita a případně jaká. Pokud zvolíme adaptivitu, tak ještě máme možnost nastavit počet adaptivních kroků a toleranci adaptivity, které když výpočet dosáhne, tak skončí. A nakonec si můžeme zvolit parametry sítě jako je její typ, počet zjemnění a řád polynomu. Volby posledních dvou parametrů se využívá při vypnuté adaptivitě.

V této verzi programu (2.0.2.1173) není možné měnit linearitu ani řešič.

2.4 Graf

Po vyřešení problému můžeme kliknutím na ikonu D v horní liště (viz obr. 2.2) otevřít jeho okno. Zde podle druhu analýzy problému máme na výběr zobrazení vybrané veličiny podél zadané geometrie nebo v zadaném časovém úseku. Například pro problém řešený v ustáleném stavu můžeme zadat souřadnice počátku a konce a úhel. Při nulovém úhlu bude vybraná veličina zobrazena podél této přímky. Pokud okno posuneme, abychom viděli na pracovní plochu, můžeme v ní vidět zobrazenou zadanou přímku a zkontrolovat si tak, zda jsme zadali správné hodnoty souřadnic. Výsledný graf si můžeme uložit jako obrázek a nebo exportovat data grafu do souboru s příponou CSV.

2.5 Editor skriptů

Poslední ikonka E (viz obr. 2.2) nabízí okno s názvem *Python Lab*, které je určeno pro práci se skripty v programovacím jazyce Python. Hlavní část tohoto okna je určena pro samotné psaní kódu skriptu. Vlevo je doplněno oknem zobrazení aktuální složky, pod ním je historie zadaných příkazů. Ještě níže je prohlížeč funkcí, proměnných a tříd. Poslední součástí je konzole, která je umístěna úplně dole.

3

Princip 2D modelování

Princip modelování spočívá v několika krocích.

- 1) Definice modelu
- 2) Nastavení problému
- 3) Vytvoření geometrie
- 4) Doplnění fyzikálních podmínek
- 5) Diskretizaci a řešení

3.1 Definice modelu

Program Agros2D pracuje pouze v dvoudimenzionální geometrii. Proto si musíme nejprve u řešeného modelu zvolit rovinu pro řešení, přitom si můžeme vybrat ze dvou druhů souřadnic. Buď budeme model řešit v kartézských nebo v osově symetrických souřadnicích.

Poté máme tedy k dispozici dvou rozměrný model, ke kterému známe jeho parametry jednotlivých oblastí a typy okrajových podmínek, případně jejich hodnoty.

3.2 Nastavení problému

Jako první v programu nastavíme všechny potřebné parametry problému v okně *Nastavení problému*. Můžeme si zvolit použití různé adaptivity. Zde si také můžeme vytvořit konstanty ze zadaných parametrů modelu pro pozdější použití. Pak nebudeme muset všude zadávat číselné hodnoty, ale jen název konstanty. Také to umožňuje rychlejší změnu hodnot modelu, protože všechny jsou umístěny na jednom místě a my se na ně pouze odkazujeme.

3.3 Vytvoření geometrie

Poté začneme vytvářet geometrii modelu podle zadání. První co musíme vytvořit jsou jednotlivé uzly modelu. Ty v režimu *Práce s uzly* vložíme na zadané souřadnice. Můžeme je vkládat postupně pomocí levého tlačítka myši a stisknuté klávesy Ctrl a nebo pomocí jejich

souřadnic. Okno pro zadání souřadnic se otevře po stisknutí kláves Alt+N. Souřadnice i všechny následující hodnoty vkládáme s desetinnou tečkou. Ostatní uzly můžeme vložit obdobně nebo využijeme možnosti výběru oblasti a následné transformace.

Po vytvoření uzlů se přepneme do režimu *Práce s hranami* a propojíme uzly tak, aby vytvořily potřebnou geometrii modelu. Přidání hrany se provede kliknutím levým tlačítkem myši poblíž jednoho uzlu a poté u druhého uzlu a zároveň při tom přidržení klávesy Ctrl. Pokud hrana není přímka, tak v její nabídce můžeme zadat úhel a hrana se zaoblí. Pokud se hrana zaoblí na druhou stranu než požadujeme, tak v její nabídce musíme ručně prohodit body, ze kterých je tvořena.

Nakonec v režimu *Práce se značkami oblastí* vložíme jednotlivé značky oblastí do vzniklých ploch. Opět pomocí kliknutí levým tlačítkem myši do plochy, kam chceme značku vložit, nebo pomocí souřadnic. Okno pro zadávání souřadnic otevřeme klávesovou zkratkou Alt+L.

3.4 Doplnění fyzikálních podmínek

Nyní máme nakreslený model, ale musíme vzniklým značkám oblastí vytvořit okrajové podmínky, které jim poté přiřadíme. Kliknutím pravým tlačítkem myši a výběrem nové okrajové podmínky nebo klávesovou zkratkou Alt+B se dostaneme do okna, kde nastavíme parametry této nové podmínky a uložíme si ji pod potřebným názvem. Poté označíme kliknutím levým tlačítkem myši všechny hrany, kterým chceme tuto okrajovou podmínku přiřadit a stiskneme mezerník. V okně, které se otevře, vybereme požadovanou okrajovou podmínku. Hrany, které mají přiřazenou okrajovou podmínku, jsou zobrazeny nepřerušovanou čarou.

Obdobně musíme vytvořit materiály. Kliknutím pravým tlačítkem myši a výběrem *Nový materiál* nebo klávesou zkratkou Alt+M, se otevře nabídka, kde opět zapíšeme zadané hodnoty. Poté dvojitým kliknutím na značku oblasti otevřeme nabídku, ve které ji přiřadíme potřebný materiál.

3.5 Diskretizace a řešení

Máme již nakreslený celý model a pokud jsme vše správně zadali, můžeme postoupit k řešení. Při spuštění výpočtu a využití adaptivity se definiční oblast diskretizuje na stále menší a menší elementy. Při dosažení zadané přesnosti je řešení u konce. A následně se zobrazí výsledky.

V následující kapitole je uveden ilustrativní příklad feromagnetického aktuátoru, kde je podrobně na příkladu vysvětleno vytváření celého modelu od zadání až po získání výsledků.

4

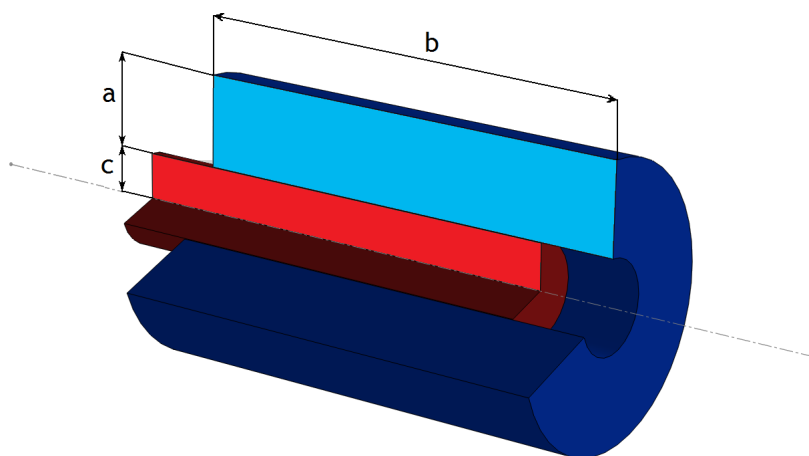
Model feromagnetického aktuátoru

4.1 Formulace problému

Je dán feromagnetický aktuátor, který je tvořen cívkou s ocelovým jádrem, obklopený vzduchem - viz obr. 4.1. Kotva je považována za lineární ocel, která má relativní permeabilitu μ_{r1} , poloměr c a délku b , které je stejná jako u cívky. Cívka i okolní vzduch mají relativní permeabilitu μ_{r0} . Cívka má rozměry: šířka a , délka b . Je tvořena měděným vodičem s průměrem d a N závitů, každým závitem teče proud I_z . Model bude řešen jako 2D osově symetrická úloha, plocha pro řešení je řez na obr. 4.1, který je vyznačen světlejšími barvami.

$$\mu_{r1} = 1500 \quad \mu_{r0} = 1 \quad N = 1000 \text{ závitů} \quad I_z = 38,5 \text{ mA}$$

$$a = 0,1 \text{ m} \quad b = 0,5 \text{ m} \quad c = 0,05 \text{ m} \quad d = 7 \text{ mm}$$



Obr. 4.1: 3D model cívky (modrá) s částečně vsunutým jádrem (červená)

4.2 Definice modelu

4.2.1 Rozdělení oblastí

Máme tři různé oblasti: cívku, kotvu a vzduch. Pro kotvu a vzduch máme již zadané všechny potřebné parametry, pro cívku musíme dopočítat proudovou hustotu J_{ext} podle následujícího vzorce.

$$J_{ext} = \frac{N \cdot I_z}{S \cdot k} = \frac{N \cdot I_z}{a \cdot b \cdot k} = \frac{1000 \cdot 0,0385}{0,1 \cdot 0,5 \cdot 0,77} = 1000 \frac{\text{A}}{\text{m}^2}$$

Koeficient plnění cívky k vychází 77 %.

$$k = \frac{\pi \cdot \left(\frac{d}{2}\right)^2 \cdot N}{a \cdot b} = \frac{\pi \cdot \left(\frac{7}{2}\right)^2 \cdot 1000}{100 \cdot 500} = 0,7697$$

4.2.2 Okrajové podmínky

Budou potřeba okrajové podmínky pro fiktivní hranici nekonečna a pro osu symetrie. Pro obě okrajové podmínky platí, že magnetický vektorový potenciál je

$$\mathbf{A} = 0,$$

protože na ose platí antisymetrie a na fiktivní hranici nekonečna spojitost potenciálu.

4.3 Vytvoření modelu

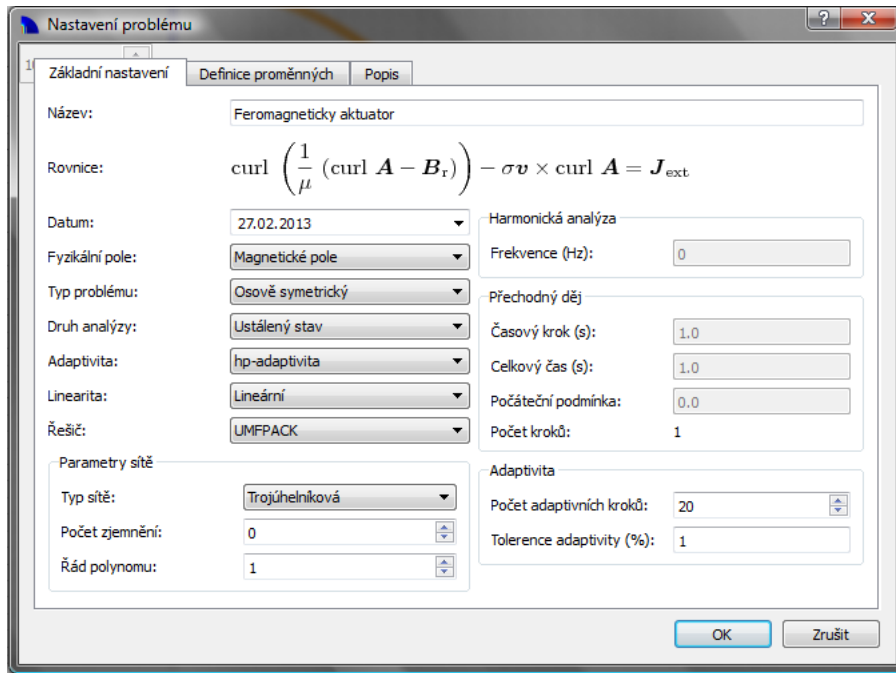
Za prvé budou zadány jednotlivé body, které poté budou spojeny čarami. Dále budou vytvořeny okrajové podmínky, které budou přiřazeny jednotlivým hranám. Nakonec se jednotlivé oblasti rozdělí pomocí značek oblastí, které budou vytvořeny podle zadaných parametrů.

4.3.1 Nastavení problému

Začneme nastavením parametrů problému: klikneme na ikonku ozubeného kolečka nebo v horní nabídce vybereme *Problém* a v seznamu klikneme na *Nastavení problému* nebo můžeme použít klávesovou zkratku *F12*. Objeví se okno s nastavením. Vše nastavíme podle obr. 4.2. Zvolili jsme hp-adaptivitu s 20 kroky a přesností 1 %. To znamená, že program bude postupně zvyšovat počet stupňů volnosti tak, aby relativní chyba byla menší než námi nastavený parametr.

4.3.2 Vytvoření konstant

Vytvoříme si konstanty pro μ_{r1} a J_{ext} tak, že se v okně *Nastavení problému* přepneme do druhé záložky *Definice proměnných* a zapíšeme konstanty podle tab. 4.1.



Obr. 4.2: Nastavení problému

μ_{r1}	1500
J_{ext}	1000

Tab. 4.1: Hodnoty konstant

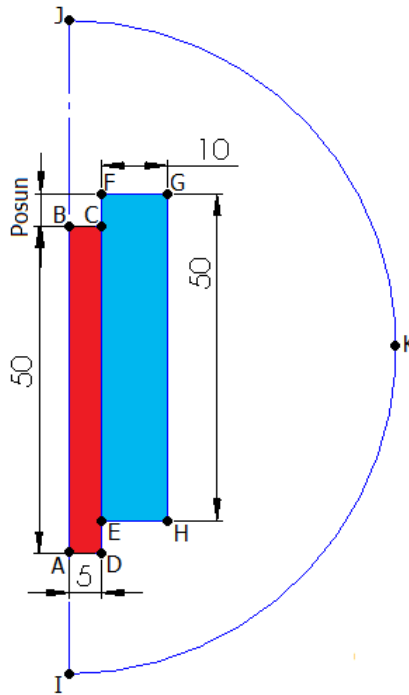
4.3.3 Vytvoření geometrie

Začneme vytvořením jednotlivých uzlů podle zadaných rozměrů tak, že pomocí horní lišty se přepneme do režimu *Práce s uzly*. Dále pravým tlačítkem myši zobrazíme nabídku a zvolíme *Nový uzel* nebo totéž pomocí klávesové zkratky *Alt+N*, kde zadáme příslušné souřadnice v metrech a potvrdíme. Tímto postupem přidáme postupně všechny potřebné uzly podle tab. 4.2 (používáme desetinnou tečku). Pokud se spleteme a chceme uzel odstranit, tak je možné na něj kliknout myší a stisknout klávesu *Delete*, ale musíme být přepnuti do správného režimu *Práce s uzly* (ikona s prázdným čtvercem v horní liště). Případně na uzel můžeme kliknout dvakrát a přesunout na správné místo.

Osa	A	B	C	D	E	F	G	H	I	J	K
R [m]	0	0	0,05	0,05	0,05	0,05	0,15	0,15	0	0	1,5
Z [m]	-0,3	0,2	0,2	-0,3	-0,25	0,25	0,25	-0,25	-1,5	1,5	0

Tab. 4.2: Souřadnice uzlů feromagnetického aktuátoru

Poté se přepneme pomocí horní lišty do režimu *Práce s hranami* a uzly propojíme tak, aby vytvořily potřebný model - viz obr. 4.3. Přidržíme klávesu *Ctrl* a klikneme v blízkosti jednoho z uzlů, které chceme spojit hranou. Nepouštíme klávesu a myší přejedeme do blízkosti druhého uzlu a znovu klikneme. Tím jsme vytvořili potřebnou hranu. Proces



Obr. 4.3: 2D model cívky s jádrem (rozměry v cm)

opakujeme pro všechny ostatní hrany. Smazat nebo změnit vytvořenou hranu můžeme obdobně jako při práci s uzly. Hrany AB BC CE ED DA tvoří obrys kotvy, hrany EC CF FG GH HE tvoří obrys cívky a nakonec hrany IA AB BJ tvoří osu symetrie a hrany JK KI fiktivní hranici nekonečna. Pouze hrany JK a KI musíme ještě zaoblit, aby vytvořili půlkruh, to provedeme tak, že jim nastavíme úhel 90 °. Pokud se hrana natočí na druhou stranu, tak ještě musíme v její nabídce prohodit body, které tvoří hranu.

Nakonec musíme pro jednotlivé oblasti vytvořit značky, abychom mohli rozdělit oblasti s rozdílnými vlastnostmi. Opět přidržíme klávesu *Ctrl* a klikneme do jednotlivých oblastí. Tím vytvoříme značky pro tři různé materiály.

4.3.4 Doplnění fyzikálních podmínek

Nyní vytvoříme okrajovou podmínku kliknutím pravým tlačítkem myši a vybereme *Nová okrajová podmínka* nebo klávesovou zkratkou *Alt+B*, nazveme ji *Antisymetrie* a necháme vektorový magnetický potenciál $\mathbf{A} = 0$.

Materiál vytvoříme pomocí pravého tlačítka myši a výběrem *Nový materiál* nebo klávesovou zkratkou *Alt+M*. Vytvoříme tři různé materiály. Jednotlivým materiálům nastavíme parametry podle tab. 4.3 a můžeme využít vytvořených konstant.

V levém okně máme vytvořenou okrajovou podmínku i materiály. Nyní je musíme správně přiřadit. Přepneme se do práce se značkami oblasti a pomocí dvojkliku levým tlačítkem myši na značku oblasti otevřeme nabídku *Značka oblasti*, kde poté vybereme ze seznamu správný materiál. Po vybrání se nad značkou oblasti zobrazí název vybraného materiálu. Proces opakujeme pro ostatní dva materiály. Dále se přepneme do režimu

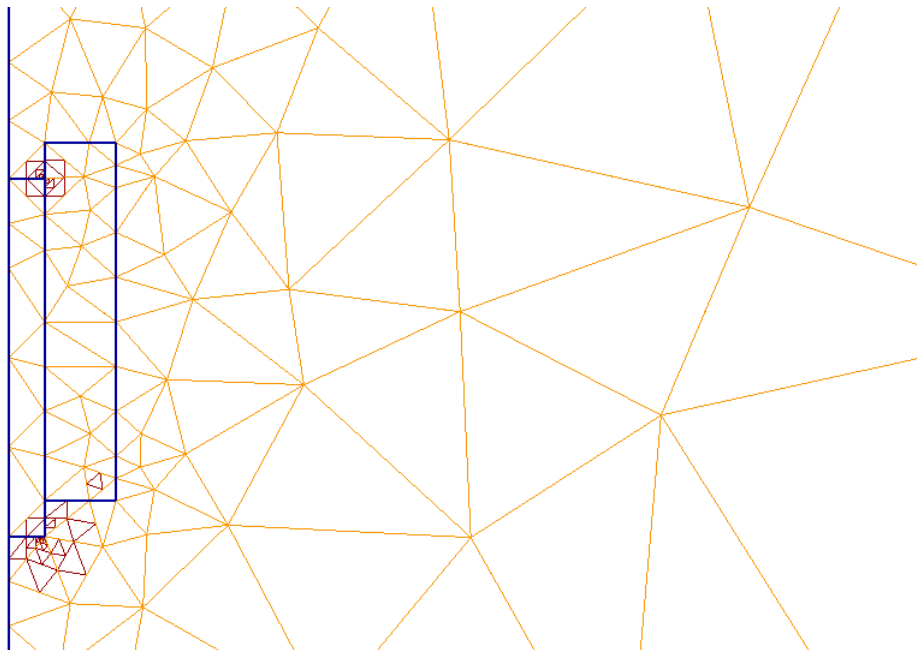
	Relativní permeabilita μ_r [-]	Proudová hustota J_{ext} [$\frac{A}{m^2}$]
Vzduch	1	0
Cívka	1	1000 nebo J_{ext}
Kotva	1500 nebo μ_{r1}	0

Tab. 4.3: Zadané hodnoty pro různé materiály

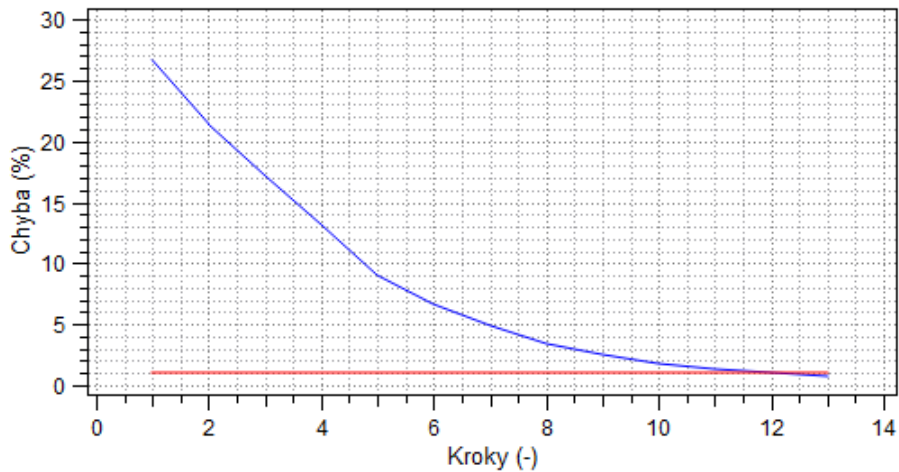
práce s hranami, kde označíme pomocí levého tlačítka myši hrany IA AB BJ JK KI a pravým tlačítkem myši zobrazíme nabídku, kde vybereme *Vlastnosti objektu* nebo můžeme zmáčknout mezerník a přiřadíme jim okrajovou podmínku *Antisymetrie*. Hrany s přiřazenou okrajovou podmínkou jsou zobrazeny nepřerušovanou čarou.

4.3.5 Diskretizace a řešení

Definiční oblast se pro řešení musí rozdělit na jednotlivé elementy. My jsme si vybrali trojúhelníkovou síť, a proto po kliknutí na ikonku se znakem T (*Diskretizace oblasti*) vidíme, jak se oblast rozdělí na trojúhelníky, které jsou na obr. 4.4 vyznačeny světlou barvou. Protože budeme řešit pomocí hp-adaptivity, tak se síť bude v průběhu výpočtu dále zjemňovat až k výsledné síti, která je zobrazena tmavou barvou. Výpočet spustíme kliknutím na ikonku se zeleným trojúhelníkem (*Vyřešit problém*). Po 12 krocích chyba klesla již pod 1 %, které jsme požadovali v zadání a řešení je u konce. V dalších záložkách máme grafy znázorňující závislost chyb a volností v různých krocích výpočtu. Například rychlost, s jakou klesá chyba v závislosti na rostoucím počtu kroků hp-adaptivity, můžeme vidět v záložce *Adapt.chyba* - viz obr. 4.5.



Obr. 4.4: Diskretizovaná definiční oblast



Obr. 4.5: Závislost chyby na počtu kroků hp-adaptivity

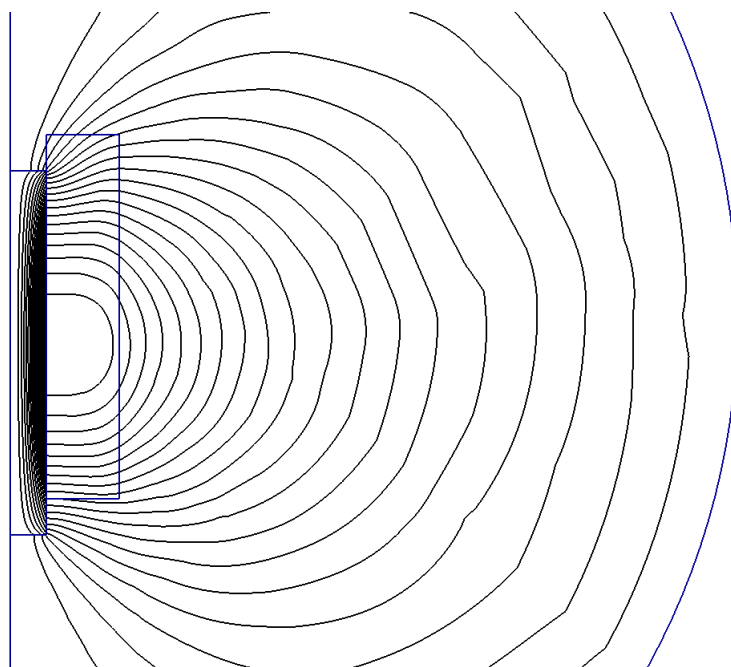
4.4 Diskuze výsledků

Výsledkem může být zobrazení výsledného pole, vykreslení veličiny do grafu (po zadané přímce nebo v čase) nebo mnoho různých veličin, které nám program spočítá v bodě, v přímce nebo v ploše.

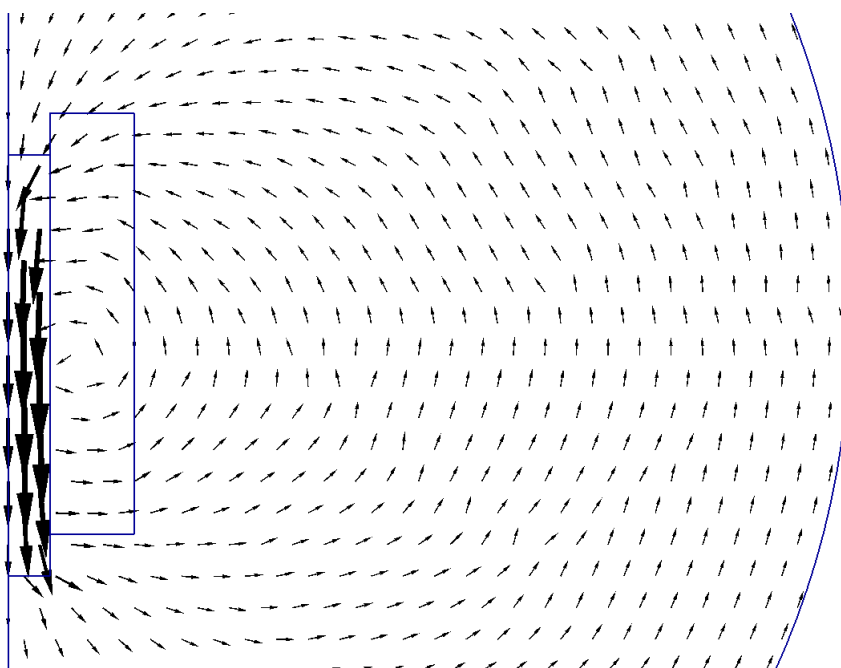
Vpravo se nachází tabulka s výsledky. V aktivní záložce *Lokální veličiny* můžeme kliknout kamkoliv do řešené plochy a zobrazí se výsledky pro konkrétní bod nebo můžeme kliknout na *Lokální veličina v bodě* pod tabulkou a zadat přesné souřadnice bodu, který nás zajímá. Další možností jsou *Povrchové integrály*. Pro práci s nimi se musíme přepnout do správného režimu v horní liště pomocí ikonky se zelenou kolmou čarou. Poté hranu vybereme kliknutím poblíž a vypočte se výsledek. Můžeme označit i více hran najednou postupným označením jednotlivých hran. Poslední možností jsou *Objemové integrály*, opět se přepneme pomocí ikonky se zeleným obdélníkem do správného režimu a můžeme označit jednotlivé plochy, pro které program vypočte energii magnetického pole.

4.4.1 Rozložení výsledného pole

Po výpočtu se vlevo zobrazilo okno *Nastavení zobrazení*, kde si můžeme vybrat, co program vykreslí. Například pro zobrazení rozložení siločár zaškrtneme v nabídce *Kontury*, 2D nabídku přepneme do *Bez zobrazení*, aby nás nerušilo barevné pozadí. Dále v záložce *Postprocesor* vybereme *Kontury a vektory*, kde můžeme nastavit počet a šířku zobrazených kontur, potvrzením se zobrazí požadovaný výsledek. Další příklad je zobrazení vektorů magnetické indukce, pro něj zaškrtneme *Vektory* a opět si v další záložce můžeme zvolit počet a měřítko vektorů. V poslední záložce *Pracovní plocha* můžeme zrušit zobrazování mřížky, os a pravítka, pro zobrazení bez rušivých elementů. Pro uložení obrázku klikneme v horní nabídce na *Soubor* a dále vybereme *Import/Export* a nakonec klikneme na *Export obrázku*. V obr. 4.6 zvoleno 20 kontur s šířkou 1,50 a v obr. 4.7 vybráno 80 vektorů v měřítku 2,00.



Obr. 4.6: Zobrazení siločár



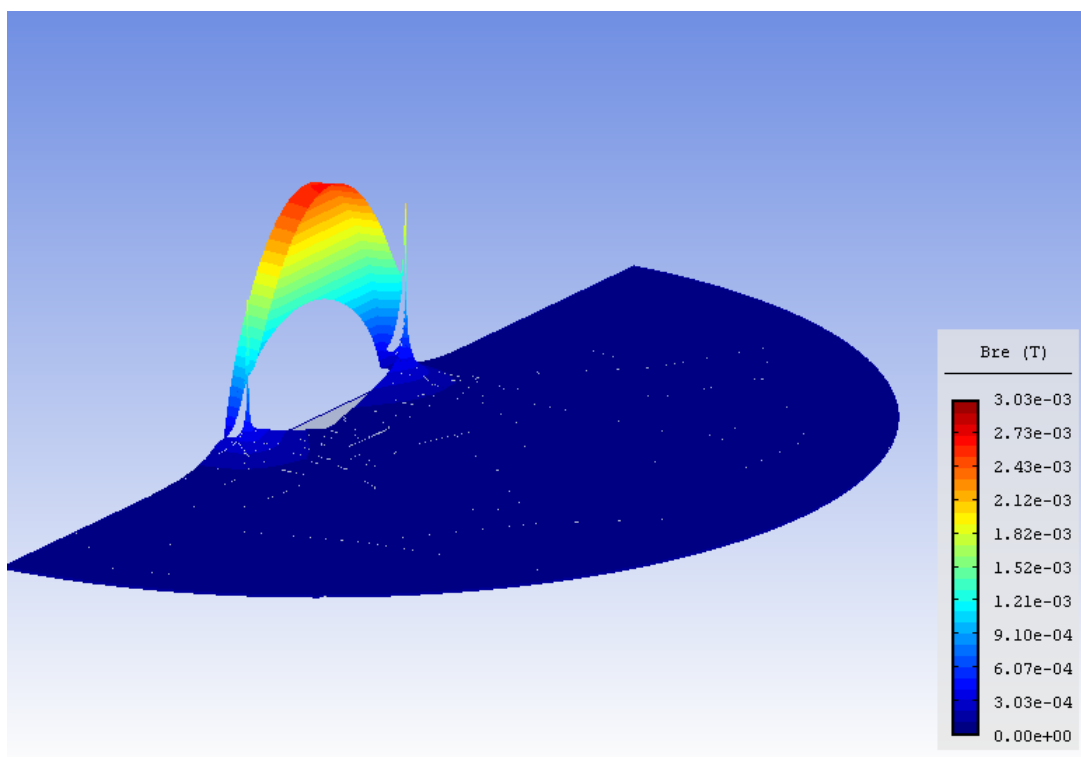
Obr. 4.7: Zobrazení vektorů

Zajímavější je zobrazení skalárního pole vybrané veličiny, pro které si necháme zaškrtnuté políčko *Autom. rozsah*, aby program sám zvolil vhodný rozsah veličiny a její barevné zobrazení spolu s barevnou legendou. Další nastavení barevné škály nalezneme opět v záložce *Postprocesor* pod *Skalárním zobrazením*, kde si můžeme zvolit například počet desetinných míst a počet kroků barvy pokud nepoužijeme filtr. Nyní jsme vybrali *2D Skalární zobrazení* a veličinu *Magnetickou indukci*, výsledek je zobrazen na obr. 4.8, kde je vidět, že se vše indukuje do plochy kotvy a do okolního vzduchu prakticky nic.



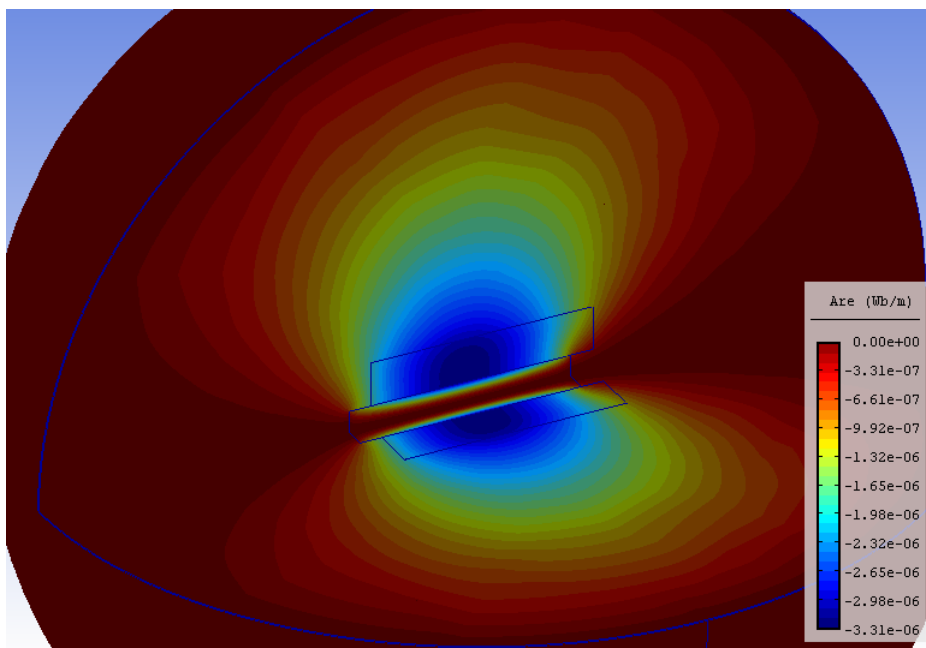
Obr. 4.8: Rozložení magnetické indukce

Stejnou veličinu si můžeme nechat vykreslit pomocí *3D Skalární zobrazení* a dostaneme prakticky stejný výsledek s tím rozdílem, že nyní budou hodnoty magnetické indukce na nové ose, která je kolmá na definiční oblast, viz obr. 4.9, s obrázkem si pomocí stisknutého levého tlačítka myši můžeme libovolně otáčet. Ve 3D zobrazení není možné vykreslit síť, kontury, vektory a pohyby částic a naopak nelze nezobrazit geometrii.



Obr. 4.9: 3D skalární rozložení magnetické indukce

Přestože program pracuje ve 2D, můžeme si nechat vykreslit veličiny jako *3D Objemové skalární pole*, například zobrazení vektorového magnetického potenciálu vidíme na obr. 4.10.



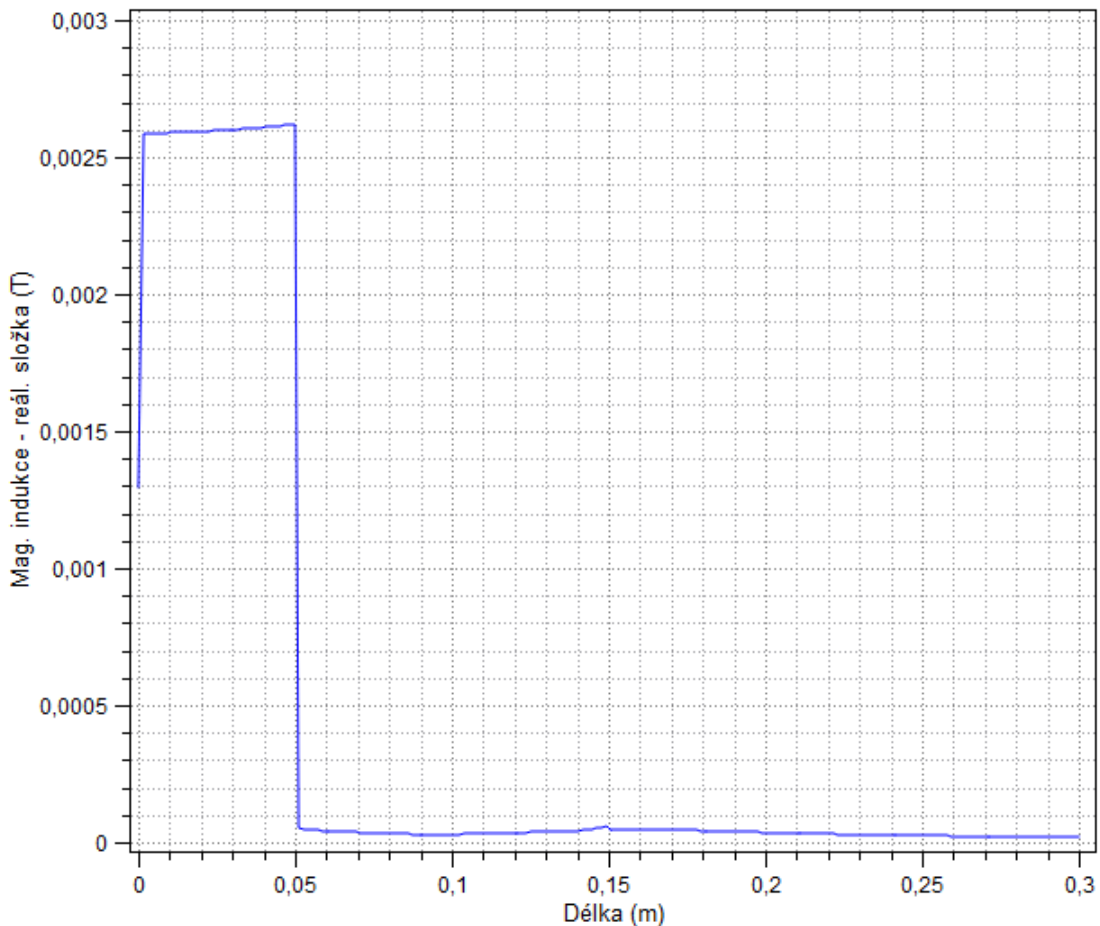
Obr. 4.10: Rozložení vektorového magnetického potenciálu

4.4.2 Zobrazení výsledků do grafu

Další možnost zobrazení výsledků je pomocí grafu, respektive tabulky výsledků. Pro zobrazení grafu klikneme na ikonku, která vypadá jako graf funkce sinus v horní liště a otevře se nám nabídka *Graf*. Zde si můžeme vybrat různé veličiny a případně jejich složky v závislosti na geometrii nebo na průběhu časového intervalu, ale protože řešíme ustálený stav je možnost času vypnuta. V geometrii zadáme pomocí souřadnic přímkou, pro kterou chceme vykreslit graf a pomocí veličiny si vybereme tu, kterou chceme zobrazit. Například pro magnetickou indukci v horizontální ose aktuátoru zadáme souřadnice podle tab. 4.4 a po vykreslení vidíme, že se do kotvy indukce magnetická indukce $B = 2,5 \text{ mT}$ - viz obr. 4.11 a do okolního vzduchu zanedbatelné hodnoty. Dále si výsledné hodnoty můžeme nechat exportovat jako tabulku výsledků se syntaxí pro vykreslení grafu, například v Matlabu nebo Mathematice.

	Počátek	Konec
R[m]	0	0,3
Z[m]	0	0

Tab. 4.4: Souřadnice geometrie pro vykreslení grafu



Obr. 4.11: Magnetická indukce v horizontální ose aktuátoru pro $z = 0$ m

4.4.3 Výpočet síly působící na kotvu

Abychom zjistili, jakou silou působí cívka na kotvu, musíme označit všechny hrany, které tvoří kotvu. Výslednou Maxwellovskou sílu nám program vypočítá a zobrazí v tabulce vpravo pod záložkou *Povrchové integrály*. Výsledná síla vyšla dle předpokladu, kotva je vtahována cívkou silou $F_{z(0,05)}$ při posunu kotvy o 5 cm viz obr. 4.3.

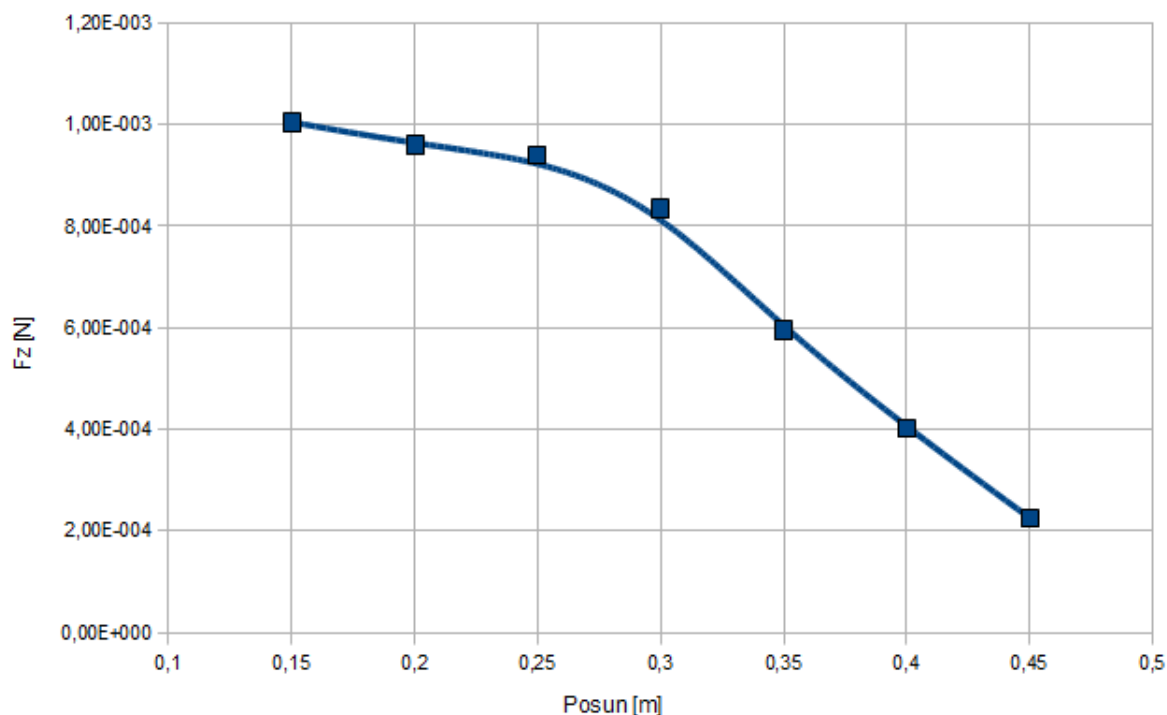
$$F_{z(0,05)} = 6,208 \cdot 10^{-4} \text{ N}$$

Nyní se vrátíme do režimu *Práce s uzly* a posuneme kotvu o 40 cm dolů. Využijeme možnosti, že můžeme přesunout více bodů současně, pomocí režimu *Výběr oblasti* označíme všechny body, které tvoří kotvu a poté v režimu *Transformace* využijeme funkci posunu ve směru osy z o $-0,4$ m, tím se posunou body kotvy a jeden bod cívky, který jsme museli označit s kotvou, ten jednoduše vrátíme na původní pozici změnou jeho souřadnic. Pokud se nám dostala značka oblasti kotvy mimo plochu kotvy, přesuneme ji zpět do plochy kotvy. Nyní postoupíme opět k řešení problému, označení hran kotvy a odečtení nové hodnoty Maxwellovské síly, která nyní vyšla $F_{z(0,45)} = 2,265 \cdot 10^{-4}$ N a je dle očekávání menší. Postup opakujeme tak, že nyní kotvu postupně vsunujeme do cívky (s každým krokem posun ve směru z o $0,05$ m). Výsledné hodnoty jsou zapsány v tab. 4.5 a vykres-

leny do grafu v obr. 4.12, z hodnot výsledků síly je vidět, že když je kotva více vsunuta, působí na ní větší síla.

Posun [m]	0,45	0,4	0,35	0,3	0,25	0,2	0,15
F_z [10^{-4} N]	2,265	4,029	5,959	8,343	9,400	9,603	10,05

Tab. 4.5: Hodnoty Maxwellovské síly působící na kotvu při vsunování do cívky



Obr. 4.12: Závislost Maxwellovské síly na posunu kotvy vůči cívkce

Síla působící na kotvu se dá také spočítat z energie magnetického pole, kde odečteme dvě hodnoty W_m pro různé vsunutí kotvy a vydělíme je rozdílem jejich posunutí. Tento výpočet byl proveden pomocí skriptu v příkladu, který je umístěn v příloze H, konkrétně v jeho části H.3.2.

5

Python skript

Modelování pomocí skriptů je náročné na znalost programovacího jazyka a používaných příkazů s klíčovými slovy. Ale pro zkušeného uživatele umožňuje vytvářet skripty, které samy po spuštění vyřeší celý problém a získají požadované výsledky podle zadání v kódu skriptu.

5.1 Příkazy a klíčová slova

Při programování skriptu jsou velice důležité příkazy programu a klíčová slova, která jsou uvedena v nápovědě. V nápovědě je nalezneme pod *Scripting / Command*, kde je uveden seznam všech dostupných příkazů a v části *Scripting / Keyword_list* je seznam všech klíčových slov. Příkazy jsou použity pro nastavování parametrů problému, pro vytváření geometrie, získávání výsledků a spoustu dalších. Klíčová slova jsou slova, která mají speciální význam v simulačním programu a s jejich využitím v příkazech se program dozví, co od něj požadujeme. Jako příklad uvedu označení energie magnetického pole klíčovým slovem *Wm*. Pokud toho klíčové slovo vložíme do příkazu pro získání výsledků v ploše vyřešené definiční oblasti, získáme hodnotu energie magnetického pole. Takže vidíme, že kombinací klíčových slov a příkazů můžeme sestavit skript, který bude ovládat simulační program.

5.2 Ostatní příkazy

Při psaní kódu skriptu se nemusíme omezovat jen na vybrané příkazy a klíčová slova z nápovědy programu. Ale můžeme v něm pracovat také s ostatními běžnými příkazy v syntaxi programovacího jazyka Python. A jejich kombinací můžeme napsat skript, který ne jenom že získá požadované hodnoty, ale může je dále zpracovat a například zobrazit do grafu.

6

Zkušenosti a vylepšení programu

6.1 Dosažené zkušenosti

Ze začátku používání programu Agros2D jsem využíval jen grafické uživatelské prostředí, které není příliš těžké na ovládání. O programovacím jazyku Python a využití jeho skriptů jsem jen slyšel. Při kreslení modelů jsem narazil na několik zádrhelů, které mě chvíli zpomalovaly. Například kreslení kružnice, kde se musí hrany správným směrem zaoblit. Než jsem se tedy naučil správně používat uživatelské prostředí tak, jak funguje a ne tak, jak bych si představoval, vytváření modelů mi trvalo delší dobu. Několik návrhů na zlepšení jsem navrhl v následující části 6.2. Dosažené zkušenosti při vytváření geometrie modelu mohou ohodnotit tak, že již velice rychle a s využitím usnadňujících funkcí umím nakreslit jakoukoliv geometrii modelu i s využitím symetrií. Usnadňujícími funkcemi myslím vytvoření proměnných pro zadávání parametrů a využití transformace pro vytváření geometrie.

Další zkušenosti jsem získal při opakovaném nastavování parametrů a adaptivity pro různé modely. Při práci s výsledky vím, kde naleznu požadované nastavení pro zobrazení výsledků, které potřebuji a pro úpravu zobrazovaných rozložení polí. Dokážu efektivně využít konzoli pro výpočty při využití proměnných a vypočtených výsledků.

Ale určitě největší zkušenosti jsem získal v oblasti Python skriptů. Které jsou velice užitečné pro změnu parametrů modelu a nebo například pro posouvání geometrie modelu. S využitím skriptu je velice pohodlná práce s výsledky, protože si například nastavíme v kódu skriptu jaké veličiny chceme vykreslit, při jakém přiblížení plochy a dalším příkazem si je uložíme do adresáře pod zadaným názvem. A pro opakované výpočty se záměnou parametrů přichází největší výhoda skriptu, že již znovu nemusíme zobrazovat a ukládat stejné výsledky. Dále jsem se naučil, jak získat hodnoty pomocí skriptu a jak s nimi dále pracovat a nakonec je zobrazit například do grafu.

6.2 Návrh na vylepšení programu

Myslím si, že jedna z asi nejdůležitějších vlastností, kterou by programy měly obsahovat je automatické ukládání. Protože se mi při kreslení a řešení ilustrativních příkladů mnohokrát stalo, že jsem celý model nakreslil a nadefinoval a po spuštění výpočtu program spadl. Protože jsem však zapomněl před kliknutím na vyřešit problém úlohu uložit, o celý model jsem přišel a musel jsem začít od znova. A ani po této zkušenosti jsem další úlohy často neukládal a spoléhal se, že je uložím až vyřešené. Proto si myslím, že aspoň před spuštěním řešení a následně ihned po něm, by měl program model automaticky uložit.

Dále jsem při kreslení modelu narazil na to, že při zadávání uzlů pomocí okna se souřadnicemi, se okno vždy po potvrzení zavře a pro další bod se musí opět otevřít. Bylo by dobré, kdyby okno zůstalo po přidání bodu otevřené a umožnilo hned přidávání dalších bodů. Uspadnilo by to práci při zadávání více bodů, které nemohou být snadno vytvořeny s využitím transformace, kde toto neuzavírání okna funguje.

Při kreslení modelu pomocí uživatelského prostředí programu bez využití skriptu se pro nezkušeného uživatele často stane, když potřebuje nakreslit zaoblenou hranu, že zadá do nastavení hrany požadovaný úhel, ale po potvrzení se hrana natočí na druhou stranu než bylo uživatelem požadováno. Poté se musí ručně prohodit v nastavení hrany jednotlivé body. Pro tyto začínající uživatele, kteří dopředu nevědí, v jakém směru hodinových ručiček spojit hranu tak, aby se zalomila, jak budou požadovat, by bylo vhodné po zadání úhlu a potvrzení okna zjednodušit přehození hran. Například pomocí ikonky, která by se zobrazila po potvrzení zadání úhlu na několik vteřin. Samozřejmě pro zkušené uživatele se jedná o naprosto banální věc, protože program má připravený uživatelský skript pro zadávání kružnic.

Když se zadávají veškeré souřadnice elementů geometrie, tak při použití desetinné čárky program nenahlásí chybu, ale zadanou hodnotu ignoruje a místo ní vloží nulu. Toho se uživatel nemusí všimnout a pak dlouho hledá chybu. Nejlepší řešení by bylo, kdyby program uměl automaticky nahradit desetinnou čárku za tečku. A nebo aspoň hlásil chybu tak, jako to dělá při pokusu o zadání neexistující proměnné.

Pokud vytvoříme značku oblasti nebo hranu a otevřeme její nabídku a snažíme se jí přidělit materiál či okrajovou podmínku, kterou jsem ještě nevytvořili, není v této nabídce možnost vytvořit ji právě teď. Přitom ikonka pro upravování již existující a vybrané okrajové podmínky v okně existuje jen je pro přiřazenou žádnou podmínku neaktivní. Navrhoval bych, využít například této ikonky a při nepřipravené žádné podmínce a otevření okna této hrany, by tato ikonka umožnila vytvoření nové okrajové podmínky či materiálu pro značky oblastí.

Program počítá veličiny na hloubku jednoho metru. Bylo by dobré, kdyby šlo zadat tuto hloubku určitou veličinou.

Závěr

Vytvořil jsem celkem devět ilustrativních příkladů pro tři typy fyzikálních polí. Prvních sedm příkladů jsem vytvořil s využitím grafického uživatelského rozhraní simulačního programu Agros2D. A poslední dva jsem vytvořil pomocí skriptu napsaného v programovacím jazyku Python. Tyto dva příklady mají stejné zadání, jako dva z předchozích příkladů, kvůli názornému porovnání náročnosti jednotlivých metod použitých při vytváření modelu a získávání výsledků.

Využití Python skriptu je vhodné pro opakující se výpočty se změnami parametrů nebo se změnami geometrie modelu. První možnost je ukázána v prvním příkladu na využití skriptu v příloze G, kde je také podrobně předvedeno, jak se pracuje s příkazy a klíčovými slovy z nápovědy programu. Protože v nápovědě je jen pár příkladů na použití příkazů a klíčových slov. Pro nového uživatele je velice těžké, bez předchozí dobré znalosti jazyka Python, rychle pochopit jejich správnou syntaxi. Druhá možnost postupné změny geometrie je nastíněna v druhém příkladu na využití skriptu v příloze H. Při porovnání náročnosti na ovládání při posunu kotvy aktuátoru pomocí grafického prostředí v kapitole 4 a pomocí skriptu v příloze H, je názorně vidět síla využití Python skriptu a pohodlnost získávání výsledků.

Při modelování ilustrativních příkladů jsem využíval hp-adaptivitu s odhadovaným počtem kroků a s přesností relativní chyby jednoho procenta. Pro většinu příkladů byla konvergence řešení velice rychlá, pouze pro vyšší kmitočty u harmonického magnetického pole nedosáhla při zadaném počtu kroků požadované přesnosti.

Literatura

- [1] *Abclinuxu* [Cit. 25. 5. 2013]. Dostupné z: <http://www.abclinuxu.cz/clanky/agros2d-aplikace-pro-reseni-fyzikalnich-poli>
- [2] *Hermes2D* [Cit. 28. 5. 2013]. Dostupné z: <http://hpfem.org/hermes/#Overview>
- [3] *Abclinuxu* [Cit. 25. 5. 2013]. Dostupné z: <http://www.abclinuxu.cz/clanky/agros2d-vypocet-kapacity-jiskriste-a-indukcnosti-dvojlinky>
- [4] *Abclinuxu* [Cit. 25. 5. 2013]. Dostupné z: <http://www.abclinuxu.cz/clanky/agros2d-reseni-prestupu-tepla-ve-zdivu>
- [5] *hp-adaptivita* [Cit. 29. 5. 2013]. Dostupné z: <http://www.agros2d.org/automatic-hp-adaptivity/>
- [6] Mayer, D.: *Aplikovaný elektromagnetismus*. České Budějovic: KOPP, 2012. ISBN 978-80-7232-424-8
- [7] Benešová, Z., Mayer, D.: *Základní příklady z teorie elektromagnetického pole*. Plzeň: TYPOS, 2008. ISBN 978-80-7043-737-7

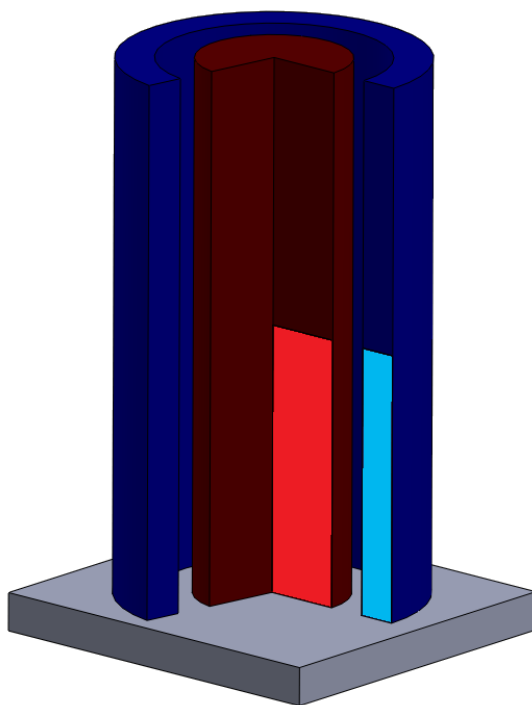
Příloha A

Indukční ohřev ocelového válce

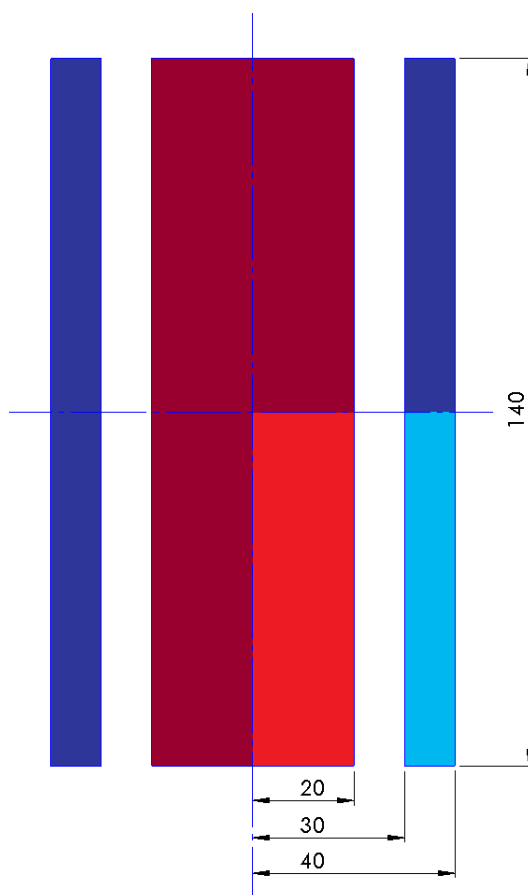
A.1 Formulace problému

Je dán ocelový válec, který je postaven na keramické podložce a umístěn ve středu cívky, celá soustava je obklopena vzduchem - viz obr. A.1. Ocelový válec má relativní permeabilitu μ_{r1} a elektrickou vodivost γ_1 . Cívka i okolní vzduch mají relativní permeabilitu μ_{r2} . Cívka je tvořena tenkým měděným vodičem s proudovou hustotou J_{ext} . Model bude řešen jako 2D osově symetrická úloha s využitím symetrie, plocha pro řešení je na obr. A.1 vyznačena světlejšími barvami. Rozměry cívky a válce jsou uvedeny na obr. A.2.

$$\mu_{r1} = 1000 \quad \gamma_1 = 4 \cdot 10^6 \frac{\text{S}}{\text{m}} \quad \mu_{r2} = 1 \quad J_{ext} = 3 \cdot 10^7 \frac{\text{A}}{\text{m}^2}$$



Obr. A.1: 3D model : ocelový válec (červená), cívka (modrá) a podložka (šedivá)



Obr. A.2: Rozměry modelu (cívka zvýrazněna modře a jádro červeně)

Využijeme symetrie v obou osách a budeme modelovat jen čtvrtinu celkového modelu. Plocha, kterou budeme řešit, je zobrazena světlejšími barvami na obr. A.2.

A.2 Definice modelu

A.2.1 Rozdělení oblastí

Máme čtyři různé oblasti: cívku, jádro, podložku a vzduch. Keramická podložka je vyrobena ze šamotu a její vlastnosti jsou stejné jako u vzduchu, proto jí do modelu nebudeme kreslit. Využijeme symetrie modelu jak v ose r , tak i v ose z a náš řešený model bude jen čtvrtina celkového 3D modelu, viz světlejší barvy na obr. A.1.

A.2.2 Okrajové podmínky

Budou potřeba dvě okrajové podmínky. Pro fiktivní hranici nekonečna a pro osu symetrie podle osy z platí, že magnetický vektorový potenciál je

$$\mathbf{A} = 0,$$

protože na ose z platí antisymetrie a na fiktivní hranici nekonečna spojitost potenciálu. Poslední podmínka je dána symetrií (podle osy r) a tím, že siločáry do této plochy musejí vstupovat kolmo a proto je povrchová proudová hustota rovna nule.

$$K_0 = 0$$

A.3 Vytvoření modelu

A.3.1 Nastavení problému

Začneme nastavením parametrů problému, klikneme na ikonku ozubeného kolečka v horní liště a vše nastavíme podle tab. A.1. Zvolili jsme harmonickou analýzu, a proto musíme zadat frekvenci, začneme na frekvenci 10 Hz, kterou pro další výpočty budeme měnit.

Typ problému	Osově symetrický
Fyzikální pole	Magnetické pole
Druh analýzy	Harmonická analýza
Frekvence [Hz]	10
Typ sítě	Trojúhelníková
Počet zjemnění	0
Řád polynomu	1
Typ adaptivity	hp-adaptivita
Počet adaptivních kroků	30
Tolerance adaptivity [%]	1

Tab. A.1: Parametry problému

A.3.2 Vytvoření konstant

Vytvoříme si jednotlivé konstanty v okně *Nastavení problému / Definice proměnných* a zapíšeme konstanty podle tab. A.2.

μ_{r1}	1000
γ_1	$4 \cdot 10^6$
μ_{r2}	1
J_{ext}	$3 \cdot 10^7$

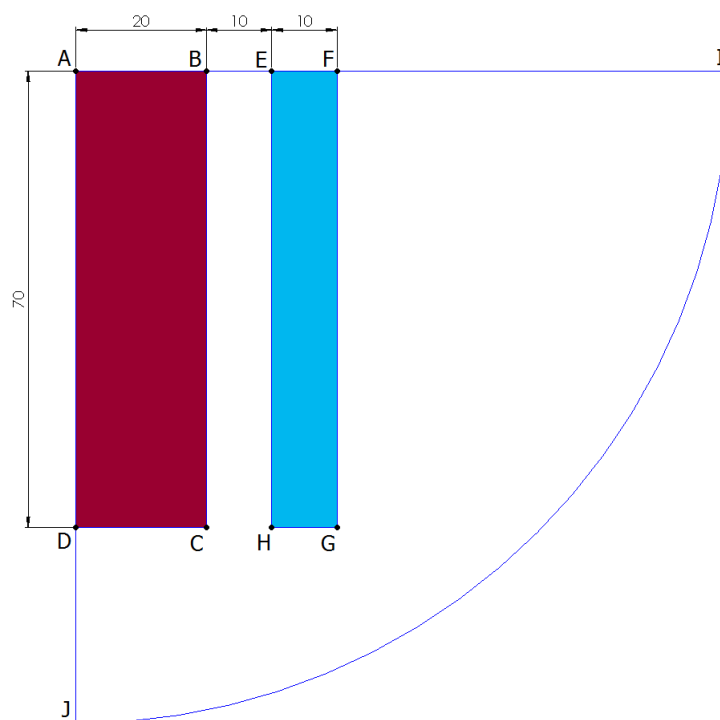
Tab. A.2: Hodnoty konstant

A.3.3 Vytvoření geometrie

Začneme vytvořením jednotlivých uzlů, přepneme se do režimu *Práce s uzly* a vytvoříme všechny potřebné uzly (používáme desetinnou tečku) podle tab. A.3.

Osa	A	B	C	D	E	F	G	H	I	J
R [m]	0	0,02	0,02	0	0,03	0,04	0,04	0,03	0,15	0
Z [m]	0	0	-0,07	-0,07	0	0	-0,07	-0,07	0	-0,15

Tab. A.3: Souřadnice jednotlivých uzlů



Obr. A.3: 2D model ocelového válce a cívky

Poté se přepneme do režimu *Práce s hranami* a jednotlivé uzly propojíme tak, aby vytvořili potřebný model - viz obr. A.3. Hrany AB BC CD DA tvoří obrys ocelového válce, hrany EF FG GH HE tvoří obrys cívky a nakonec hrany AD DJ tvoří osu symetrie a hrany IJ fiktivní hranici nekonečna. Protože využíváme symetrie i ve směru osy z a kreslíme jen polovinu modelu, hrany AB BE EF FI tvoří další osu symetrie.

A.3.4 Doplnění fyzikálních podmínek

Nyní vytvoříme okrajovou podmínku, nazveme ji *Antisymetrie* a necháme vektorový magnetický potenciál $\mathbf{A} = 0$. Označíme hrany AD DJ JI a přiřadíme jim okrajovou podmínku *Antisymetrie*. Druhou okrajovou podmínku nazveme *Symetrie*, vybereme u ní povrchovou proudovou hustotu $K_0 = 0$ a přiřadíme ji k hranám AB BE EF FI.

Nakonec v režimu *Práce se značkami oblasti* umístíme značky do jednotlivých oblastí a vytvoříme tři různé materiály. Jednotlivým materiálům nastavíme parametry podle tab. A.4 nebo můžeme využít vytvořených konstant.

Cívka by se v praxi samozřejmě také zahřívala, ale protože řešíme jen ilustrativní příklad dáme cívce nulovou elektrickou vodivost, abychom řešili jen ohřev ve válci, který je předmětem našeho zajmu.

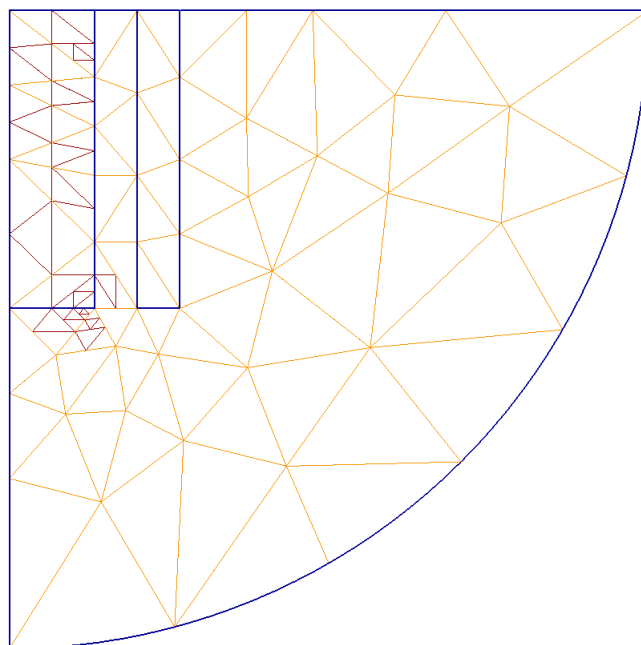
	Vzduch	Cívka	Ocelový válec
Relativní permeabilita μ_r [-]	1 nebo μ_{r2}	1 nebo μ_{r2}	1000 nebo μ_{r1}
Proudová hustota J_{ext} [$\frac{A}{m^2}$]	0	$3 \cdot 10^7$ nebo J_{ext}	0
Elektrická vodivost σ [$\frac{S}{m}$]	0	0	$4 \cdot 10^6$ nebo γ_1

Tab. A.4: Zadané hodnoty pro různé materiály

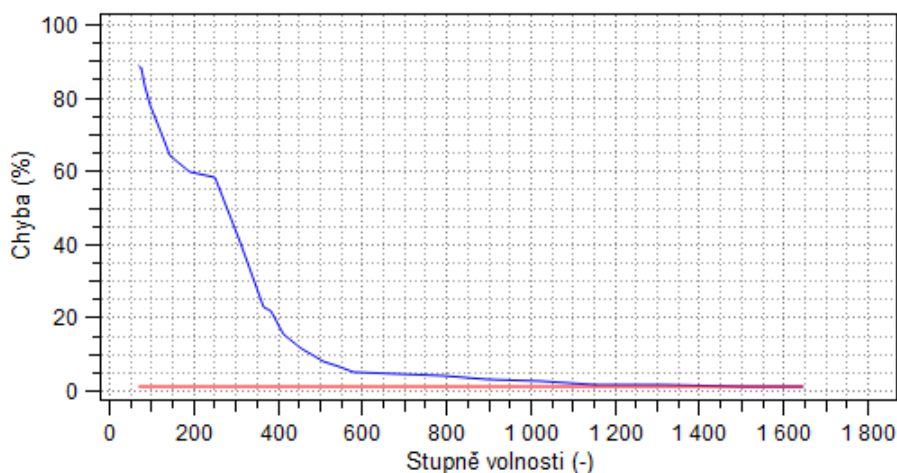
A.3.5 Diskretizace a řešení

Definiční oblast se při řešení diskretizuje na jednotlivé elementy, byla vybrána trojúhelníková síť, a proto vidíme na obr. A.4 jednotlivé trojúhelníky počáteční síť (světlá barva) a výsledné síť (tmavá barva), která vznikla po 24 krocích hp-adaptivity. Rychlost, s jakou klesá chyba v závislosti na počtu stupňů volnosti, je znázorněna na obr. A.5 (ten získáme v záložce *Adapt.conv.* po kliknutí na ikonku *Vyřešit problém*).

Po 24 krocích chyba klesla již pod 1 %, které jsme požadovali v zadání, a proto se řešení již zobrazí.



Obr. A.4: Diskretizovaná definiční oblast pro frekvenci 10 Hz



Obr. A.5: Závislost chyby na počtu stupňů volnosti pro frekvenci 10 Hz

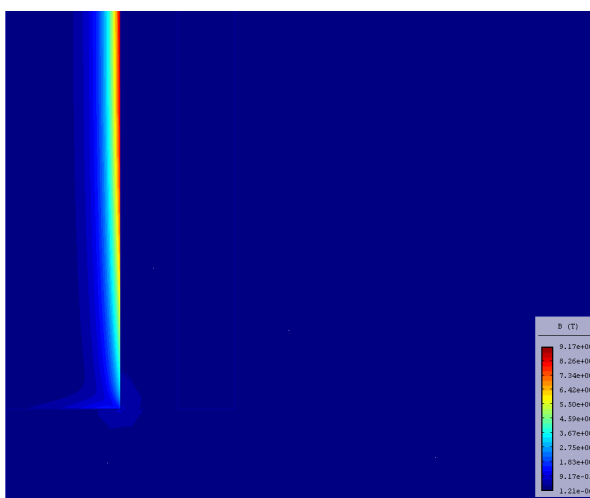
A.4 Diskuze výsledků

Výsledkem může být zobrazení výsledného pole, vykreslení veličiny do grafu po zadání přímce nebo mnoho různých veličin, které nám program spočítá v bodě, v přímce nebo v ploše.

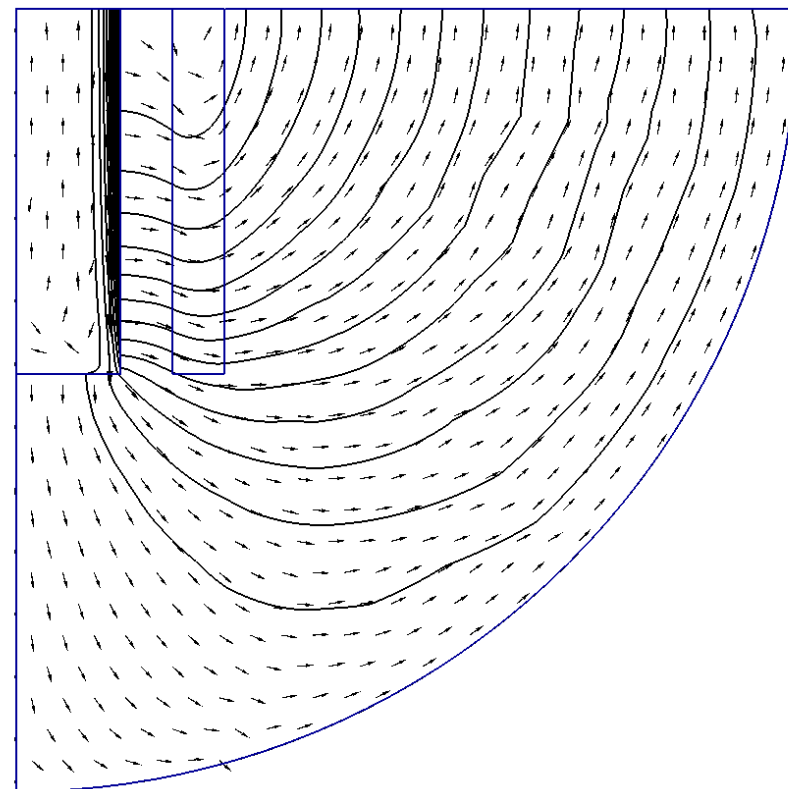
A.4.1 Rozložení výsledného pole

Pro zobrazení siločár a vektorů je zaškrtneme v levém okně a vybereme *2D Bez zobrazení*, aby nebyly rušeny zobrazením skalárního pole vybrané veličiny. V další záložce *Postprocessor* si můžeme nastavit počet a velikost vektorů i siločár. V obr. A.7 bylo zvoleno 14 kontur se šířkou 1,5 a 50 vektorů v měřítku 1,9.

Abychom si ověřili, že prakticky veškerá magnetická indukce bude v ocelovém jádru, zvolíme *2D Skalární zobrazení* a veličinu *Magnetickou indukci*, výsledek vidíme na obr. A.6.



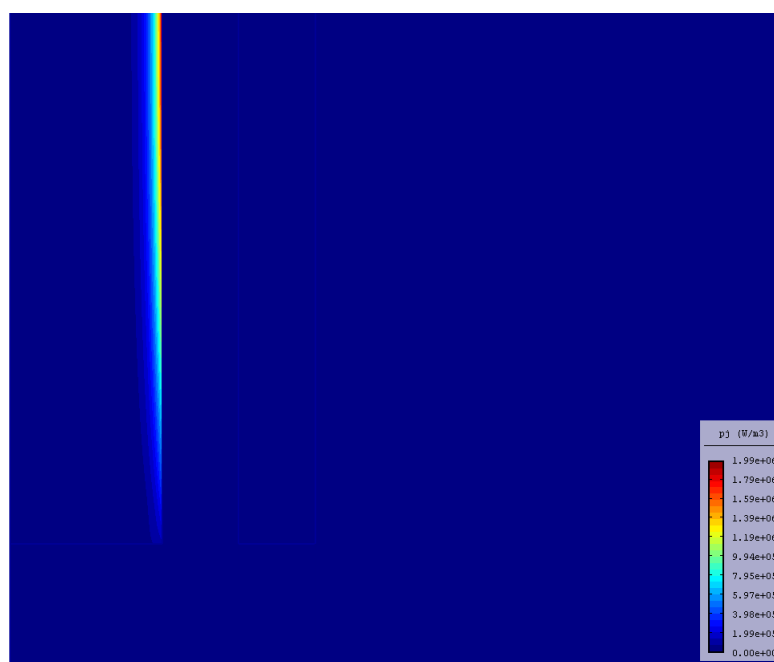
Obr. A.6: Rozložení magnetické indukce pro frekvenci 10 Hz



Obr. A.7: Zobrazení siločár a vektorů magnetické indukce pro frekvenci 10 Hz

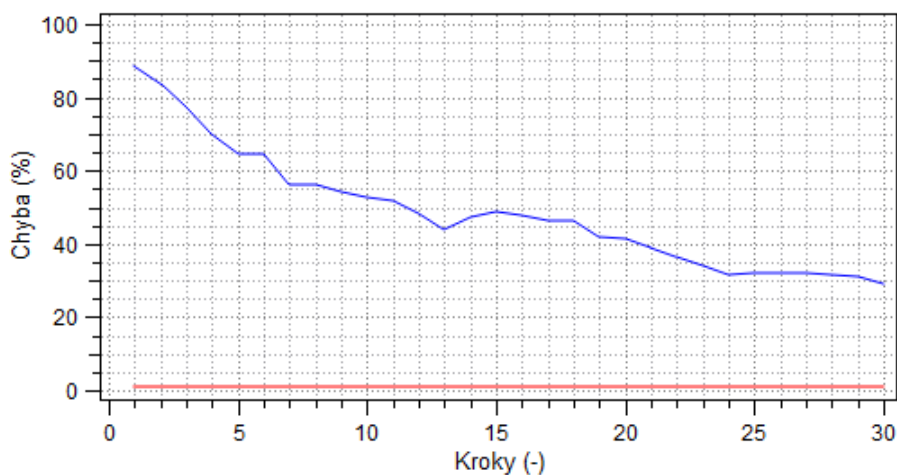
A.4.2 Výpočet Jouleových ztrát

Program nám umožní zobrazení měrných Jouleových ztrát pomocí skalárního pole, jako veličinu vybereme *Ztráty*, výsledek pro frekvenci 10 Hz je zobrazen na obr. A.8.

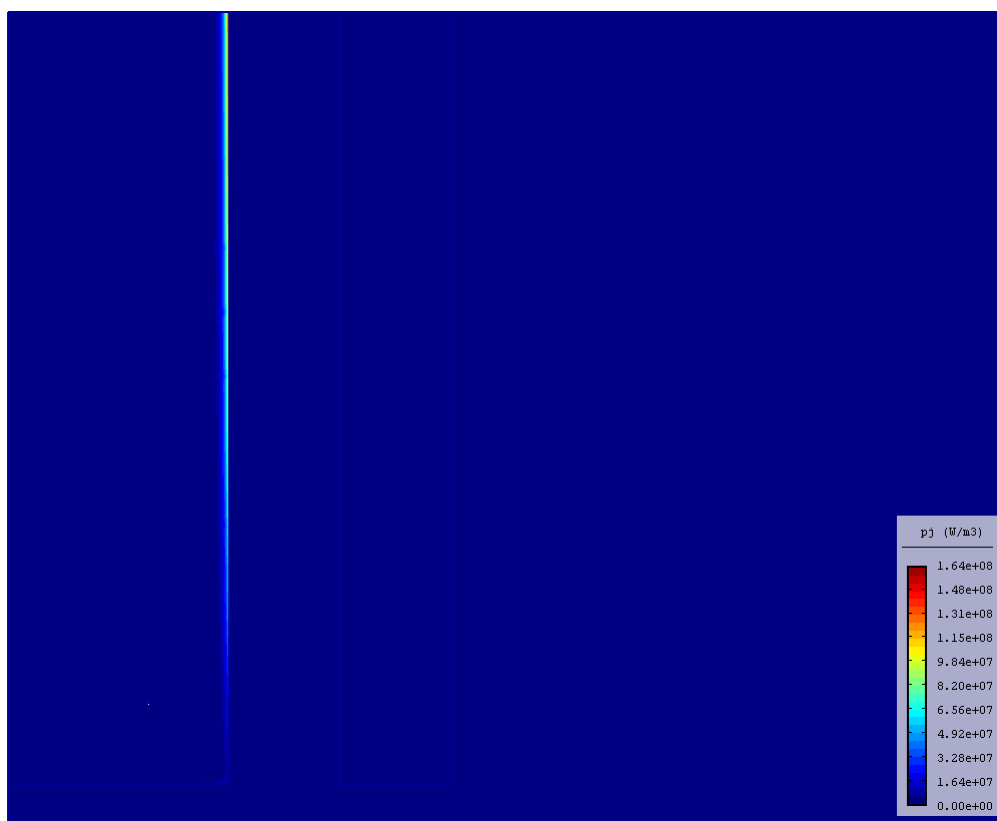


Obr. A.8: Zobrazení měrných Jouleových ztrát pro frekvenci 10 Hz

Nyní změním frekvenci pro zobrazení nového skalárního pole a následně výpočtů Jouleových ztrát. To provedeme změnou hodnoty zadané frekvence v nabídce *Nastavení problému*, nyní zadáme frekvenci 100 Hz, výsledné rozložení měrných Jouleových ztrát se změní a výsledek vidíme na obr. A.10. Při změně frekvence se změní i průběh výpočtu a rychlost konvergence chyby k naší požadované hodnotě, pro vyšší frekvence je výpočet náročnější, a proto pro frekvenci 1000 Hz již s 30 kroky hp-adaptivity nedosáhneme přesnosti pod 1 %, ale jen 29,261 %, viz obr. A.9



Obr. A.9: Závislost chyby na počtu kroků hp-adaptivity pro frekvenci 1000 Hz



Obr. A.10: Zobrazení měrných Jouleových ztrát pro frekvenci 100 Hz

Pro zjištění celkových Jouleových ztrát klikneme v horní liště na ikonku obdélníku, čímž aktivujeme výběr ploch, následně klikneme do plochy ocelového válce, poté se vlevo v záložce *Povrchové integrály* zobrazí pod názvem *Ztráty* hodnota Jouleových ztrát v ploše ocelového válce.

Pro frekvenci $f = 0 \text{ Hz}$ jsou celkové Jouleovy ztráty $P = 0 \text{ W}$

Pro frekvenci $f = 10 \text{ Hz}$ jsou celkové Jouleovy ztráty $P = 1,284 \cdot 10^1 \text{ W}$

Pro frekvenci $f = 100 \text{ Hz}$ jsou celkové Jouleovy ztráty $P = 3,219 \cdot 10^2 \text{ W}$

Pro frekvenci $f = 1000 \text{ Hz}$ jsou celkové Jouleovy ztráty $P = 5,008 \cdot 10^4 \text{ W}$

Z vypočtených hodnot je patrné, že pro zvyšující se frekvenci se i Jouleovy celkové ztráty v ocelovém válci zvyšují. Proto se válec víc ohřívá. Pro rychlý indukční ohřev je tedy vhodné používat vyšší frekvence.

A.4.3 Hloubka vniku

Pro zjištění hloubky vniku využijeme analytický i numerický postup a výsledky porovnáme. Analytický postup spočívá ve využití vztahu A.1, který pracuje s materiálovými konstantami a frekvencí pole.

$$\delta = \sqrt{\frac{2}{\omega \cdot \mu \cdot \gamma}} \quad (\text{A.1})$$

Numerický postup spočívá v zjištění velikosti magnetické indukce od hrany válce k jeho středu. K tomu využijeme možnosti grafu, kde zadáme přímkou podle tab. A.5. Z grafického vyjádření veličiny magnetické indukce poté odečteme vzdálenost, kde hodnota veličiny klesne na 37 %.

	$R[\text{m}]$	$Z[\text{m}]$
Počátek	0,02	-0,02
Konec	0	-0,02

Tab. A.5: Hodnoty souřadnic

Pro hodnotu frekvence 10 Hz byla z grafu odečtena hodnota $\delta_{N10} = 0,0025 \text{ m}$. A dosazením hodnot do vztahu vyšla $\delta_{A10} = 2,2508 \cdot 10^{-3} \text{ m}$. Obě hodnoty vyšly prakticky stejně a hloubka vniku pro frekvenci 10 Hz je 0,0025 m.

Pro vyšší frekvenci čekáme, že bude hloubka vniku menší. Nyní ji budeme zjišťovat pro frekvenci 100 Hz. Podle vztahu vyšla $\delta_{A100} = 0,712 \cdot 10^{-3} \text{ m}$ a odečtením z grafu vyšla $\delta_{N100} = 0,0007 \text{ m}$. Opět jsou hodnoty prakticky stejné a hloubka vniku je pro vyšší frekvenci menší.

Příloha B

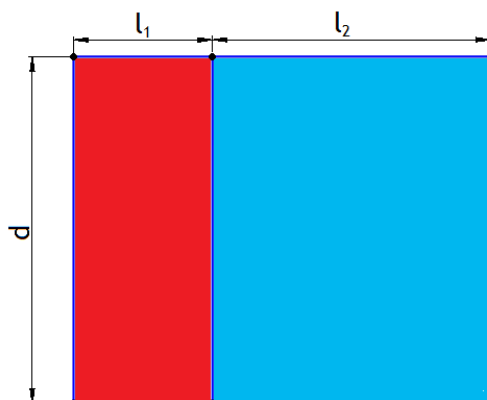
Deskový kondenzátor s respektováním okrajového jevu

B.1 Formulace problému

Je dán deskový kondenzátor se dvěma různými dielektriky, který bude řešen jako dvou dimenzionální úloha v kartézských souřadnicích. Kondenzátor bude obklopen vzduchem až do fiktivní hranice, která je teoreticky v nekonečnu. Pro reálné řešení bude nakreslena jako kruh kolem kondenzátoru. Zadané parametry pro konkrétní řešení daného problému jsou: napětí U mezi elektrodami, relativní permitivita vzduchu ε_{r0} , relativní permitivita prvního dielektrika ε_{r1} a relativní permitivita druhého dielektrika ε_{r2} . Rozměry kondenzátoru jsou: šířka elektrod d , délka prvního dielektrika l_1 a délka druhého dielektrika l_2 - viz obr. B.1, dielektrikum jedna je zobrazeno červenou plochou a dielektrikum dva modrou plochou.

$$\varepsilon_{r0} = 1 \quad \varepsilon_{r1} = 3 \quad \varepsilon_{r2} = 9 \quad U = 100 \text{ V}$$

$$d = 100 \text{ mm} \quad l_1 = 40 \text{ mm} \quad l_2 = 80 \text{ mm}$$



Obr. B.1: 2D model kondenzátoru

B.2 Definice modelu

B.2.1 Rozdělení oblastí

Oblast pro řešení musíme rozdělit na jednotlivé oblasti podle zadaných parametrů. Máme celkem tři oblasti. První dielektrikum, druhé dielektrikum a obklopující vzduch.

B.2.2 Okrajové podmínky

Pro fiktivní hranici použijeme Neumannovu okrajovou podmínku, která definuje derivaci elektrického potenciálu ve směru normály. Protože se jedná o fiktivní hranici, která je teoreticky v nekonečnu, bude derivace elektrického potenciálu nulová. Na obě elektrody použijeme Dirichletovu okrajovou podmínku, která určuje velikost potenciálu.

B.3 Vytvoření modelu

B.3.1 Nastavení problému

Zprvá musíme správně zadat parametry problému, v nabídce *Nastavení problému* zadáme vše podle tab. B.1. Jedná se o ustálený stav, proto jsou možnosti zadání frekvence a časových kroků znemožněny.

Typ problému	Kartézský
Fyzikální pole	Elektrostatické pole
Druh analýzy	Ustálený stav
Typ sítě	Trojúhelníková
Počet zjemnění	0
Řád polynomu	1
Typ adaptivity	hp-adaptivita
Počet adaptivních kroků	20
Tolerance adaptivity [%]	1

Tab. B.1: Nastavení parametrů problému

B.3.2 Vytvoření konstant

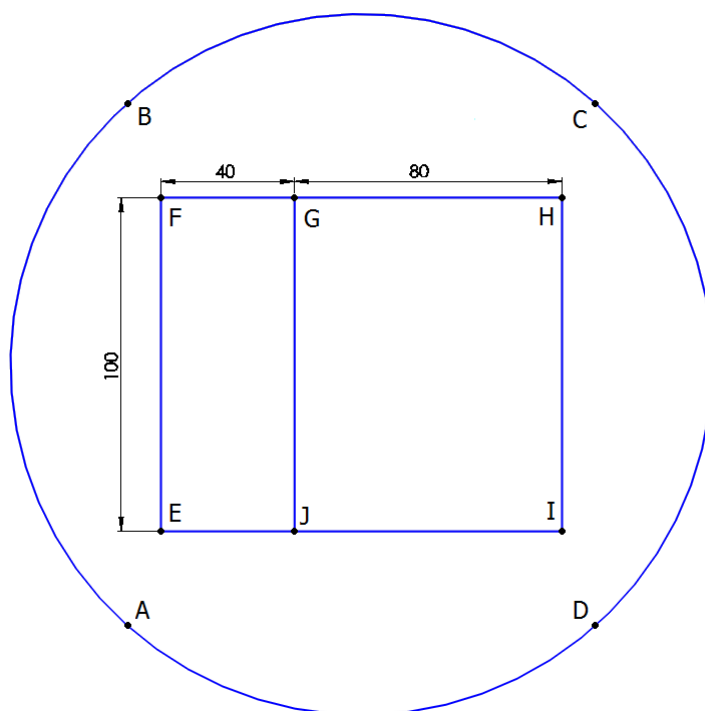
Dále pro lepší práci se zadáváním parametrů následujícím hranám a značkám oblastí si vytvoříme konstanty. To provedeme v okně *Nastavení problému / Definice proměnných*, kde všechny konstanty vytvoříme podle tab. B.2.

U	100
ε_{r0}	1
ε_{r1}	3
ε_{r2}	9

Tab. B.2: Hodnoty konstant

B.3.3 Vytvoření geometrie

Začneme vytvořením jednotlivých uzlů, přepneme se do režimu *Práce s uzly* a vytvoříme všechny potřebné uzly (používáme desetinnou tečku) podle tab. B.3.



Obr. B.2: 2D model kondenzátoru s kruhovou fiktivní hranicí

Osa	A	B	C	D	E	F	G	H	I	J
R [m]	0	0	0,52	0,52	0,2	0,2	0,24	0,32	0,32	0,24
Z [m]	0	0,5	0,5	0	0,2	0,3	0,3	0,3	0,2	0,2

Tab. B.3: Souřadnice jednotlivých uzlů

Poté se přepneme do režimu *Práce s hranami* a jednotlivé uzly propojíme tak, aby vytvořili potřebný model viz obr. B.2. Hrana EF tvoří levou elektrodu, hrana HI tvoří pravou elektrodu a hrany AB BC CD DA tvoří fiktivní hranici nekonečna. Hrany fiktivní hranice nekonečna musíme zalomit, aby vytvořili kruh, to provedeme tak, že po kliknutí na příslušnou hranu jí nastavíme úhel 90°.

B.3.4 Doplnění fyzikálních podmínek

Dále přiřadíme okrajové podmínky jednotlivým hranám. Máme zadané napětí mezi elektrodami, takže na levou elektrodu dáme hodnotu napětí o velikosti nula a na pravou elektrodu zadáme velikost napětí U . Na fiktivní hranici nekonečna necháme nulovou povrchovou hustotu náboje.

Poté vložíme značky do jednotlivých oblastí v režimu *Práce se značkami oblasti* a vytvoříme tři různé materiály. Od levé elektrody začíná dielektrikum jedna a od pravé dielektrikum dva. Vytvořeným materiálům přiřadíme hodnoty podle tab. B.4 nebo zadáme název vytvořené konstanty.

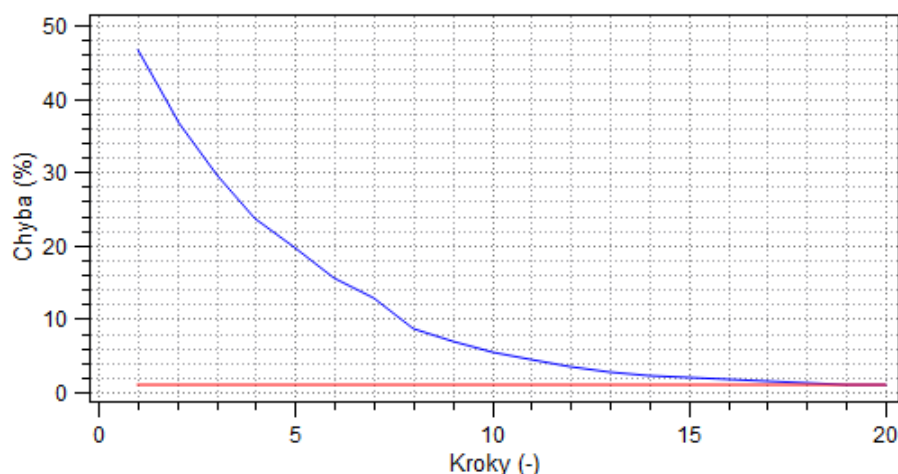
	Vzduch	Dielektrikum jedna	Dielektrikum dva
Relativní permitivita ε_r [-]	ε_{r0}	ε_{r1}	ε_{r2}
Hustota náboje ρ [$\frac{C}{m^3}$]	0	0	0

Tab. B.4: Zadané hodnoty pro různé materiály

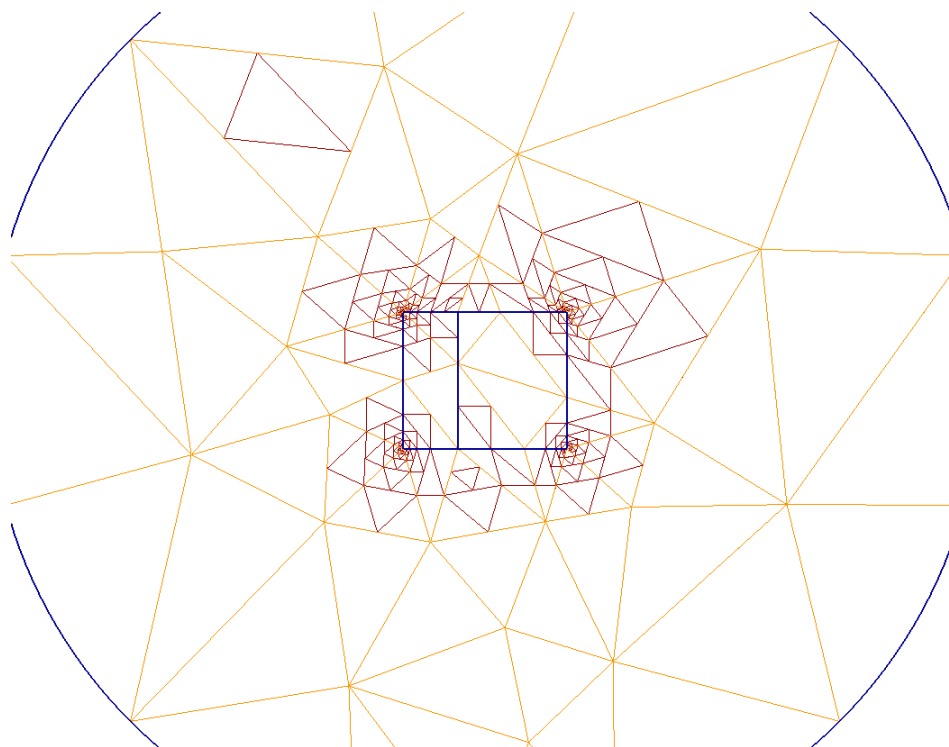
B.4 Diskuze výsledků

B.4.1 Diskretizace a řešení

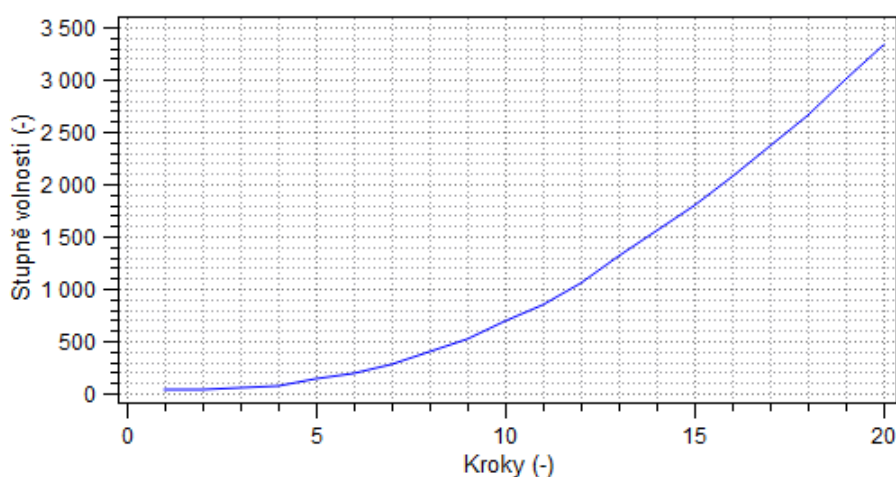
Po 18 krocích hp-adaptivity je chyba 0,84 % a to již splňuje naše zadání a výpočet skončí s 276 uzly - viz obr. B.4, 73 elementy a 3007 stupni volnosti po 48,739 vteřinách (čas výpočtu je samozřejmě ovlivněn hardwarovými parametry použitého počítače). V dalších záložkách průběhu řešení jsou závislosti chyby - viz obr. B.3 a stupně volnosti na počtu kroků hp-adaptivity - viz obr. B.5 a jejich kombinace na poslední záložce.



Obr. B.3: Závislost chyby na počtu kroků hp-adaptivity



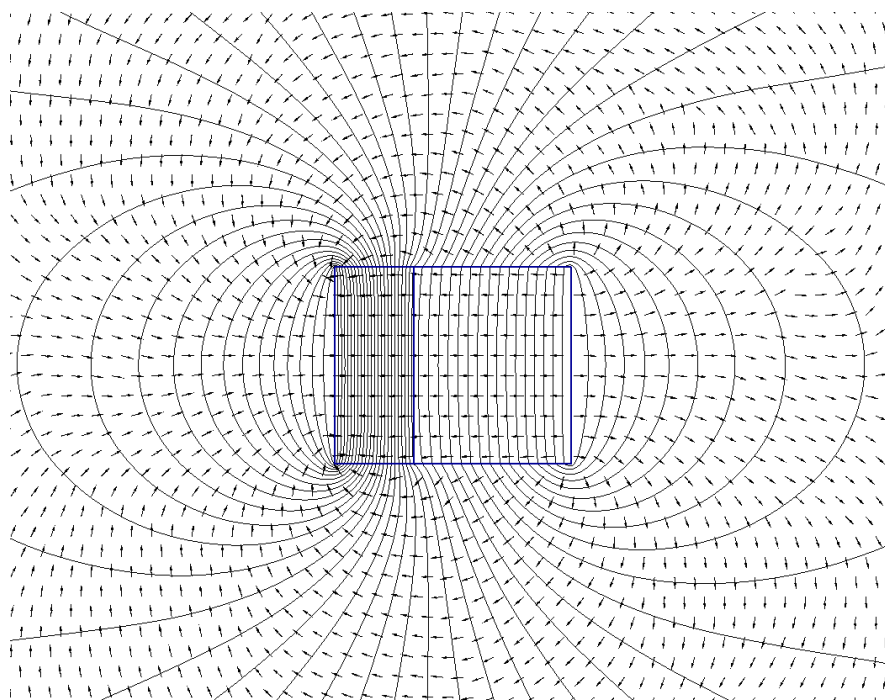
Obr. B.4: Zobrazení počáteční (světlá) a řešené sítě (tmavá)



Obr. B.5: Závislost stupňů volnosti na počtu kroků hp-adaptivity

B.4.2 Rozložení výsledného pole

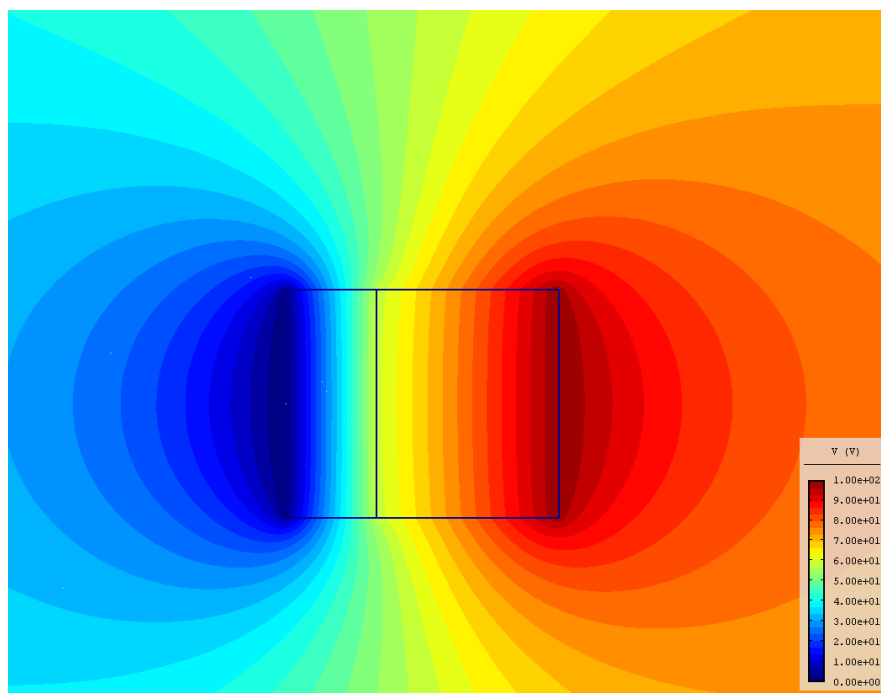
Po výpočtu se vlevo zobrazilo okno *Nastavení zobrazení*, kde si můžeme vybrat, co program vykreslí. Například pro zobrazení rozložení ekvipotenciál zaškrtneme v nabídce *Kontury*, 2D nabídku přepneme do *Bez zobrazení*, aby nás nerušilo barevné pozadí. Dále v záložce *Postprocesor* vybereme *Kontury a vektory*, kde můžeme nastavit počet a šířku zobrazených kontur, potvrzením se zobrazí požadovaný výsledek - viz obr. B.6. Další příklad je zobrazení vektorů intenzity elektrického pole, pro něj zaškrtneme *Vektory* a opět si v další záložce můžeme zvolit počet a měřítko vektorů. V obr. B.6 bylo zvoleno 40 kontur se šířkou 1,20 a zároveň 100 vektorů v měřítku 2,20.



Obr. B.6: Rozložení ekvipotenciál a vektorů intenzity elektrického pole ve vyšetřované oblasti

Z obr. B.6 je patrné, že řešení je správné, protože ekvipotenciály respektují fyzikální rozložení mezi elektrodami a vektory směřují od elektrody s vyšším potenciálem k elektrodě s nižším potenciálem.

Na obr. B.7 je zobrazeno rozložení skalárního elektrického potenciálu vidíme, jak se elektrický potenciál snižuje z nejvyšší hodnoty na pravé elektrodě k nejnižší na levé.

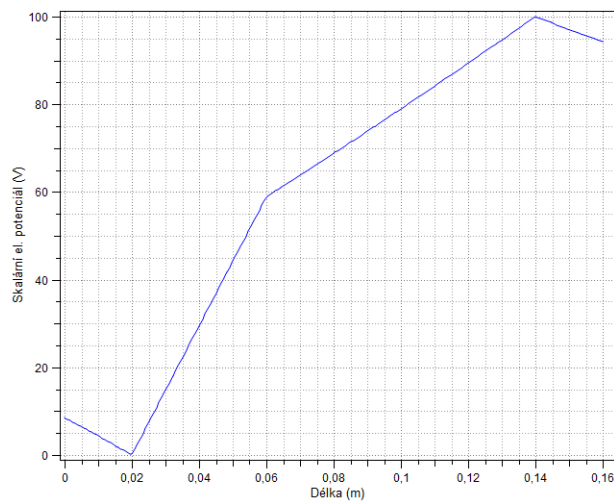


Obr. B.7: Rozložení skalárního elektrického potenciálu ve vyšetřované oblasti

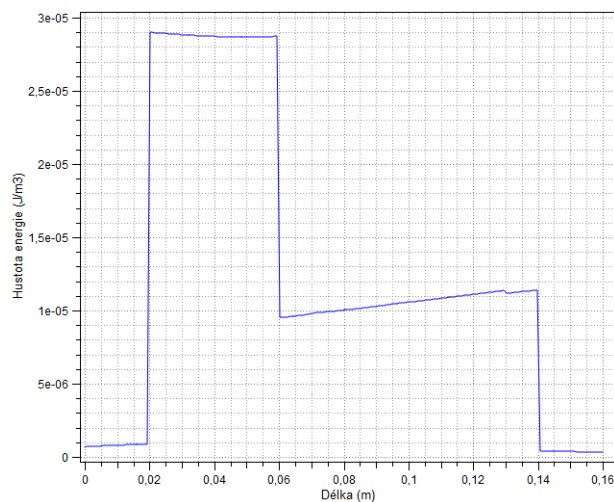
Další možnost zobrazení výsledků je pomocí grafu, respektive tabulky výsledků. Po kliknutí na ikonku v horní liště, která vypadá jako graf funkce sinus, se dostaneme do nabídky *Graf*. Zde si můžeme vybrat různé veličiny a případně jejich složky. Pomocí souřadnic zadáme úsečku, pro kterou se má výsledek zobrazit. Například pro zobrazení skalárního elektrického potenciálu v ose kondenzátoru, zadáme následující souřadnice.

Počátek $x = 0,18$, $y = 0,25$ a konec $x = 0,34$, $y = 0,25$.

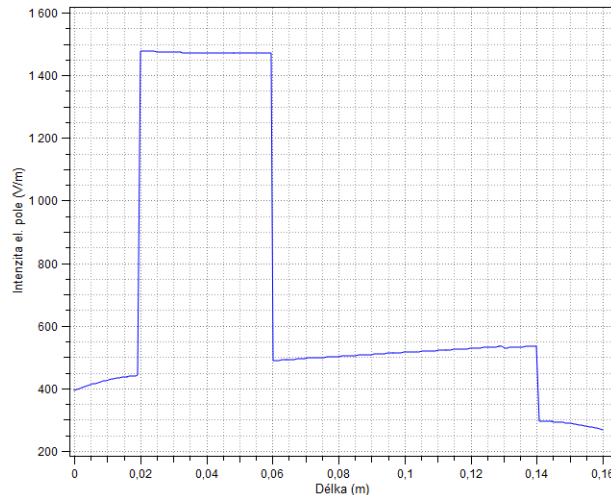
Vybereme *Délka* a zvolíme veličinu *Skalární el. potenciál* a složku *Skalár*. Když si okno s grafem zminimalizujeme vidíme, že se nám zadaná úsečka znázornila v řešené ploše úlohy. Graf si můžeme uložit jako obrázek (viz obr. B.8) a nebo exportovat tabulku výsledků se syntaxí pro vykreslení grafu, například v Matlabu nebo Mathematice. Další možnosti grafů jsou vidět na obr. B.9 pro hustotu energie pole a na obr. B.10 je vykreslena intenzita elektromagnetického pole, oba grafy opět pro střed kondenzátoru, který nás zajímá.



Obr. B.8: Skalární elektrický potenciál v ose kondenzátoru



Obr. B.9: Hustota energie v ose kondenzátoru



Obr. B.10: Intenzita elektromagnetického pole v ose kondenzátoru

B.5 Výpočet kapacity kondenzátoru

Simulační program umožňuje výpočet kapacity, například přes rozložení energie elektrického pole.

$$C = \frac{2 * W_e}{U^2} \quad [F] \quad (B.1)$$

Nebo pomocí elektrického náboje na jedné z elektrod.

$$C = \frac{Q}{U} \quad [F] \quad (B.2)$$

Jak bylo zmíněno výše, použijeme *Objemové integrály* a jejich možnost vypočítat energii. Označíme obě části kondenzátoru s různou permitivitou a oblast okolního vzduchu, vypočtenou energii si pomocí kliknutí pravým tlačítkem myši na její hodnotu uložíme do schránky a klikneme do konzole v dolní části obrazovky. Vložíme ze schránky hodnotu energie a doplníme ostatní veličiny podle vztahu B.1 (Protože jsme si hodnotu napětí uložili jako proměnnou, můžeme jí zavolat pod její referencí U) a po stisknutí klávesy *Enter* se zobrazí výsledek. Kapacita kondenzátoru vyšla $C = 4,968 \cdot 10^{-11}$ F. Přesnost výpočtu energie je omezena vzdáleností fiktivní hranice.

Druhou možnost pomocí náboje provedeme v režimu *Povrchové integrály* a označíme jednu z elektrod, například tu s potenciálem napětí. Opět si hodnotu náboje uložíme do schránky, poté označíme druhou elektrodu a její výsledek náboje zprůměrujeme s předchozím a následně pomocí konzole a vztahu B.2 vypočteme výslednou kapacitu, která nyní vyšla $C = 4,967 \cdot 10^{-11}$ F. Tato hodnota by měla být přesnější než u předchozího příkladu.

Příloha C

Model elektromagnetu

C.1 Formulace problému

Model je tvořen stacionárním magnetickým polem vyvolaným stejnosměrným elektrickým proudem a permanentními magnety - viz obr. C.1. Je dána cívka tvořená několika tenkými měděnými vodiči s relativní permeabilitou μ_{r0} a elektrickou vodivostí γ protékaná proudem s proudovou hustotou J . Kotva i železo jsou tvořeny lineární ocelí s relativní permeabilitou μ_{r1} . Permanentní magnety jsou tvořeny materiálem s remanentní magnetickou indukcí B_r a relativní permeabilitou μ_{r2} . Celý model je obklopen vzduchem s relativní permeabilitou μ_{r0} . Rozměry všech částí elektromagnetu jsou uvedeny na obr. C.1, kde jsou pomocí barev rozděleny jednotlivé oblasti, kotva je znázorněna šedivou barvou, cívky jsou kružnice a permanentní magnety jsou znázorněny podle jejich směru polarity modrou a červenou barvou. Poslední neobarvená část tvoří železo, které rozvádí magnetickou indukci do permanentních magnetů.

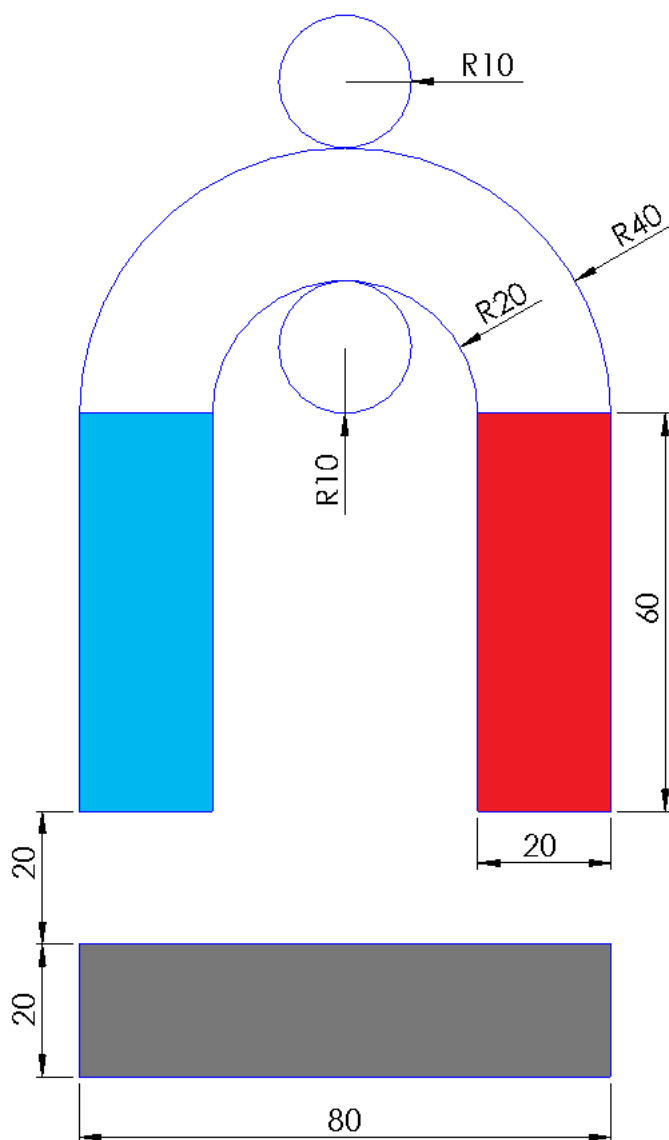
$$\mu_{r0} = 1 \quad \mu_{r1} = 1000 \quad \mu_{r2} = 1,05$$

$$\gamma = 6 \cdot 10^7 \frac{\text{S}}{\text{m}} \quad J = 2 \cdot 10^4 \frac{\text{A}}{\text{m}^2} \quad B_r = 1,07 \text{ T}$$

C.2 Definice modelu

C.2.1 Rozdělení oblastí

Bude potřeba vytvořit šest rozdílných oblastí. Dvě cívky pro jednotlivé směry proudu, dva magnety každý s jinou orientací, kotva a železo mezi cívkami a nakonec okolní vzduch. Všem oblastem musíme zadat parametry podle zadání a u magnetů ještě zvolit jejich směr remanence.



Obr. C.1: Rozměry elektromagnetu v cm

C.2.2 Okrajové podmínky

Pouze jediná okrajová podmínka bude muset být vytvořena a to pro fiktivní hranici nekonečna. Této fiktivní hranici nekonečna přiřadíme parametr *Vektorový mag. potenciál* tak, aby respektoval fyzikální rozložení pole. Protože v nekonečnu je vektorový magnetický potenciál nulový, zapíšeme nulu jako hodnotu parametru okrajové podmínky.

C.3 Vytvoření modelu

Model vytvoříme podle obr. C.2. Vytvoříme jednotlivé body podle souřadnic uvedených v tab. C.3, C.4, které spojíme hranami a vložíme do jednotlivých oblastí potřebné značky oblastí. Nakonec hranám i značkám oblastí přiřadíme parametry podle zadání.

C.3.1 Nastavení problému

Otevřeme okno *Nastavení problému*, kde se nastavují počáteční parametry problému. Veškeré parametry nastavíme podle tab. C.1.

Typ problému	Kartézský
Fyzikální pole	Magnetické pole
Druh analýzy	Ustálený stav
Typ sítě	Trojúhelníková
Počet zjemnění	0
Řád polynomu	1
Typ adaptivity	hp-adaptivita
Počet adaptivních kroků	20
Tolerance adaptivity [%]	1

Tab. C.1: Parametry nastavení problému

C.3.2 Vytvoření konstant

Pro jednotlivé hodnoty zadání budou vytvořeny konstanty tak, že se v okně *Nastavení problému* přepneme do druhé záložky *Definice proměnných* a konstanty vytvoříme podle tab. C.2.

μ_{r0}	1
μ_{r1}	1000
μ_{r2}	1,05
γ	6e7
J	2e4
B_r	1,07

Tab. C.2: Hodnoty konstant

C.3.3 Vytvoření geometrie

Jako první budou vytvořeny jednotlivé uzly geometrie podle tab. C.3, C.4. Při zadávání souřadnic používáme desetinnou tečku.

Body U, V, W a X nejsou zobrazeny na obr. C.2 kvůli tomu, že jsou příliš daleko od modelu, následně jejich propojením vytvoříme kruhovou hranici nekonečna.

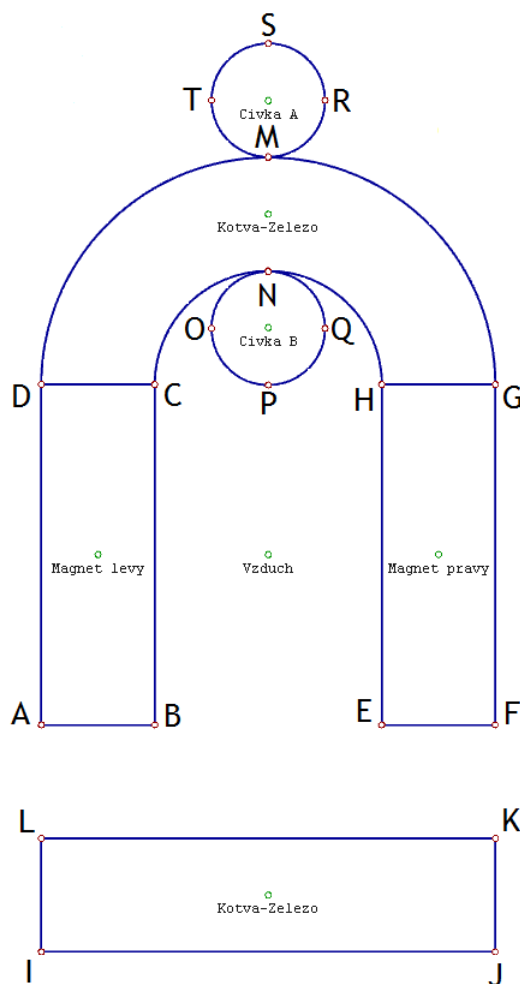
Protože by se hrany při propojení do kruhu cívky křížily s hranami železa bylo nutné body N a M vytvořit ještě jednou o 0,001 m posunuté jako body N' a M', na obr. C.2 jsou vidět jen jako jeden bod N a M. Aby obrys cívky tvořil kruh, byly i jeho ostatní body posunuty o tento rozměr, jak je uvedeno v tab. C.3.

Osa	A	B	C	D	E	F	G	H	I	J	K
X [m]	0	0,2	0,2	0	0,6	0,8	0,8	0,6	0	0,8	0,8
Y [m]	0	0	0,6	0,6	0	0	0,6	0,6	-0,4	-0,4	-0,2
Osa	L	M	M'	N	N'	O	P	Q	R	S	T
X [m]	0	0,4	0,4	0,4	0,4	0,3	0,4	0,5	0,5	0,4	0,3
Y [m]	-0,2	1	1,001	0,8	0,799	0,699	0,599	0,699	1,101	1,201	1,101

Tab. C.3: Souřadnice uzlů elektromagnetu

Osa	U	V	W	X
X [m]	0,4	2,4	0,4	-1,6
Y [m]	-1,75	0,35	2,35	0,35

Tab. C.4: Souřadnice uzlů fiktivní hranice kolem elektromagnetu



Obr. C.2: Model elektromagnetu s označením jednotlivých bodů

Jak vidíme na obr. C.2, propojením vytvořených bodů získáme obrys modelu elektromagnetu. Hrany AB BC CD DA tvoří obrys levého magnetu, hrany EF FG GH HE vytvářejí obrys pravého magnetu, hrany IJ JK KL LI tvoří obrys kotvy, hrany DC CN NH HG GM MD tvoří železo mezi cívkami a nakonec horní cívka je tvořena hranami MR RS ST TM a dolní cívka hranami OP PQ QN NO. Všechny hrany, které mají tvořit oblouk musíme ještě zalomit tak, že jim přidáme jako úhel velikost 90° .

Na obr. C.2 jsou vidět i již přidáné značky oblasti. My je tedy také vytvoříme a vložíme do jednotlivých oblastí.

C.3.4 Doplnění fyzikálních podmínek

Dále označíme všechny hrany kruhu, které tvoří fiktivní hranici nekonečna a přiřadíme jim okrajovou podmínku s hodnotou nula u vektorového magnetického potenciálu.

Poslední, co musíme udělat je, že vytvořeným značkám oblasti přiřadíme správný materiál, který vytvoříme podle tab. C.5 a tab. C.6. Můžeme při vytváření materiálů využít vytvořených konstant místo zapisování číselných hodnot zadání.

	Vzduch	Cívka A	Cívka B	Kotva-Železo
Relativní permeabilita μ_r [-]	μ_{r0}			μ_{r1}
Proudová hustota J_{ext} [$\frac{A}{m^2}$]	0	-J	J	0
Elektrická vodivost [$\frac{S}{m}$]	0	γ		0

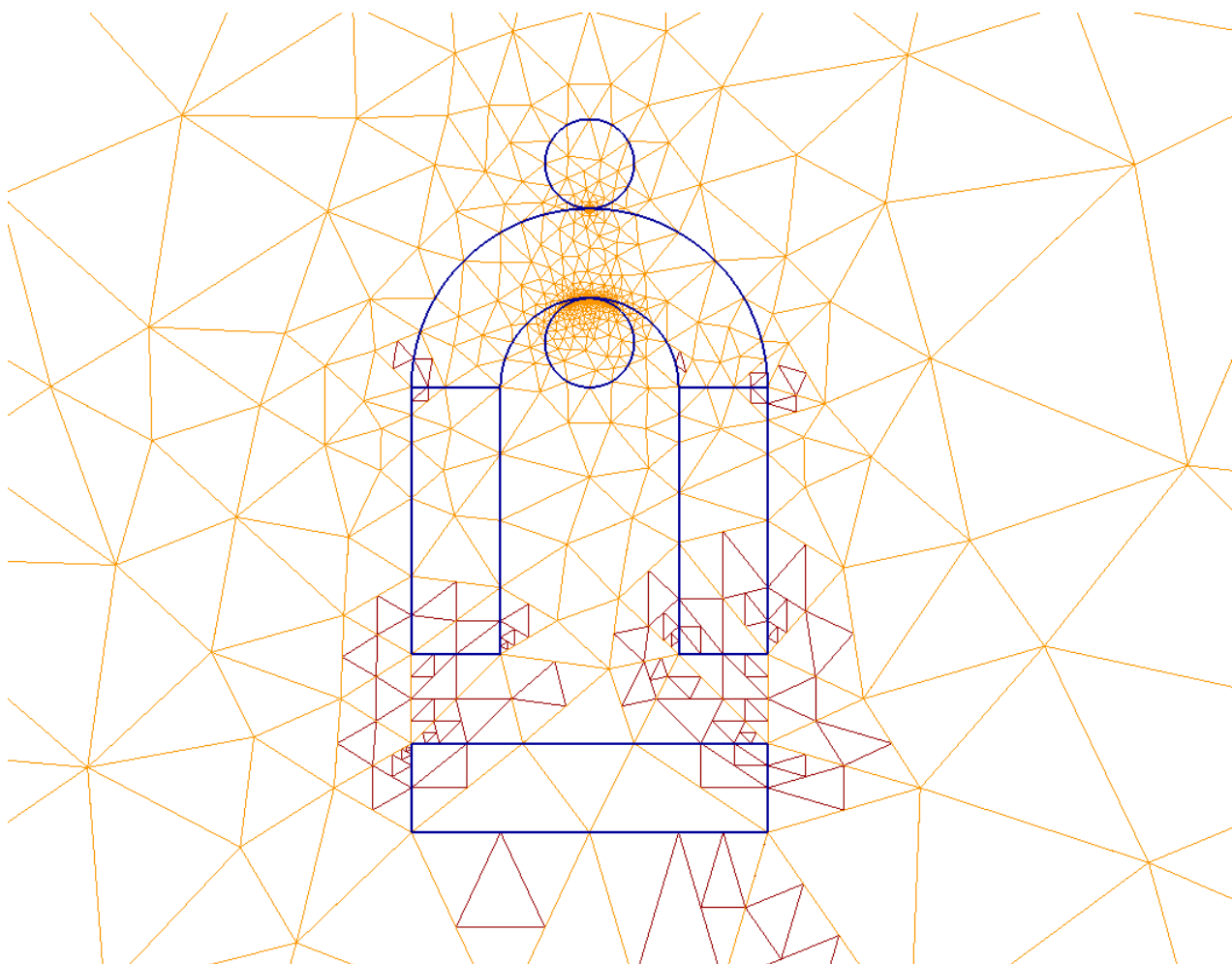
Tab. C.5: Zadané hodnoty pro cívky, vzduch a železo

	Magnet levý	Magnet pravý
Relativní permeabilita μ_r [-]	μ_{r2}	
Remanentní magnetická indukce B_r [T]	B_r	
Směr remanence α [°]	90	270

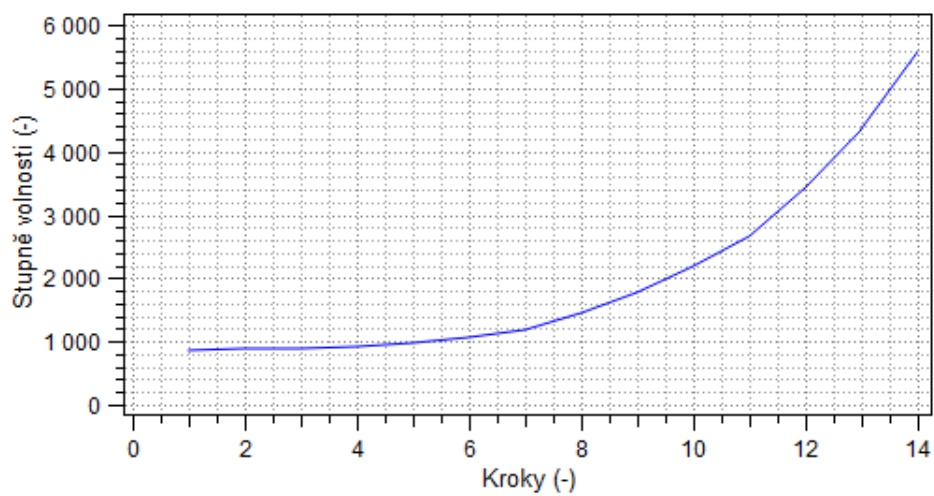
Tab. C.6: Zadané hodnoty pro pravý a levý magnet

C.3.5 Diskretizace a řešení

Nakreslený model se nejprve rozdělí na jednotlivé elementy, protože jsme si vybrali hp-adaptivitu, je při řešení postupně síť zjemňována (viz obr. C.3), dokud nedosáhne relativní chyba řešení pod zadanou hodnotu jednoho procenta. Průběh řešení vidíme v okně *Průběh...*, kde se postupně vypisují počty uzlů, elementů a stupňů volnosti v jednotlivých krocích hp-adaptivity. Po vyřešení jsou na dalších záložkách k dispozici grafy znázorňující průběh hp-adaptivity, respektive závislosti relativní chyby, stupňů volnosti na krocích výpočtu. Například rychlost, s jakou stoupají stupně volnosti s kroky hp-adaptivity, je zobrazena na obr. C.4.



Obr. C.3: Řešená a počáteční síť (počáteční světlou barvou)

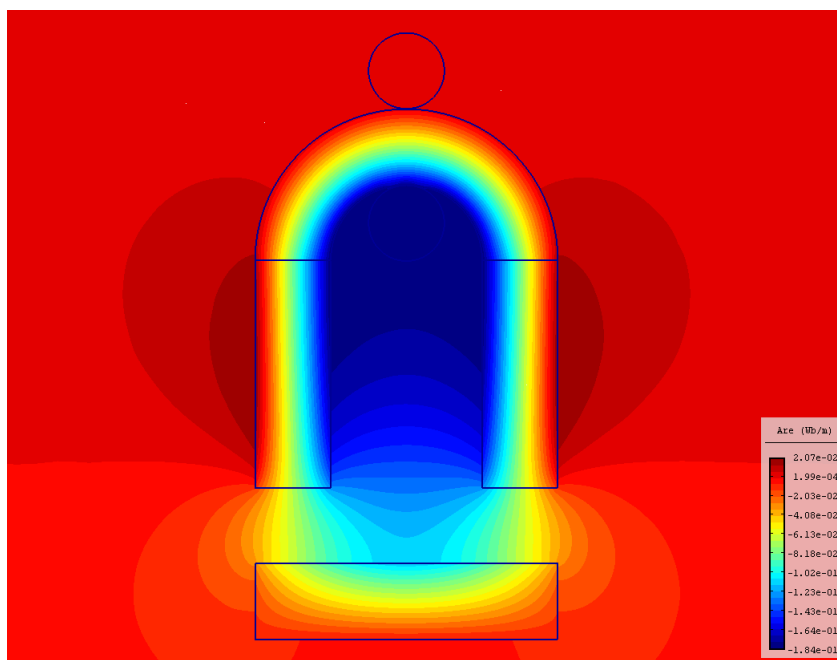


Obr. C.4: Závislost stupňů volnosti na počtu kroků hp-adaptivity

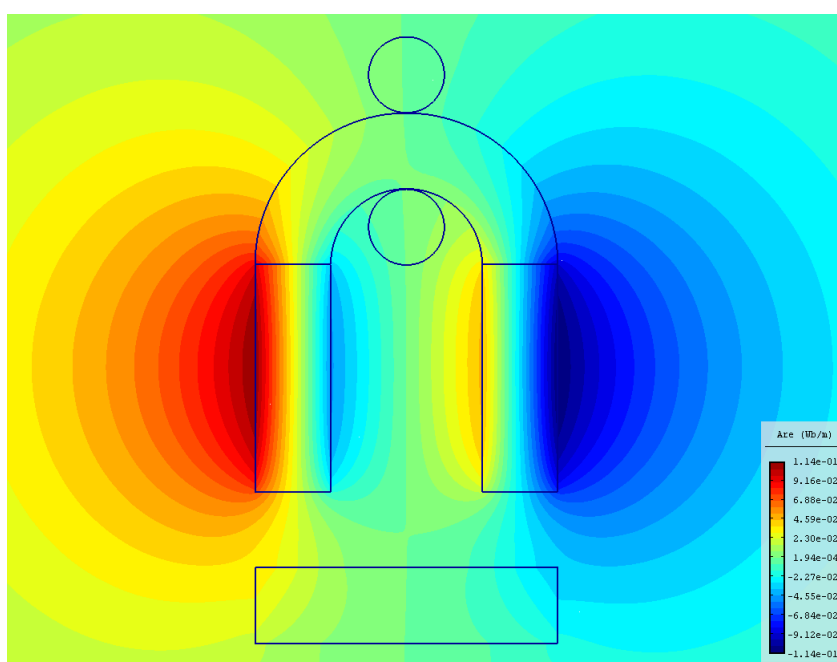
C.4 Diskuze výsledků

C.4.1 Rozložení výsledného pole

Na obr. C.5 je vektorový magnetický potenciál pro nesouhlasně uložené magnety a na obr. C.6 pro souhlasně. Nesouhlasné uložení zvyšuje účinnost elektromagnetu, protože se do plochy kotvy dostane mnohem větší energie.

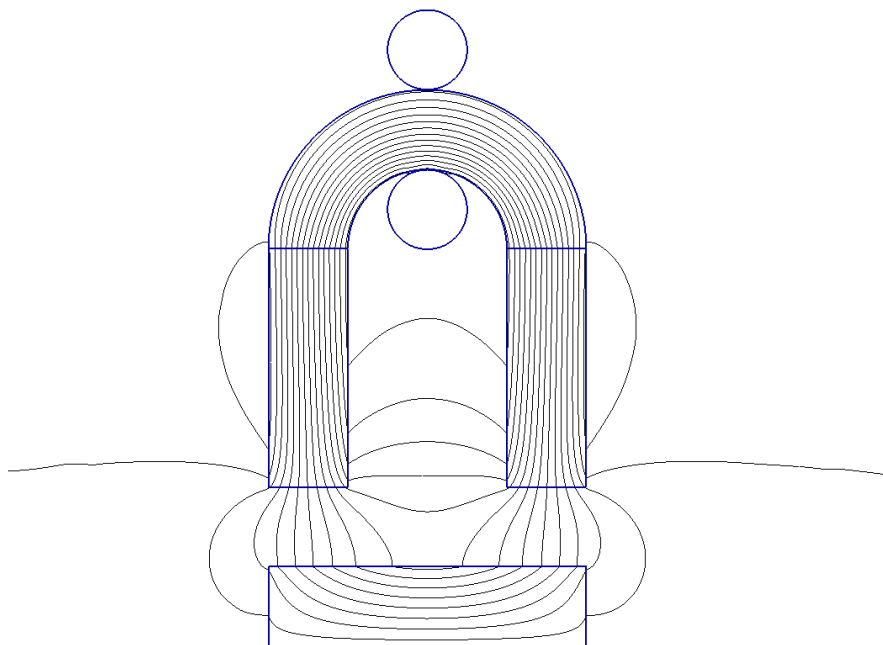


Obr. C.5: Rozložení vektorové magnetického potenciálu pro magnety nesouhlasně

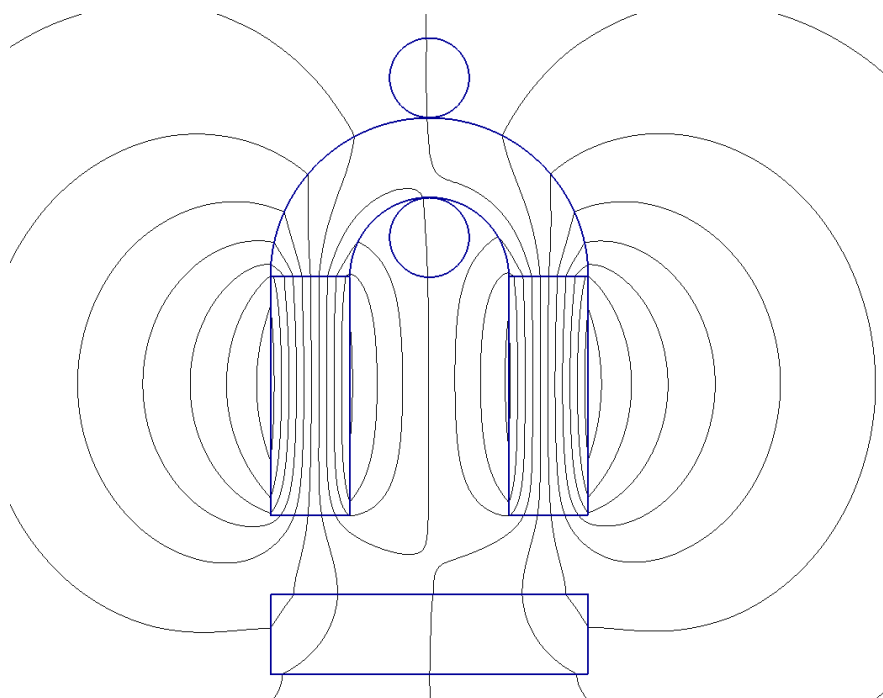


Obr. C.6: Rozložení vektorové magnetického potenciálu pro magnety souhlasně

Při zobrazení siločár je vidět, jak permanentní magnety, pokud jsou uloženy nesouhlasně, vedou siločáry směrem ke kotvě viz obr. C.7. Pokud jsou však oba magnety uloženy souhlasně, sice stále část siločár vedou svým vnitřkem, ale protože dolní okraj mají na stejné polaritě, navzájem se odpuzují a směřují siločáry od kotvy pryč, to je zobrazeno na obr. C.8.

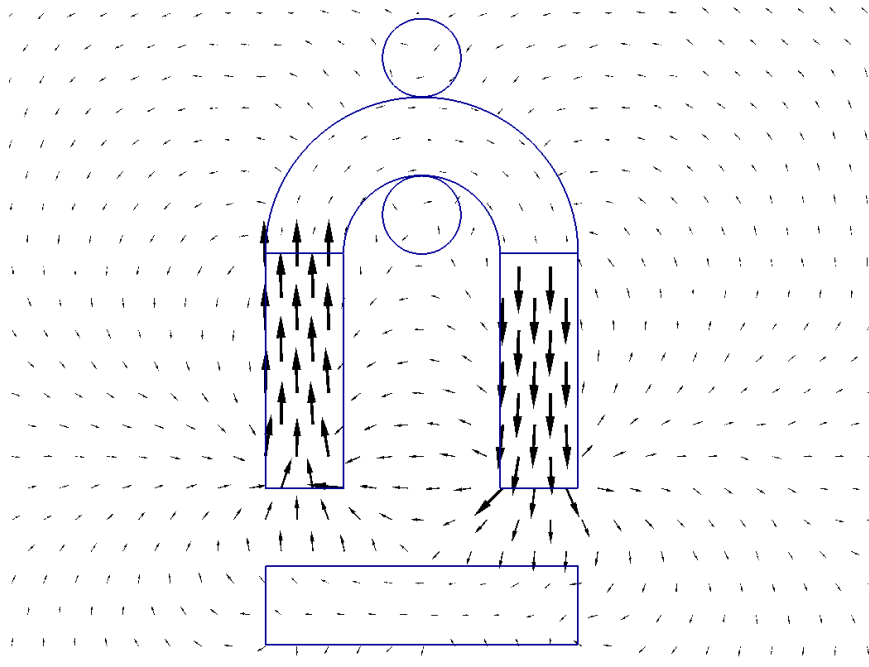


Obr. C.7: Zobrazení siločár pro magnety nesouhlasně

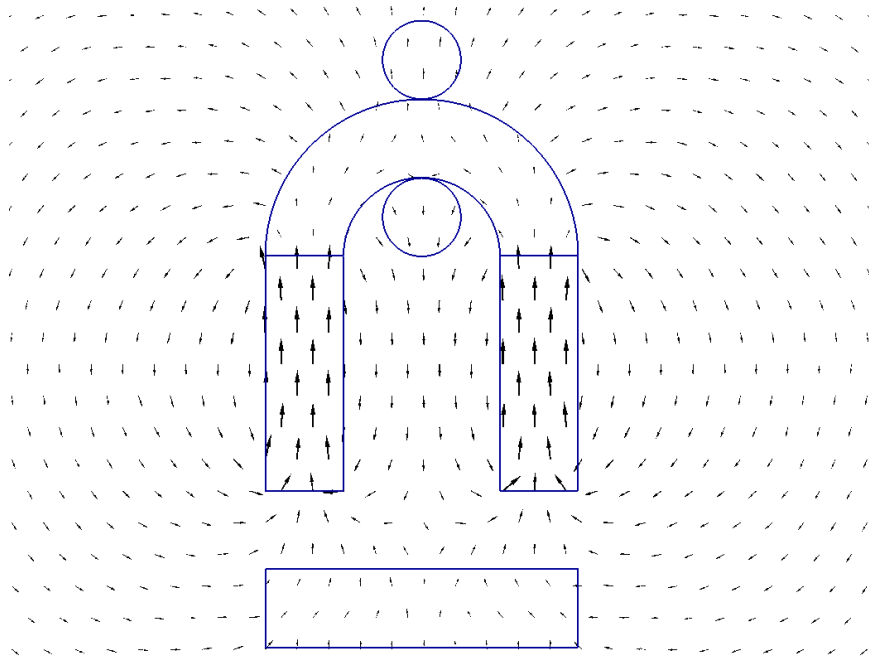


Obr. C.8: Zobrazení siločár pro magnety souhlasně

Výsledné pole je tvořeno vektory, které se sčítají s vektory remanentní magnetické indukce a vektory magnetické indukce vyvolané cívkou. Jejich součet je vidět na obr. C.9 a na obr. C.10. Protože jsou vektory na obrázcích zobrazeny v měřítku, dá se i z těchto obrázků posoudit velikost sečteného pole pro souhlasně a nesouhlasně uložené magnety.



Obr. C.9: Vektory magnetického pole pro magnety nesouhlasně



Obr. C.10: Vektory magnetického pole pro magnety souhlasně

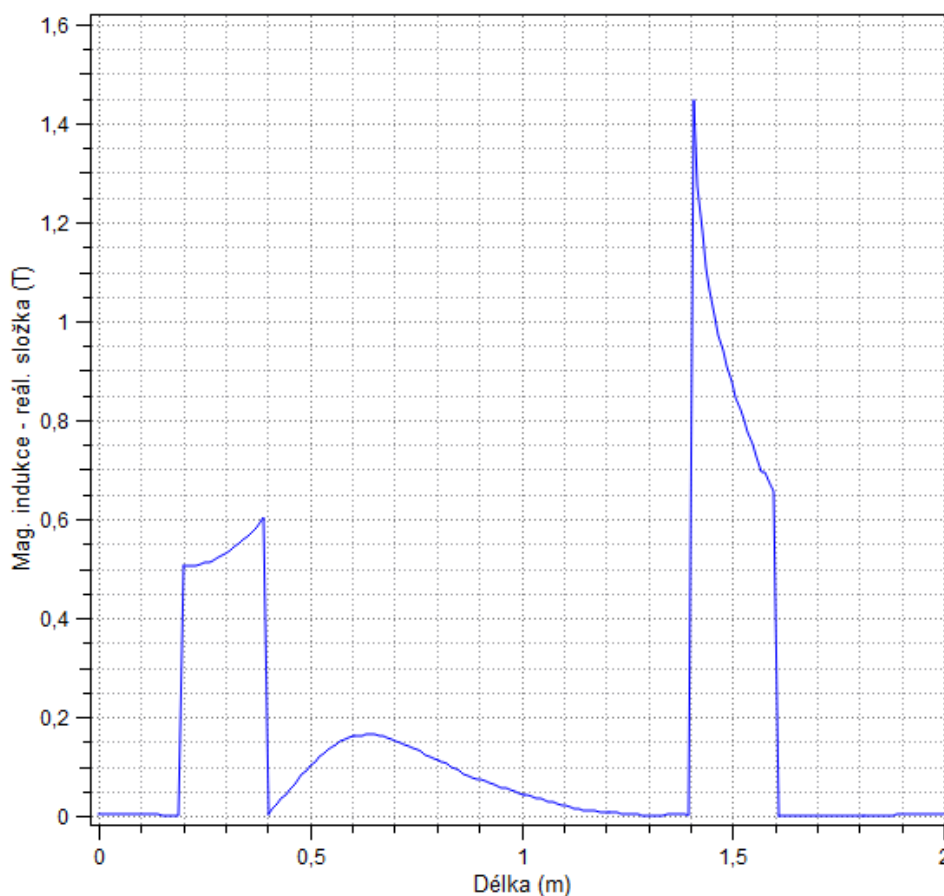
C.4.2 Zobrazení výsledků do grafu

Chceme si zobrazit výsledek ne jako skalární pole, ale jako graf podél zadané přímky. To provedeme tak, že si otevřeme okno *Graf*. V tomto okně je nabídka na zadání souřadnic, které budou tvořit přímku, na které si necháme vybranou veličinu vykreslit. Zadáme souřadnice podle tab. C.7.

Zvolíme veličinu *Magnetická indukce* a její složku necháme celkovou, neboli *Modul*. Z 200 jednotlivých bodů se vytvoří graf a jeho výsledek vidíme na obr. C.11. Zprava vidíme špičku magnetické indukce v železe vloženého mezi vodiče, kde její největší hodnota je u cívky s kladnou hodnotou proudové hustoty. Další výrazný vzestup magnetické indukce je v levé části, přesně na rozměrech tohoto nárůstu magnetické indukce je umístěna kotva elektromagnetu. Graf je vytvořen pro nesouhlasně uložené magnety. Střední hodnota magnetické indukce v kotvě je 0,55 T.

	Počátek	Konec
X[m]	0,4	0,4
Y[m]	-0,6	1,4

Tab. C.7: Souřadnice geometrie pro vykreslení grafu



Obr. C.11: Magnetická indukce podél zadané přímky v tab. C.7

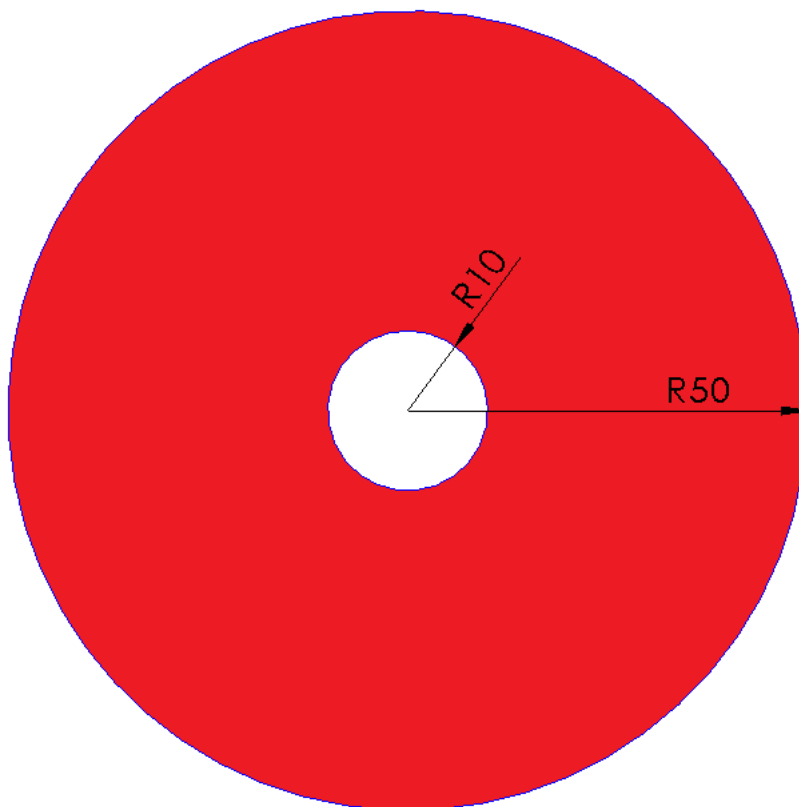
Příloha D

Koaxiální kabel s nedokonalou izolací

D.1 Formulace problému

Je dán koaxiální kabel, viz obr. D.1, který má nedokonalou izolaci o konduktivitě γ . Jeho rozměry jsou: poloměr vnitřního vodiče R_1 a poloměr vnějšího vodiče R_2 . Vnitřní vodič má napětí U a vnější vodič je uzemněn. Izolace vodiče je zvýrazněna červenou barvou na obr. D.1.

$$R_1 = 1 \text{ cm} \quad R_2 = 5 \text{ cm} \quad U = 10 \text{ V} \quad \gamma = 50 \cdot 10^{-7} \frac{\text{S}}{\text{m}}$$



Obr. D.1: 2D model koaxiálního kabelu

D.2 Definice modelu

Problém bude řešen s využitím symetrie v ose x . Bude tedy řešen jako dvoudimenzionální osově symetrická úloha v kartézských souřadnicích.

D.2.1 Rozdělení oblastí

Bude potřeba jen jedna oblast. A to oblast, kde se nachází dielektrikum izolace.

D.2.2 Okrajové podmínky

Budou potřeba celkem tři okrajové podmínky. První pro vnitřní vodič, druhá pro vnější vodič a poslední pro osu symetrie.

D.3 Vytvoření modelu

D.3.1 Nastavení problému

Nejdříve musíme nastavit parametry problému. Pomocí okna *Nastavení problému*, kde nastavíme všechny parametry podle tab. D.1.

Typ problému	Kartézský
Fyzikální pole	Elektrické proudové pole
Druh analýzy	Ustálený stav
Typ sítě	Trojúhelníková
Počet zjemnění	0
Řád polynomu	1
Typ adaptivity	hp-adaptivita
Počet adaptivních kroků	10
Tolerance adaptivity [%]	1

Tab. D.1: Parametry nastavení problému koaxiálního kabelu

D.3.2 Vytvoření konstant

Abychom nemuseli všude zapisovat jednotlivé hodnoty, tak si pro ně vytvoříme konstanty. Pomocí okna *Nastavení problému / Definice proměnných* a zapíšeme konstanty podle tab. D.2.

U	10
γ	$50e^{-7}$

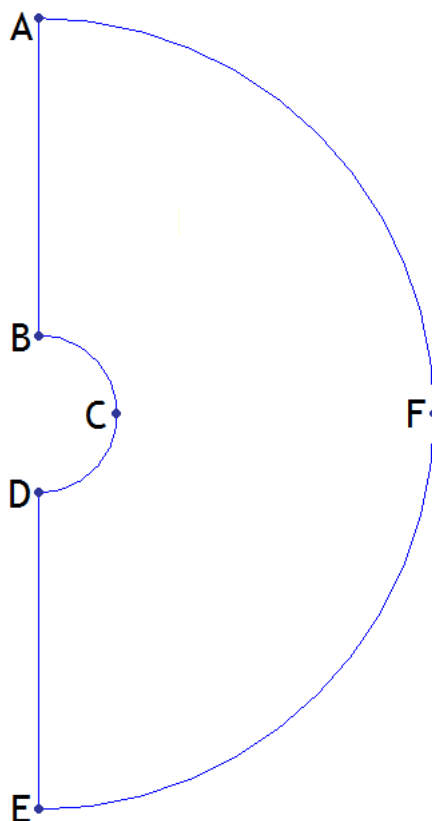
Tab. D.2: Hodnoty konstant

D.3.3 Vytvoření geometrie

Jako první musíme vytvořit jednotlivé uzly, přepneme se do režimu *Práce s uzly* a vytvoříme všechny potřebné uzly (používáme desetinnou tečku) podle tab. D.3.

Osa	A	B	C	D	E	F
R [m]	0	0	0,01	0	0	0,05
Z [m]	0,05	0,01	0	-0,01	-0,05	0

Tab. D.3: Souřadnice jednotlivých uzlů



Obr. D.2: 2D model koaxiálního kabelu s body pro vytvoření geometrie

Následně se přepneme do režimu *Práce s hranami* a jednotlivé uzly propojíme tak, aby vytvořili potřebný model - viz obr. D.2. Hrany BC CD tvoří obrys vnitřního vodiče, hrany EF FA tvoří obrys vnějšího vodiče a nakonec hrany AB DE tvoří osu symetrie. Hrany, které mají tvořit půlkruh zaoblíme tak, že jim nastavíme úhel 90° .

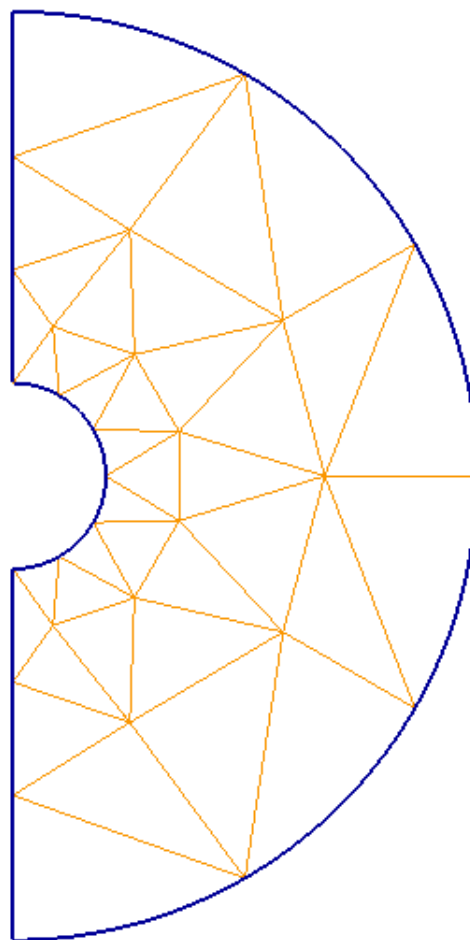
D.3.4 Doplnění fyzikálních podmínek

Nyní musíme vytvořit okrajové podmínky. První okrajovou podmínku pro vnitřní vodič, kde její hodnota je potenciál napětí U . Další okrajová podmínka pro vnější vodič má opět potenciál napětí a tentokrát hodnotu nula, protože je vodič uzemněn. Třetí okrajovou podmínku, která bude přiřazena hraně symetrie, respektuje fyzikální vlastnosti osy symetrie, vybereme pro ní povrchovou hustotu náboje o hodnotě nula.

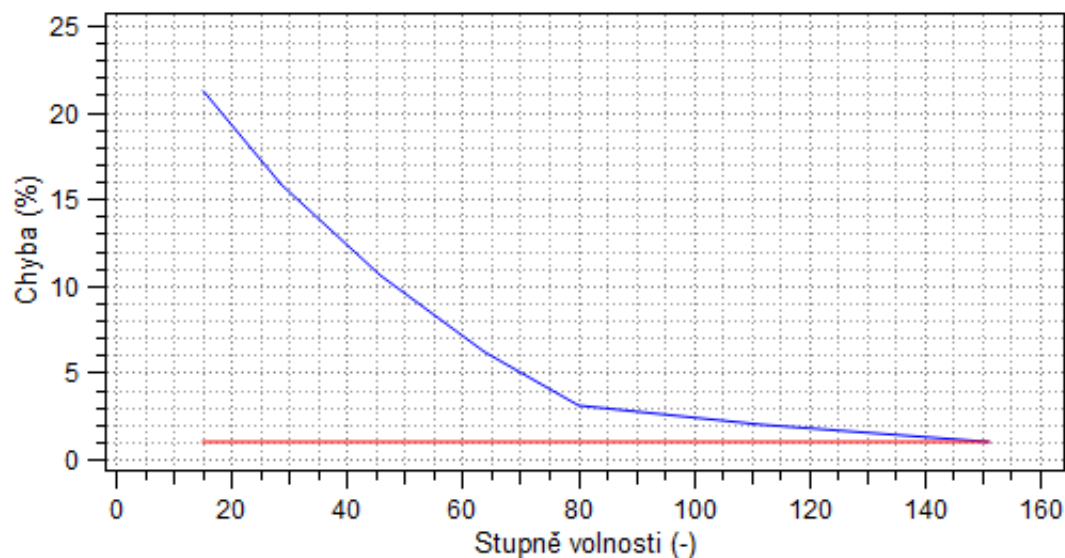
Poslední, co musíme udělat, je značka oblastí. Vytvoříme ji a s hodnotou konduktivity izolace γ a vložíme kamkoliv do oblasti izolace.

D.3.5 Diskretizace a řešení

Při řešení se definiční oblast dělí na menší elementy podle vybrané sítě a dle parametrů použité adaptivity. My jsme vybrali hp-adaptivitu s relativní přesností jednoho procenta. Program tedy zvyšuje počet stupňů volnosti tak, než dosáhne pod tuto zadanou přesnost - viz obr. D.4. Rozložení počáteční a řešené sítě je stejné a výsledek vidíme na obr. D.3.



Obr. D.3: Řešená a počáteční síť



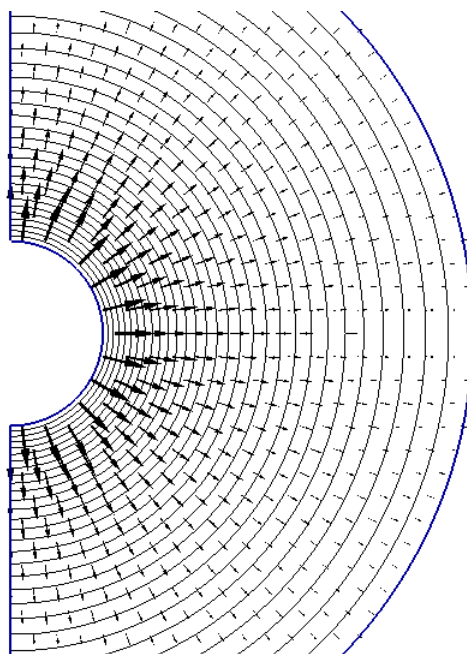
Obr. D.4: Závislost relativní chyby na počtu stupňů volnosti

D.4 Diskuze výsledků

D.4.1 Rozložení výsledného pole

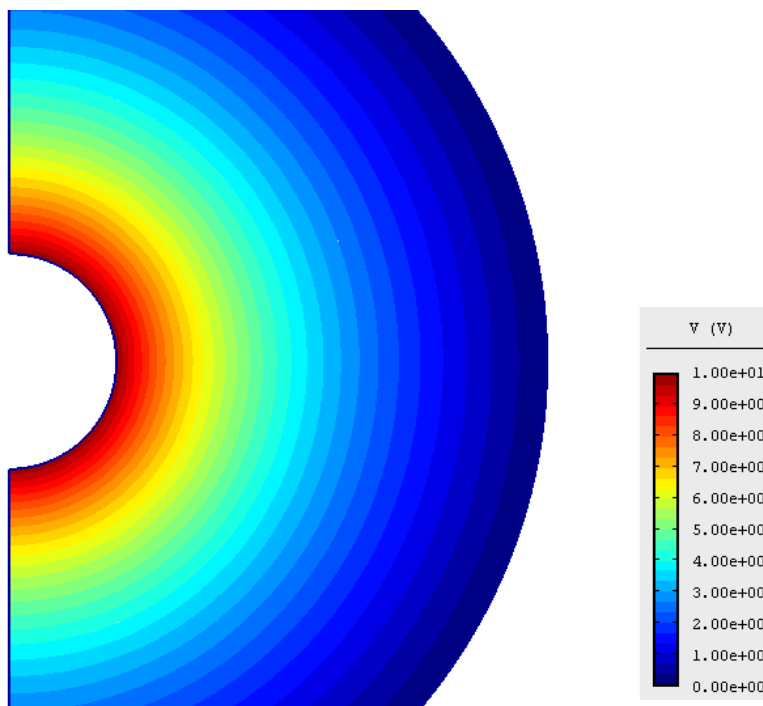
Pro vyřešený problém program umožňuje zobrazit různé veličiny a jejich rozložení v definiční oblasti a také vypočítá různé hodnoty v bodě, v ploše nebo v objemu.

Pro zobrazení vektorů a ekvipotenciál je zaškrtneme a zobrazíme bez skalárního pole pomocí *2D Bez zobrazení*, abychom viděli jen je. V další záložce *Postprocesor* jim můžeme nastavit různé parametry, jako počet a velikost kontur i vektorů. Na obr. D.5 je zobrazen výsledek pro 30 kontur se šířkou 1,00 a 60 vektorů v měřítku 1,20.



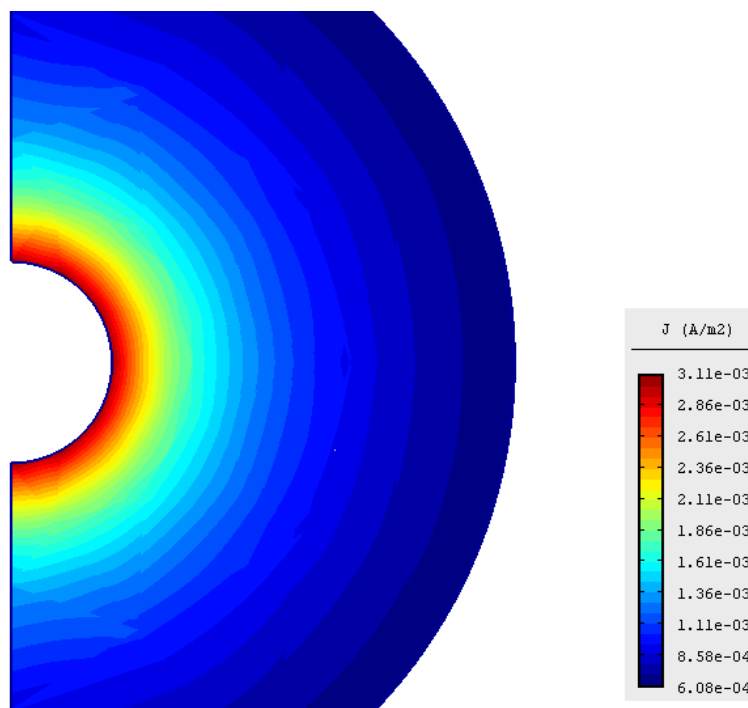
Obr. D.5: Zobrazení ekvipotenciál a vektorů pro proudovou hustotu

Dále si můžeme zobrazit skalární pole vybrané veličiny. Například pro veličinu *Skalární el.potenciál* je výsledek zobrazen na obr. D.6.



Obr. D.6: Skalární rozložení elektrického potenciálu

Pak nás zajímá rozložení proudové hustoty v izolaci koaxiálního kabelu, proto tuto veličinu vybereme a její skalární rozložení je zobrazeno na obr. D.7.



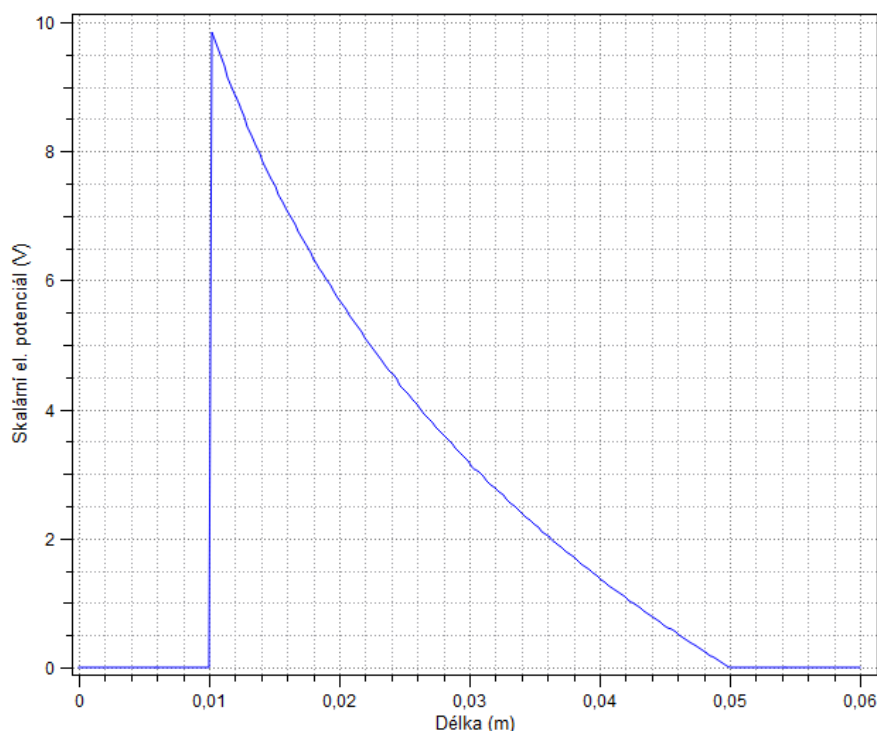
Obr. D.7: Rozložení proudové hustoty v izolaci koaxiálního kabelu

D.4.2 Zobrazení výsledku pomocí grafu

Další možností zobrazení výsledků je zobrazení vybrané veličiny do grafu podél vybrané přímky. Otevřeme nabídku *Graf*. V tomto okně si můžeme nastavit počáteční a konečné souřadnice, případně i úhel, po které se vybraná veličina zobrazí do grafu. My postupně zobrazíme tři veličiny, které nás zajímají. Za prvé skalární elektrický potenciál (obr. D.8), za druhé intenzitu elektrického pole (obr. D.9) a nakonec proudovou hustotu (obr. D.10). Všechny veličiny pro stejné rozměry přímky, které jsou uvedeny v tab. D.4. Vybrané souřadnice tvoří přímku, která vede ze středu vnitřního vodiče ve směru osy y , až za hranici vnějšího vodiče.

	X [m]	Y [m]
Počátek	0	0
Konec	0,06	0

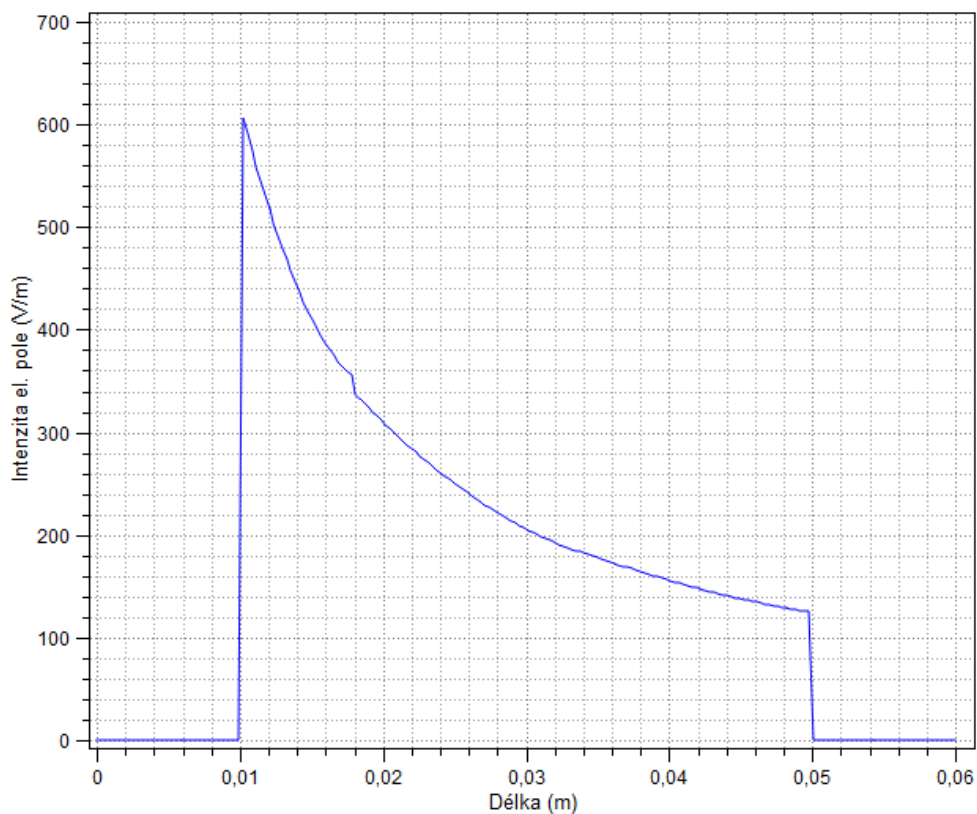
Tab. D.4: Hodnoty souřadnic



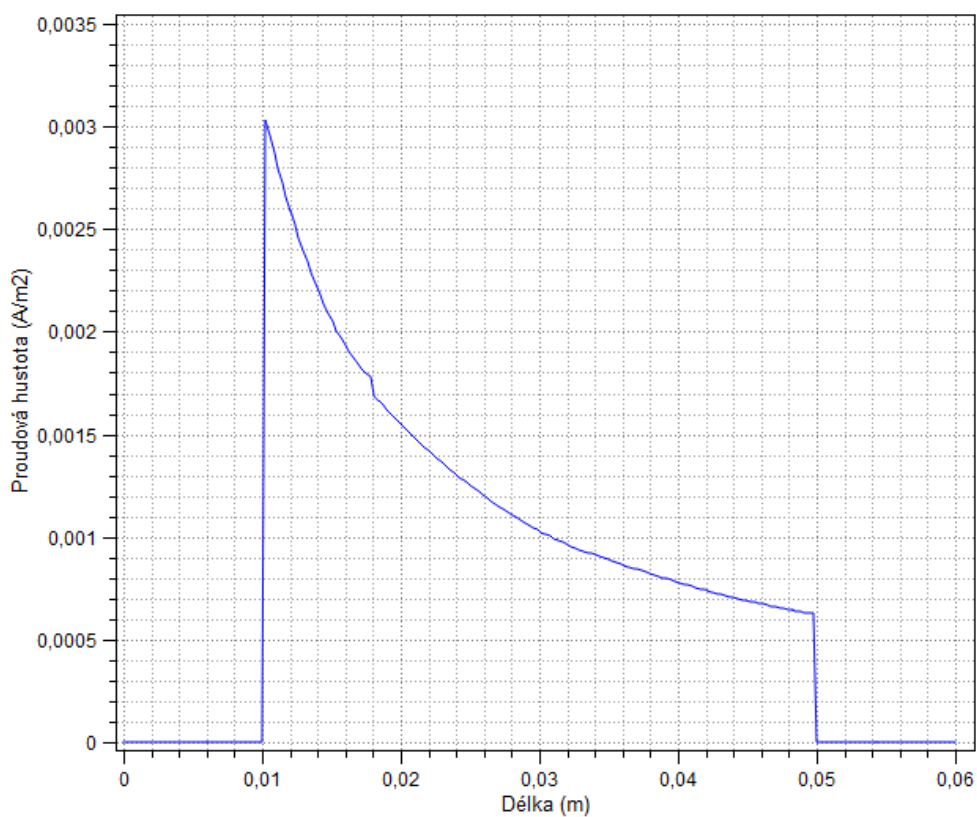
Obr. D.8: Graf skalárního elektrického potenciálu podél osy r

D.4.3 Ztráty v izolaci

Program nám spočte velikost ztrát v ploše izolace na jeden metr délky. Přepneme se do režimu *Objemové integrály* a po označení plochy izolace nám program v poslední záložce zobrazí hodnotu ztrát, kterou vynásobíme dvěma, protože řešíme jen polovinu modelu. V našem příkladě vyšly ztráty po vynásobení $P = 19,51 \cdot 10^{-4}$ W.



Obr. D.9: Graf intenzity elektrického pole podél osy r



Obr. D.10: Graf proudové hustoty podél osy r

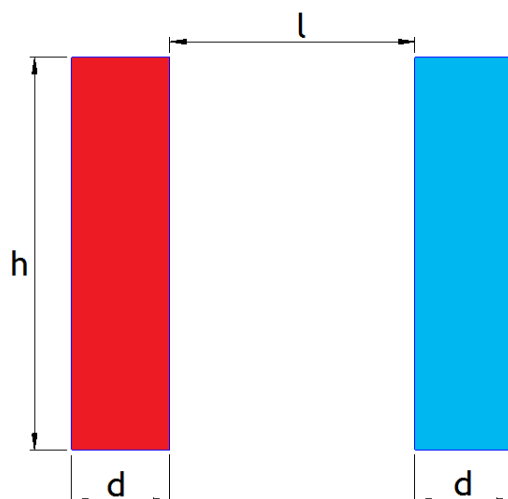
Příloha E

Dvou vodičové pásové vedení

E.1 Formulace problému

Je dáno dvou vodičové pásové vedení, které je tvořeno dvěma stejnými vodiči tvaru obdélníku viz obr. E.1. Rozměry obdélníkového vodiče jsou výška h a šířka d a jejich vzdálenost je l . Vodiče jsou vyrobeny z materiálu, který má relativní permeabilitu μ_{r1} , jsou protékány proudem s proudovou hustotou J_{ext} a mají elektrickou vodivost γ . Celé vedení je obklopeno vzduchem s relativní permeabilitou μ_{r0} . Levý vodič je zobrazen červenou barvou a má kladnou hodnotu proudové hustoty, pravý vodič je zobrazen modrou barvou a má zápornou proudovou hustotu viz obr. E.1.

$$h = 0,08 \text{ m} \quad d = 0,02 \text{ m} \quad l = 0,1 \text{ m} \quad \mu_{r0} = 1 \quad \mu_{r1} = 10 \quad J_{ext} = 15 \cdot 10^6 \frac{\text{A}}{\text{m}^2} \quad \gamma = 6 \cdot 10^7 \frac{\text{S}}{\text{m}}$$



Obr. E.1: 2D model dvou vodičového pásového vedení

E.2 Definice modelu

E.2.1 Rozdělení oblastí

Definiční oblast bude rozdělena na tři oblasti s rozdílnými parametry. Budou to okolní vzduch, první vodič a druhý vodič. Pro vodiče máme zadanou proudovou hustotu J_{ext} a pro vzduch relativní permeabilitu μ_{r0} .

E.2.2 Okrajové podmínky

Jediná okrajová podmínka bude potřeba a to okrajová podmínka pro fiktivní hranici nekonečna. Tato okrajová podmínka bude mít jako typ okrajové podmínky zadávaný parametr *Vektorový mag. potenciál* o velikosti nula, protože tento parametr respektuje fyzikální rozložení magnetického pole v nekonečnu.

E.3 Vytvoření modelu

Model bude vytvořen tak, že nejprve se umístí jednotlivé body. Následovat bude spojení bodů hranami a vložení značek oblastí do jednotlivých oblastí. Nakonec se přiřadí okrajové podmínky správným hranám a materiály značkám oblastí.

E.3.1 Nastavení problému

Nejdříve je potřeba správně připravit parametry řešení. Proto otevřeme okno *Nastavení problému* a nastavíme veškeré parametry podle tab. E.1.

Typ problému	Kartézský
Fyzikální pole	Magnetické pole
Druh analýzy	Harmonická analýza
Frekvence [Hz]	50
Typ sítě	Trojúhelníková
Počet zjemnění	0
Řád polynomu	1
Typ adaptivity	hp-adaptivita
Počet adaptivních kroků	40
Tolerance adaptivity [%]	1

Tab. E.1: Parametry nastavení problému

E.3.2 Vytvoření konstant

Pro μ_{r0} , μ_{r1} , γ a J_{ext} budou vytvořeny konstanty tak, že se v okně *Nastavení problému* přepneme do druhé záložky *Definice proměnných* a zapíšeme konstanty podle tab. E.2.

μ_{r0}	1
μ_{r1}	10
J_{ext}	15e6
γ	6e7

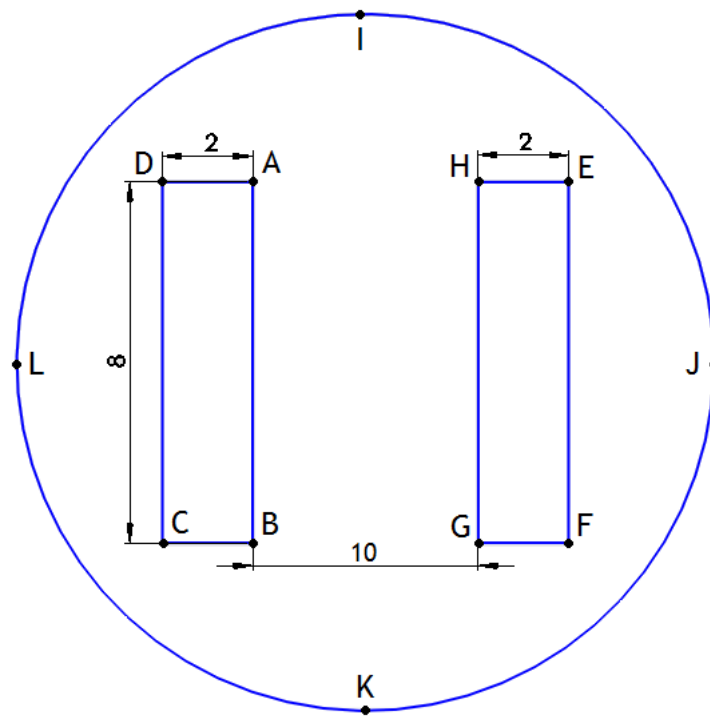
Tab. E.2: Hodnoty konstant

E.3.3 Vytvoření geometrie

Geometrii začneme vytvářet tak, že se přepneme do režimu *Práce s uzly* a vytvoříme všechny potřebné uzly podle tab. E.3. Při zadávání souřadnic používáme desetinnou tečku.

Osa	A	B	C	D	E	F	G	H	I	J	K	L
X [m]	0	0	-0,02	-0,02	0,12	0,12	0,1	0,1	0,05	0,33	0,05	-0,25
Y [m]	0	-0,08	-0,08	0	0,0	-0,08	-0,08	0	0,25	-0,05	-0,33	-0,05

Tab. E.3: Souřadnice uzlů dvou vodičového pásového vedení



Obr. E.2: 2D model dvou vodičového pásového vedení (rozměry v cm)

Po vytvoření všech uzlů je musíme spojit podle obr. E.2, aby vytvořili obrys potřebného modelu. Hrany AB BC CD DA tvoří první vodič, hrany EF FG GH HE tvoří druhý vodič a fiktivní hranice nekonečna je tvořena hranami IJ JK KL LI. Pouze poslední hrany fiktivní hranice nekonečna musíme ještě nastavit tak, aby vytvořily kružnici. To provede kliknutím na ně a zapsáním 90° jako velikost jejich úhlu.

Následovně zbývá přidat značky oblasti, abychom mohli rozdělit oblasti vodičů a okolního vzduchu. Proto vytvoříme tři značky oblasti a přidáme je do každé jednotlivé oblasti.

E.3.4 Doplnění fyzikálních podmínek

Dalším úkolem je vytvořit fyzikální podmínku pro fiktivní hranici nekonečna. To provedeme a vybereme u ní *Vektorový mag. potenciál* a necháme hodnotu na nule. Poté označíme příslušné hrany modelu, které tvoří tuto hranici fiktivního nekonečna a přidáme jim vytvořenou okrajovou podmínku.

Nakonec musíme vytvořit tři materiály pro jednotlivé oblasti. K tomu můžeme využít vytvořených konstant a bez zadávání stejných hodnot jen zapíšeme název vytvořené konstanty viz tab. E.4. Tímto způsobem vytvoříme všechny tři konstanty pro oba vodiče i okolní vzduch.

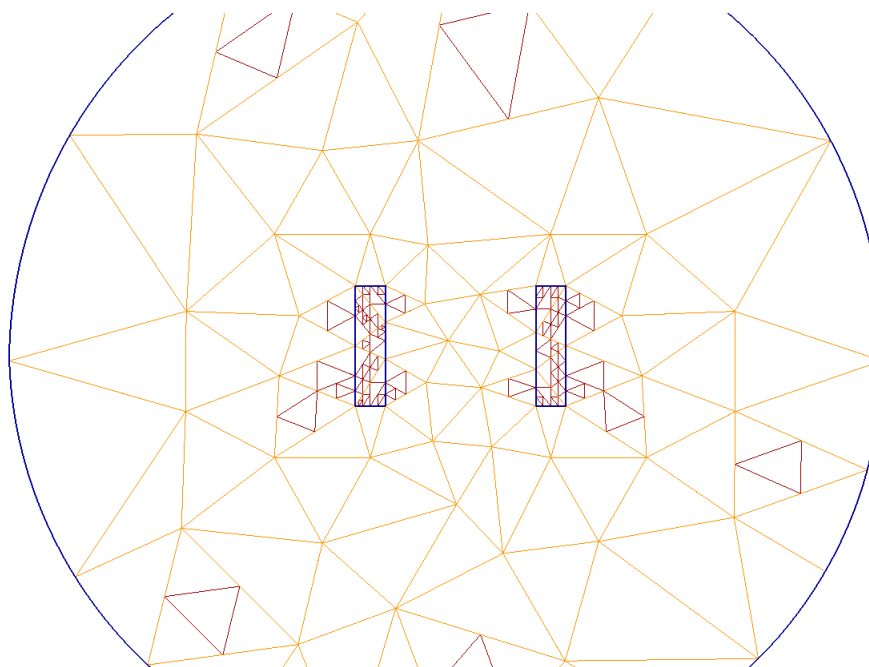
	Vzduch	Vodič 1	Vodič 2
Relativní permeabilita μ_r [-]	μ_{r0}	μ_{r1}	
Proudová hustota J_{ext} [$\frac{A}{m^2}$]	0	J_{ext}	$-J_{ext}$
Elektrická vodivost [$\frac{S}{m}$]	0	γ	

Tab. E.4: Zadané hodnoty materiálů pro vodiče a vzduch

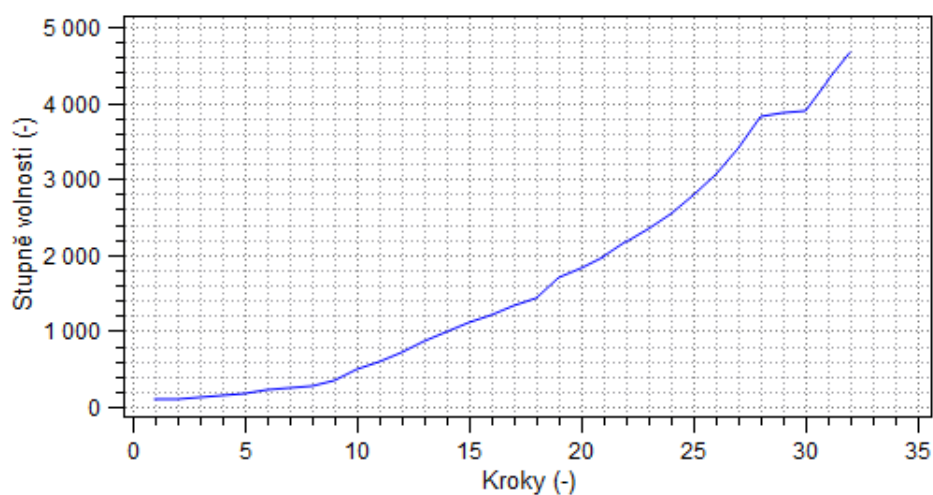
E.3.5 Diskretizace a řešení

Při řešení se definiční oblast rozdělí na menší části a poté je problém vyřešen. Při použití hp-adaptivity se oblast dělí a dělí, dokud nedosáhne požadované přesnosti zadání, pro nás jednoho procenta. Na obr. E.3 můžeme vidět počáteční a řešenou síť po třiceti dvou krocích hp-adaptivity.

Průběh řešení můžeme vidět na obr. E.4, který nám zobrazuje, jak stoupají stupně volnosti v jednotlivých krocích hp-adaptivity (tento graf získáme na záložce *Adapt.DOFs* v okně *Průběh...*).



Obr. E.3: Diskretizovaná definiční oblast (počáteční síť světlou barvou a řešená síť tmavší barvou)



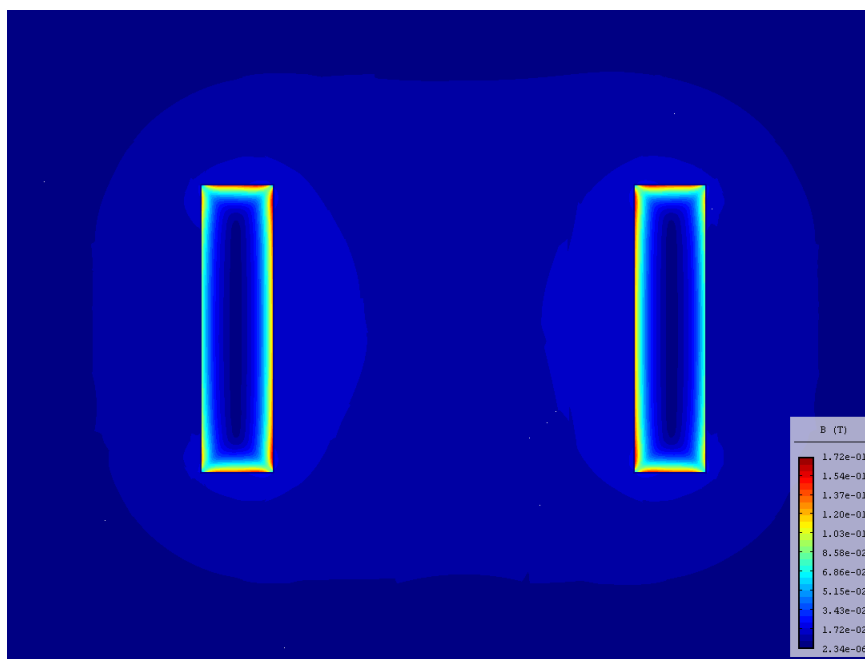
Obr. E.4: Závislost stupňů volnosti na počtu kroků hp-adaptivity

E.4 Diskuze výsledků

E.4.1 Rozložení výsledného pole

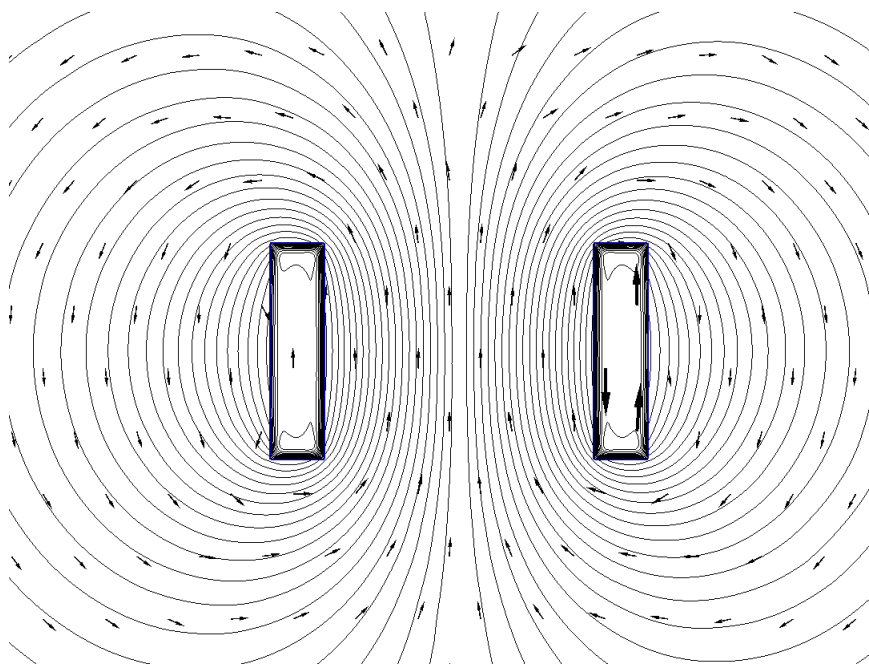
Po vyřešení problému si můžeme jako jeden z možných výsledků nechat programem vykreslit barevné rozložení skalárního pole vybrané veličiny. Například pro zobrazení rozložení magnetické indukce v řešené oblasti vybereme *Mag. indukce* a výsledek vidíme na obr. E.5.

Protože jsme zadali parametry tak, že jedním vodičem teče proud pryč a druhým se vrací, při zobrazení siločár a vektorů, vidíme správnost řešení ve směru vykreslených



Obr. E.5: Rozložení skalárního pole magnetické indukce

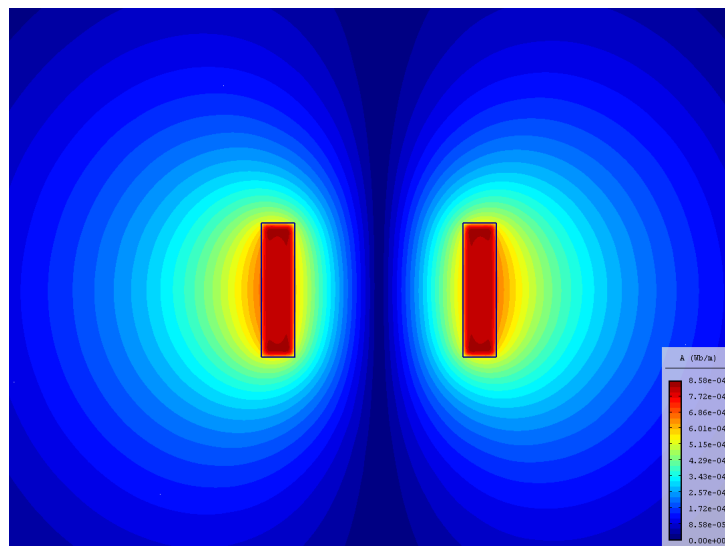
vektorů a rozložení siločár. Směr vektorů dodržuje takzvané pravidlo pravé ruky a siločáry prochází mezi vodiči opačné polarity viz obr. E.6, kde bylo zvoleno v záložce *Postprocesor* 30 kontur se šířkou 1,00 a 50 vektorů v proporčním měřítku 1,00.



Obr. E.6: Zobrazení siločár a vektorů

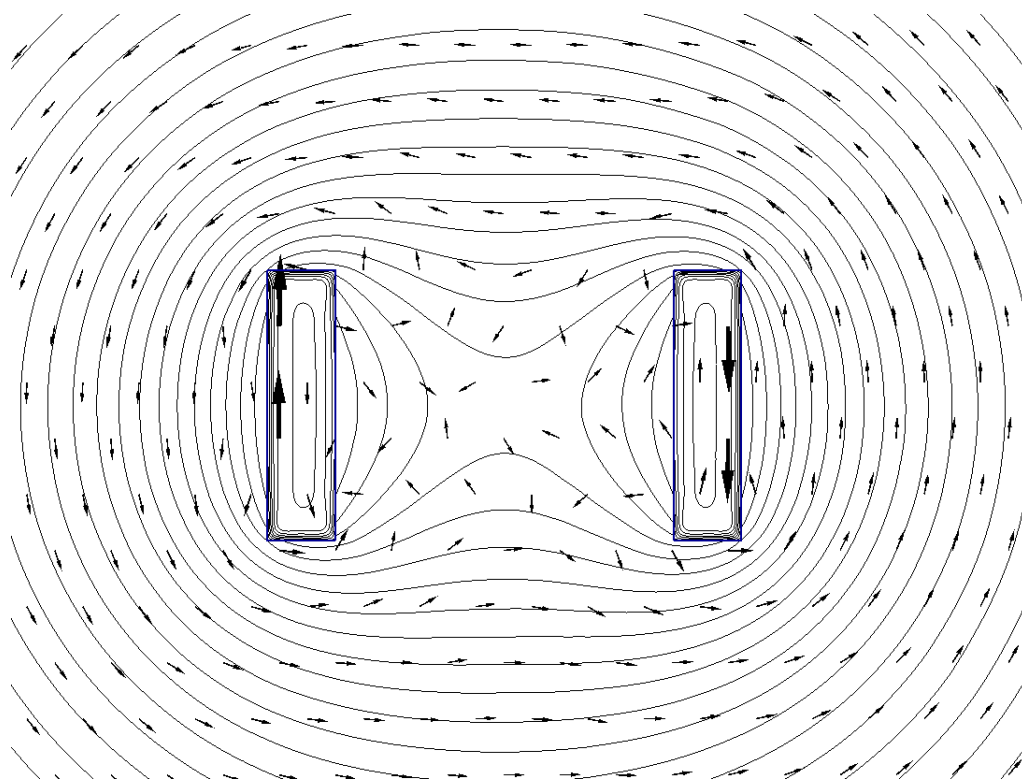
Nakonec si vybereme pro zobrazení *Vektorový mag. potenciál* a na obr. E.7 je zobrazeno výsledné pole, ze kterého vidíme, že maximální hodnota vektorového magnetického potenciálu je ve středu jednotlivých vodičů.

Po změně vedení tak, že nyní oba vodiče povedou proud ve stejném směru (otočíme



Obr. E.7: Rozložení skalárního pole vektorového magnetického potenciálu

polaritu proudové hustoty u pravého vodiče), dojde ke změně v rozložení vektorů i siločár, stále však platí pravidlo pravé ruky a vektory směřují ve směru prstů pravé ruky, pokud palec směřuje ve směru toku proudu vodičem, v prostoru mezi vodiči mají jednotlivé vektory od každého vodiče nyní rozdílný směr a jejich vzájemné působení je vidět na obr. E.8.



Obr. E.8: Zobrazení siločár a vektorů

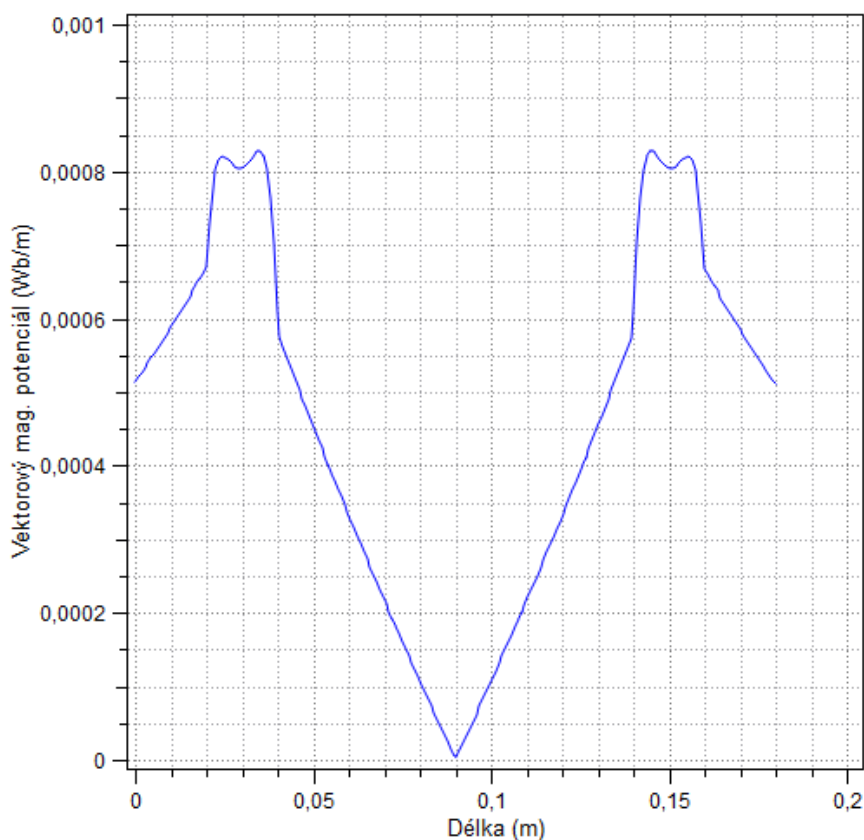
E.4.2 Zobrazení výsledků do grafu

Program nám umožní zobrazení vybrané veličiny do grafu podél zadané přímky. Otevřeme okno a dostaneme se do nabídky *Graf*, kde zadáme souřadnice geometrie podle tab. E.5 a tím si vybereme přímku v ose vodičů.

	Počátek	Konec
X[m]	-0,04	0,14
Y[m]	-0,04	-0,04

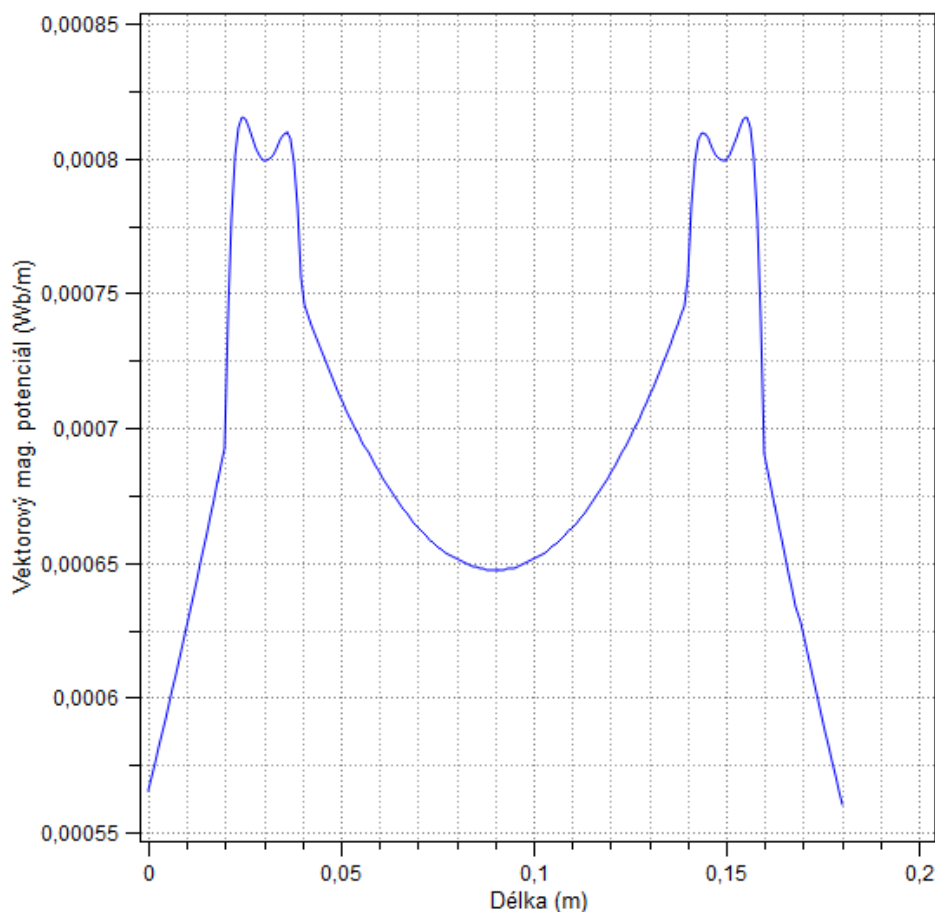
Tab. E.5: Souřadnice geometrie pro vykreslení grafu

Vybereme veličinu *Vektorový mag. potenciál* a zvolíme 200 bodů, aby byl graf sestaven z tohoto počtu jednotlivých bodů a byl dostatečně hladký. Výsledek je zobrazen na obr. E.9, ze kterého je opět vidět, že nejvyšší hodnoty potenciálu jsou na okrajích vodičů a na obě strany klesají (což můžeme porovnat se stejným výsledkem ve skalárním rozložení pole vektorového magnetického potenciálu viz obr. E.7). Protože proud protékající vodiči je stejný a má jen opačný směr, tak je pokles potenciálu symetrický pro oba vodiče. Graf si můžeme uložit jako obrázek nebo exportovat jako data pro pozdější použití v dalších výpočetních softwarech.



Obr. E.9: Vektorový magnetický potenciál pro rozdílný směr proudu tekoucí vodiči

Pro vodiče, kterými by proud tekł stejným směrem, by byl výsledek trochu jiný, protože, jak jsme viděli na obr. E.8, vektory mezi vodiči nemají stejný směr a mezi vodiči je tedy magnetický potenciál nenulový. Graf je zobrazen na viz obr. E.10 opět pro stejné body přímky viz tab. E.5.

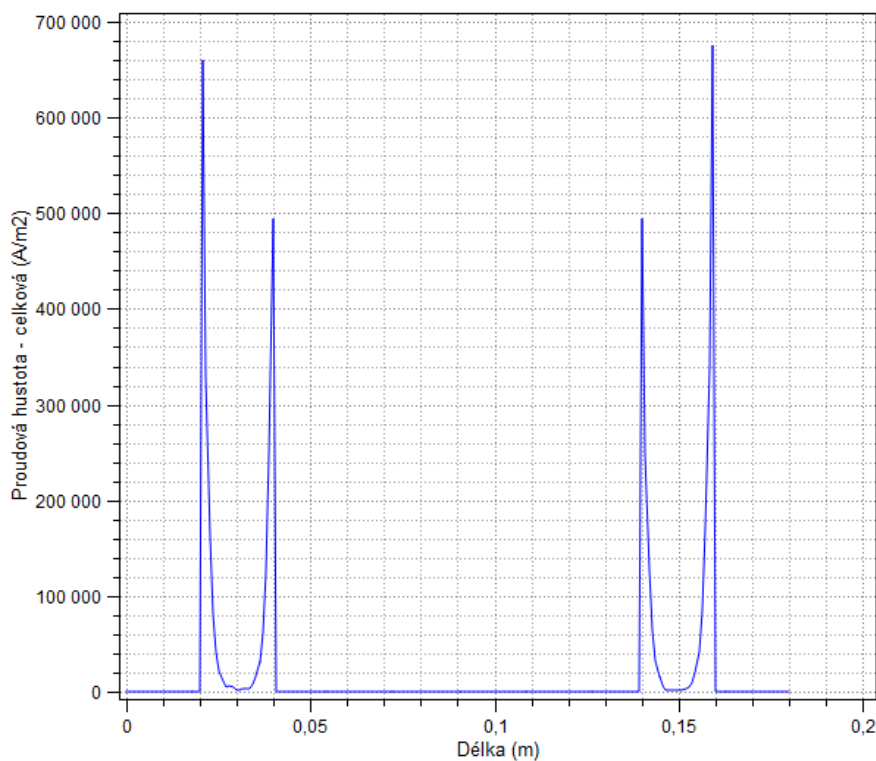


Obr. E.10: Vektorový magnetický potenciál pro stejný směr proudu tekoucí vodiči

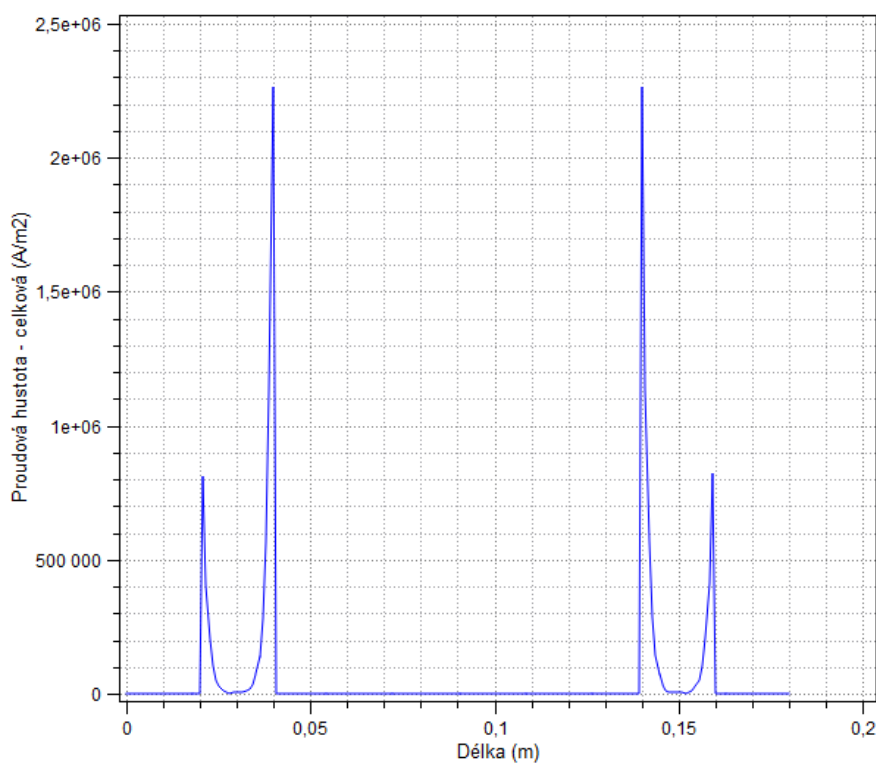
E.4.3 Jev blízkosti a skin efekt

Z grafického zobrazení celkové proudové hustoty můžeme pozorovat jev blízkosti a skin efekt. Proto si tuto veličinu vybereme a zobrazíme podél přímky dle tab. E.5. Výsledek pro stejný směr proudu je zobrazen na obr. E.11. Pro rozdílný směr proudu tekoucí každým vodičem je výsledek na obr. E.12. Z grafů vidíme, že proudová hustota ve středu vodiče je nejmenší a ke krajům se zvětšuje, tento jev se nazývá skin efekt a způsobuje, že většina proudu teče jen na jeho povrchu. Na obrázcích jsou ještě velikosti proudové hustoty na krajích vodiče rozdílné, to je způsobeno jevem blízkosti. Protože jsou vodiče relativně blízko u sebe, působí na sebe navzájem. Pro souhlasný směr proudu tekoucí vodiči se proudová hustota na přilehlých krajích vodičů zmenšuje - viz obr. E.11. A naopak pro

rozdílný směr proudu tekoucí vodiči se proudová hustota na těchto přilehlých krajích vodičů zvětšuje - viz obr. E.12.



Obr. E.11: Celková proudová hustota pro stejný směr proudu tekoucí vodiči



Obr. E.12: Celková proudová hustota pro rozdílný směr proudu tekoucí vodiči

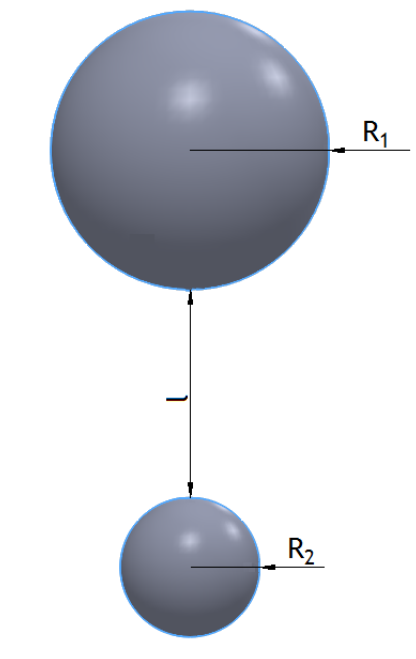
Příloha F

Kulové jiskřiště

F.1 Formulace problému

Je dáno kulové jiskřiště, které je umístěno ve vzduchu viz obr. F.1. Jiskřiště je tvořeno dvěma kulovými elektrodami, horní elektroda má poloměr R_1 , spodní elektroda má poloměr R_2 a vzdálenost mezi jejich okraji je l . Napětí mezi elektrodami je U a spodní elektroda je uzemněna. Relativní permitivita vzduchu je rovna ϵ_{r0} a nakonec zadaná elektrická pevnost vzduchu je E_p .

$$R_1 = 10 \text{ cm} \quad R_2 = 5 \text{ cm} \quad l = 50 \text{ cm} \quad U = 100 \text{ kV} \quad \epsilon_{r0} = 1 \quad E_p = 30 \frac{\text{kV}}{\text{cm}}$$



Obr. F.1: 3D model kulového jiskřiště (l není v měřítku)

F.2 Definice modelu

Problém bude řešen jako 2D osově symetrická úloha. Bude využita symetrie v ose z .

F.2.1 Rozdělení oblastí

Oblast bude rozdělena na tři různé: horní elektroda, spodní elektroda a vzduch. Uvnitř elektrod nás rozložení pole nezajímá, a proto si vystačíme s jedinou definovanou oblastí a tou bude okolní vzduch.

F.2.2 Okrajové podmínky

Budou potřeba tři různé okrajové podmínky. První pro obrys horní elektrody, druhá pro obrys spodní elektrody a nakonec třetí pro fiktivní hranici nekonečna a pro osu symetrie.

F.3 Vytvoření modelu

F.3.1 Nastavení problému

Zprvė nastavíme parametry modelu, otevřeme okno *Nastavení problému* a veškeré parametry nastavíme podle tab. F.1.

Typ problému	Osově symetrický
Fyzikální pole	Elektrostatické pole
Druh analýzy	Ustálený stav
Typ sítě	Trojúhelníková
Počet zjemnění	0
Řád polynomu	1
Typ adaptivity	hp-adaptivita
Počet adaptivních kroků	15
Tolerance adaptivity [%]	1

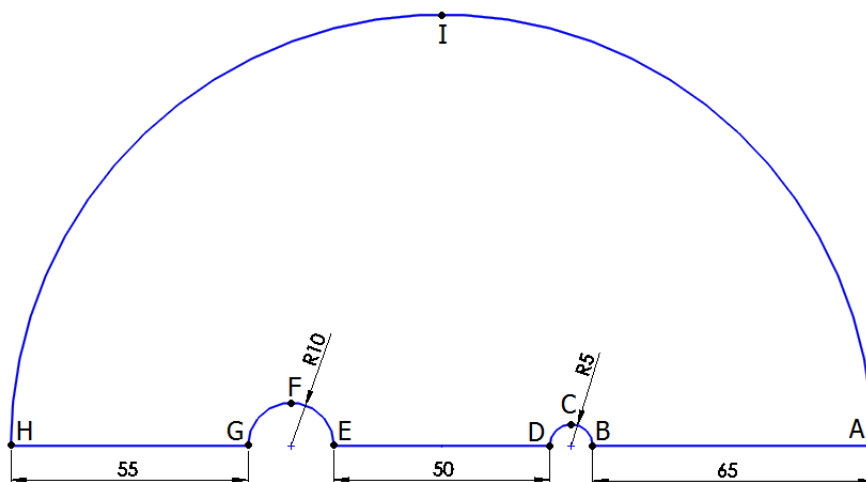
Tab. F.1: Parametry nastavení problému

F.3.2 Vytvoření konstant

Vytvoříme si jednotlivé konstanty v okně *Nastavení problému / Definice proměnných* a zapíšeme konstanty podle tab. F.2.

U	100000
ϵ_{r0}	1

Tab. F.2: Hodnoty konstant



Obr. F.2: 2D model jiskřiště (rozměry v centimetrech)

F.3.3 Vytvoření geometrie

Začneme vytvořením jednotlivých uzlů, přepneme se do režimu *Práce s uzly* a vytvoříme všechny potřebné uzly (používáme desetinou tečku) podle tab. F.3.

Osa	A	B	C	D	E	F	G	H	I
R [m]	0	0	0,05	0	0	0,1	0	0	1
Z [m]	-1	-0,35	-0,3	-0,25	0,25	0,35	0,45	1	0

Tab. F.3: Souřadnice jednotlivých uzlů

Poté se přepneme do režimu *Práce s hranami* a jednotlivé uzly propojíme tak, aby vytvořili potřebný model - viz obr. F.2. Hrany BC CD tvoří obrys spodní elektrody, hrany EF FG tvoří obrys horní elektrody a nakonec hrany AB DE GH tvoří osu symetrie a hrany HI IA fiktivní hranici nekonečna. Potřebné hrany zaoblíme tak, že jim nastavíme úhel 90°.

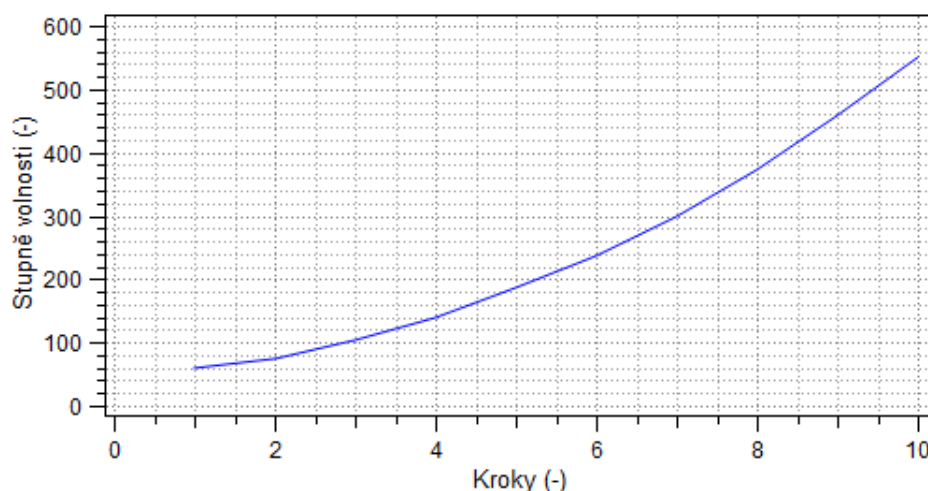
F.3.4 Doplnění fyzikálních podmínek

Vytvoříme první okrajovou podmínku pro horní elektrodu a nastavíme jí potenciál napětí U pomocí vytvořené konstanty. Poté vytvoříme okrajovou podmínku pro spodní elektrodu a jí nastavíme potenciál napětí na nulu. Poslední okrajovou podmínku s názvem *Neumannova*, která respektuje osu symetrie a fyzikální vlastnosti fiktivní hranice, vytvoříme a vybereme u ní povrchovou hustotu náboje a necháme ho na nule a přiřadíme ji k hranám AB DE GH a HI IA.

Dále umístíme značku oblasti do oblasti vzduchu v režimu *Práce se značkami oblasti* a nastavíme jí parametry podle zadání, to znamená relativní permitivitu pomocí vytvořené konstanty ϵ_{r0} .

F.3.5 Diskretizace a řešení

Pro řešení se definiční oblast musí rozdělit na jednotlivé elementy, při použití hp-adaptivity se postupně zvětšuje stupeň volnosti. Po devíti krocích hp-adaptivity chyba klesne na 0,682 % a výpočet skončí s 551 stupni volnosti - viz obr. F.3 (ten získáme v záložce *Adapt.DOFs* po kliknutí na ikonku *Vyřešit problém*).



Obr. F.3: Závislost stupňů volnosti na počtu kroků výpočtu

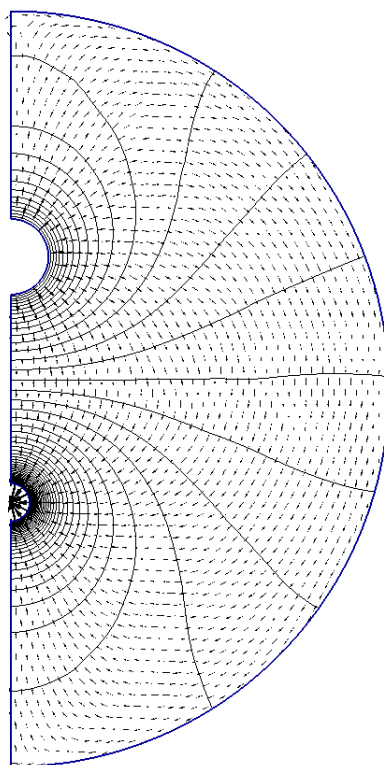
F.4 Diskuze výsledků

F.4.1 Rozložení výsledného pole

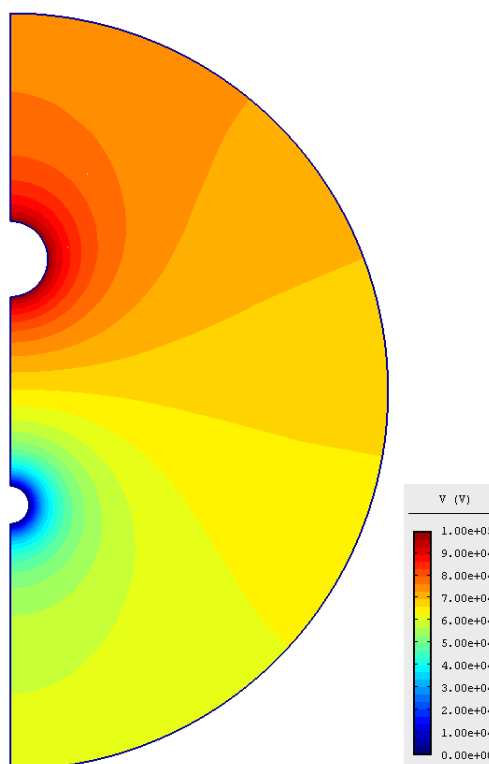
Po výpočtu si v levém okně můžeme vybrat, co si necháme programem zobrazit a v pravém okně získáme vypočítané hodnoty pro bod, hranu nebo plochu.

Například pro vykreslení vektorů a ekvipotenciál je zaškrtneme a zobrazíme bez skalárního pole pomocí *2D Bez zobrazení*, abychom viděli jen je. V záložce *Postprocessor* jim můžeme zvolit různé parametry, jako počet a velikost kontur i vektorů. Bylo nastaveno 50 kontur se šířkou 1,20 a vektory v měřítku 2,40 v počtu 100. Výsledné zobrazení je znázorněno na obr. F.4, kde vidíme jak vektory směřují od horní elektrody s vyšším potenciálem ke spodní uzemněné elektrodě.

Nyní si necháme zaškrtnutím možnosti *2D skalární zobrazení* a vybráním veličiny *Skalární el.potenciál* vykreslit rozložení elektrického potenciálu ve vyšetřované oblasti viz obr. F.5.



Obr. F.4: Zobrazení ekvipotenciál a vektorů



Obr. F.5: Skalární rozložení elektrického potenciálu

Na obr. F.5 je vidět, že velikost elektrického potenciálu se mění s jednotlivými ekvipotenciálami, tím jsme si ověřili správnost řešení, které respektuje fyzikální zákony.

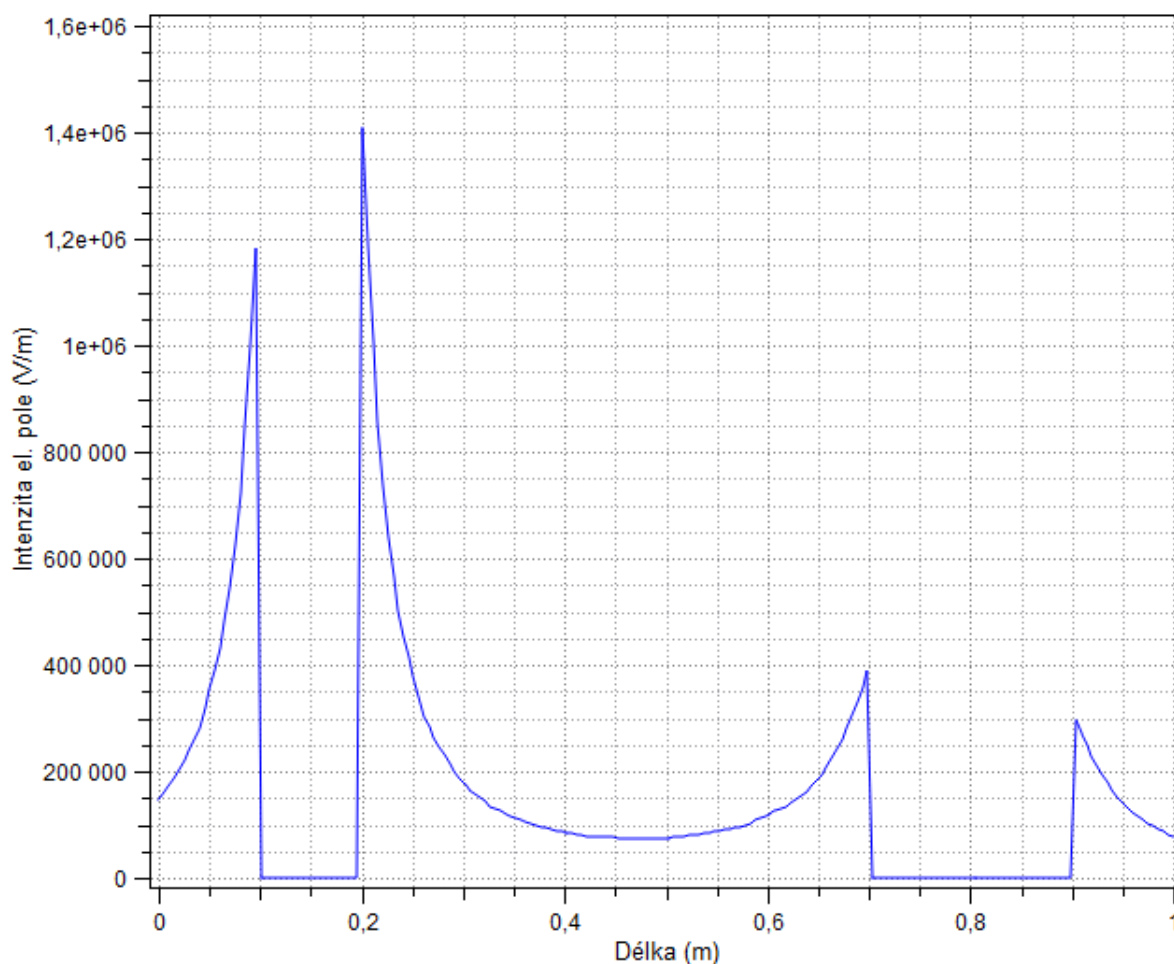
F.4.2 Zobrazení výsledku pomocí grafu

Po kliknutí na ikonku, která vypadá jako graf funkce sinus v horní liště, se otevře nabídka *Graf*, kde si můžeme nechat vykreslit graf pro vybranou veličinu podél zadané přímky.

Vybereme intenzitu elektrického pole a zadáme souřadnice podle tab. F.4, výsledný graf podél osy jiskřiště je zobrazen na obr. F.6.

	$R[m]$	$Z[m]$
Počátek	0	-0,45
Konec	0	0,55

Tab. F.4: Hodnoty souřadnic



Obr. F.6: Graf intenzity elektrického pole podél osy jiskřiště

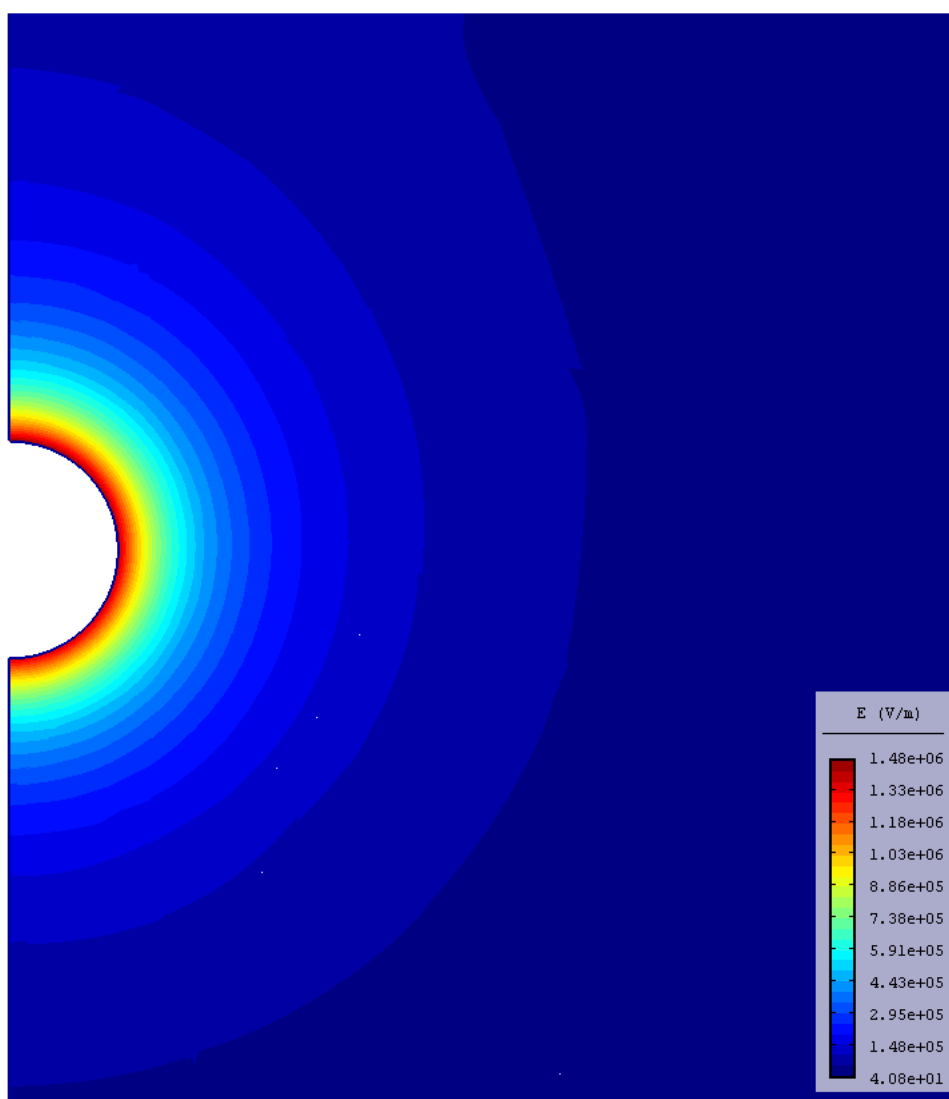
Z grafu je vidět, že na horním okraji spodní elektrody je největší hodnota intenzity elektrického pole. Tento bod nás tedy bude zajímat při výpočtu, zda jiskřiště nepřekročí zadanou elektrickou pevnost okolního vzduchu. Pro zjištění přesné hodnoty intenzity na horním okraji spodní elektrody klikneme na *Lokální veličina v bodě* pod pravým oknem výsledků *Lokální veličiny* a zadáme hodnoty $R = 0$ a $Z = -0,25$. Výsledná intenzita

elektrického pole je $E = 1,455 \cdot 10^6 \frac{\text{V}}{\text{m}}$ a zadaná hodnota elektrické pevnosti vzduchu byla $E_p = 30 \frac{\text{kV}}{\text{cm}}$, která po přepočtu na stejné jednotky vychází $E_p = 3 \cdot 10^6 \frac{\text{V}}{\text{m}}$.

$$3 > 1,455$$

$$E_p > E$$

A proto nebude vzduch pro zadané napětí mezi elektrodami jiskříště proražen. Dokonce vychází, že rezerva je přibližně dvojnásobná. Ale nesmíme zapomenout, že elektrická pevnost vzduchu se mění s jeho vlastnostmi, hlavně vlhkostí a může klesnout až trojnásobně. Pro takovou hodnotu by již jiskříště vzduch prorazilo. Rozložení intenzity elektrického pole kolem spodní elektrody vidíme na obr. F.7



Obr. F.7: Skalární zobrazení intenzity elektrického pole kolem spodní elektrody

Příloha G

Model elektromagnetu pomocí Python skriptu

G.1 Python skript

Zadání modelu bude stejné jako pro předchozí model elektromagnetu řešený bez využití Python skriptu. Ale nyní nebudeme model ručně kreslit, zadávat jednotlivé body, spojovat hrany, přidávat okrajové podmínky apod. ale využijeme možnosti, že se to samé dá provést pomocí programovacího jazyka Python.

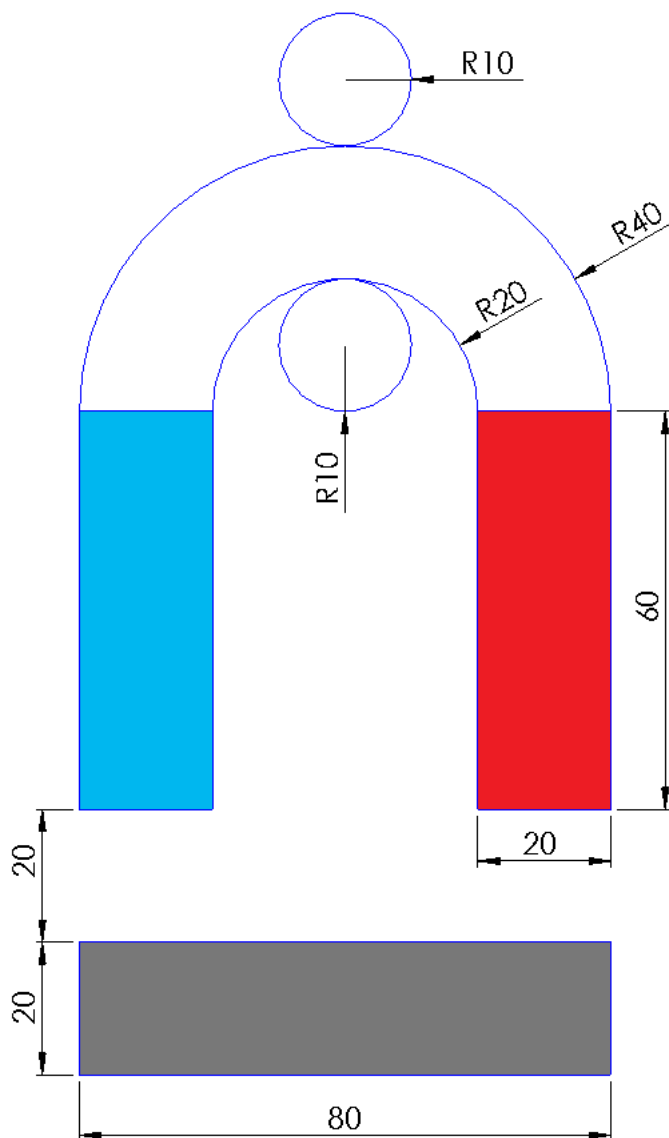
Pomocí příkazů, které můžeme nalézt v nápovědě programu pod *Scripting / Command*, a s využitím klíčových slov, které jsou uvedeny též v nápovědě pod *Scripting / Keyword.list*, vytvoříme v programovacím jazyce Python skript, který nakreslí celý model a vygeneruje nám výsledky.

G.2 Formulace problému

Model je tvořen stacionárním magnetickým polem vyvolaným stejnosměrným elektrickým proudem s permanentními magnety - viz obr. G.1. Je dána cívka tvořená tenkým měděným drátem s relativní permeabilitou μ_{r0} , elektrickou vodivostí γ protékaná proudem s proudovou hustotou J . Kotva i železo jsou tvořeny lineární ocelí s relativní permeabilitou μ_{r1} . Permanentní magnety jsou tvořeny materiálem s remanentní magnetickou indukcí B_r a relativní permeabilitou μ_{r2} . Celý model je obklopen vzduchem s relativní permeabilitou μ_{r0} . Rozměry všech částí elektromagnetu jsou uvedeny na obr. G.1, kde jsou pomocí barev rozděleny jednotlivé oblasti, kotva je znázorněna šedivou barvou, cívky jsou kružnice a permanentní magnety jsou znázorněny podle jejich polarity modrou a červenou barvou. Poslední neobarvená část tvoří železo, které rozvádí magnetickou indukci do permanentních magnetů.

$$\mu_{r0} = 1 \quad \mu_{r1} = 1000 \quad \mu_{r2} = 1,05$$

$$\gamma = 6 \cdot 10^7 \frac{\text{S}}{\text{m}} \quad J = 2 \cdot 10^4 \frac{\text{A}}{\text{m}^2} \quad B_r = 1,07 \text{ T}$$



Obr. G.1: Rozměry elektromagnetu v cm

G.3 Popis využitých příkazů

Zde bude popsán význam jednotlivých použitých příkazů. Parametry zapsané v uvozovkách jsou slovní parametry a musejí být zapsány do uvozovek, parametry bez uvozovek budou tvořeny číslem, které není v uvozovkách. Seznam všech klíčových slov, která se píšou místo jednotlivých parametrů, je umístěn v nápovědě pod *Scripting / Keyword.list*. Můžeme si vytvořit konstanty pro často používané hodnoty tak, že jednoduše napíšeme na řádek název proměnné a za rovnítko její hodnotu. V programovacím jazyku Python

se za příkazy nedávají středníky, jak je tomu zvykem u většiny ostatním programovacích jazyků.

Například pro vytvoření konstanty pro proudovou vodivost $J = 2 \cdot 10^4$ napíšeme:

J = 2e4

Prvním příkazem nastavíme parametry problému.

newdocument("název", "typ problému", "fyzikální pole", počet zjemnění, řád polynomu, "typ adaptivity", počet adaptivních kroků, tolerance adaptivity, frekvence, "druh analýzy", časový krok, celkový čas, počáteční podmínka)

My řešíme model podle tab. G.1 a proto zapíšeme příkaz v tomto tvaru:

newdocument("Elektromagnet", "planar", "magnetic", 0, 1, "hp-adaptivity", 30, 1, 0, "steadystate", 1.0, 1.0, 0.0)

Typ problému	Kartézský
Fyzikální pole	Magnetické pole
Druh analýzy	Ustálený stav
Typ sítě	Trojúhelníková
Počet zjemnění	0
Řád polynomu	1
Typ adaptivity	hp-adaptivita
Počet adaptivních kroků	20
Tolerance adaptivity [%]	1

Tab. G.1: Parametry nastavení problému

Dalším příkazem vytvoříme okrajovou podmínku.

addboundary("název", "typ", hodnota)

V našem případě jde o fiktivní hranici nekonečna a jako její parametr zadáme magnetický vektorový potenciál o hodnotě nula.

addboundary("Fiktivni hranice nekonecna", "magnetic`vector`potential", 0, 0)

Poté příkazem vytvoříme jednotlivé materiály.

addmaterial("název", proudová hustota reálná složka, proudová hustota imaginární složka, permeabilita, vodivost, remanentní magnetická indukce, směr remanence, rychlost ve směru x, rychlost ve směru y, úhlová rychlost)

Například materiál cívky vytvoříme příkazem s těmito parametry:

addmaterial("Civka B", J, 0, ur0, Gama, 0, 0, 0, 0, 0)

Následně vytvoříme jednotlivé hrany pomocí příkazu:

addedge(X počátku, Y počátku, X konce, Y konce, úhel, "název okrajové podmínky")

Pro jednu z hran je uveden příklad. Pokud hrana nemá okrajovou podmínku, napíšeme místo názvu "none".

addedge(0.4, 2.35, -1.6, 0.35, 90, "Fiktivni hranice nekonecna")

Umístění značek oblasti provede následující příkaz.

addlabel(souřadnice X, souřadnice Y, 0, 0, "název materiálu")

Nuly jsou pro parametr plochy trojúhelníku sítě a napíšeme je u každé značky oblasti.

addlabel(0.4, 0.3, 0, 0, "Vzduch")

Spustí řešení problému - **solve()**

Nastaví celý model do viditelné oblasti - **zoombestfit()**

Nastaví zobrazení vybrané oblasti na základě jejich souřadnic.

zoomregion(X počátku, Y počátku, X konce, Y konce)

Další příkazy jsou již pro manipulaci s výsledky nebo se zobrazením definiční oblasti.

Jejich jediný parametr je booleovský, tedy nula pro nezobrazení a jedna pro zobrazení.

Zobrazení mřížky - **showgrid(1)**

Zobrazení geometrie - **showgeometry(1)**

Zobrazení vektorů - **showvectors(1)**

Zobrazení počáteční sítě - **showinitialmesh(1)**

Zobrazení výsledné sítě - **showsolutionmesh(1)**

Zobrazení siločár - **showcontours(1)**

Zobrazí vybrané pole zvolené veličiny v zadaném rozsahu.

showscalar("typ", "veličina", "složka veličiny", rozsah od, rozsah do)

kde typ je { "none", "scalar", "scalar3d", "order" }

Příklad pro zobrazení skalárního pole vektorového magnetického potenciálu.

showscalar("scalar", "magnetic vector potential real", "scalar", -0.2, 0.2)

Uloží výsledek, který je právě zobrazován v definiční oblasti a s příslušným aktuálním nastavením vybrané oblasti do souboru ve stejné složce, ve které se nachází soubor scriptu.

saveimage("název-obrázku.png")

Dalším příkazem můžeme uživateli zobrazit zprávu, kterou musí potvrdit. Tím mu můžeme dát na vědomí například, že bude obrácena polarita pravého magnetu.

message("Nyni bude obracena polarita praveho magnetu")

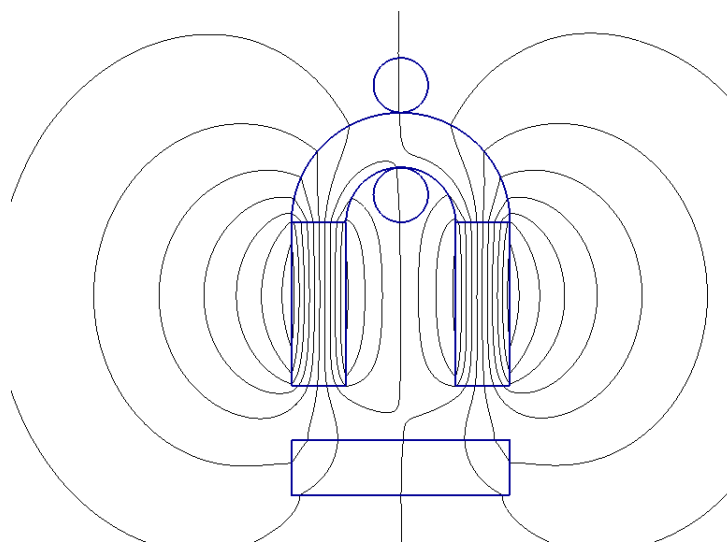
Poslední příkaz, který byl ve scriptu použit, je pro úpravu materiálů. Tento příkaz má stejné parametry, jako příkaz pro vytvoření materiálů. Byl použit pro otočení polarit magnetů a směru toku proudu cívkou.

modifymaterial("Magnet pravy", 0, 0, ur2, 0, Br, 90, 0, 0, 0)

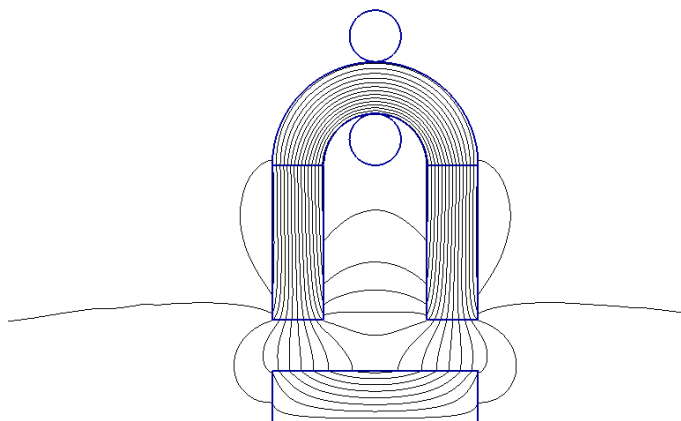
G.3.1 Vygenerované výsledky

Skript nám vygeneruje rozložení siločár a vektorů, zobrazí řešenou a počáteční síť, rozložení skalárního pole vektorového magnetického potenciálu a rozložení skalárního pole magnetické indukce. Všechny výsledky uloží do obrázků do složky, kde je umístěn soubor

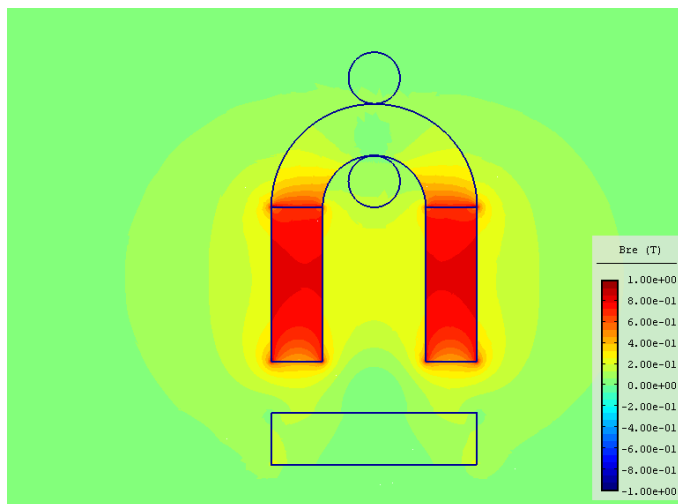
skriptu. Následují vybrané výsledky, které nám skript vytvořil. Na obr. G.2 jsou zobrazeny siločáry pro souhlasné uložení magnetů, výsledek můžeme porovnat s nesouhlasným uložení magnetů na obr. G.3. Na dalších obrázcích je znázorněno rozložení magnetické indukce opět pro souhlasně uložené magnety obr. G.4 a pro nesouhlasně uložené magnety obr. G.5. Protože jsme u obou obrázků zvolili stejné hodnoty rozsahu hodnot, můžeme snadněji porovnat jejich rozdíly. Nakonec na posledním vybraném obrázku je zobrazeno vektorové magnetické pole pro magnety uložené souhlasně viz obr. G.6.



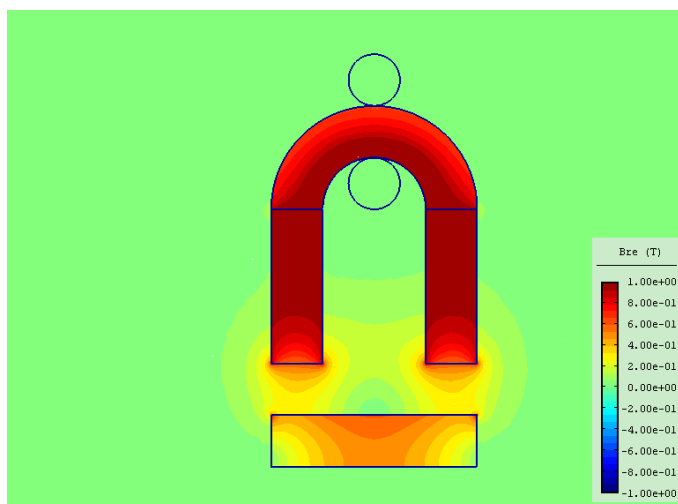
Obr. G.2: Zobrazení siločár pro magnety uložené souhlasně



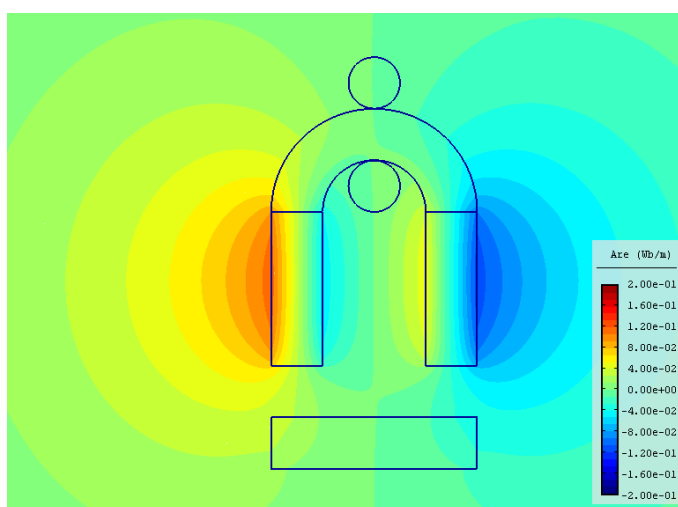
Obr. G.3: Zobrazení siločár pro magnety uložené nesouhlasně



Obr. G.4: Rozložení skalárního pole magnetické indukce pro magnety uložené souhlasně



Obr. G.5: Rozložení skalárního pole magnetické indukce pro magnety uložené nesouhlasně



Obr. G.6: Rozložení vektorového magnetického potenciálu pro magnety uložené souhlasně

G.4 Kompletní script pro model elektromagnetu

```

# Model elektromagnetu
  newdocument("Elektromagnet", "planar", "magnetic", 0, 1, "hp-adaptivity", 30, 1, 0,
"steadystate", 1.0, 1.0, 0.0)
  # Zadane hodnoty
  ur0 = 1
  ur1 = 1000
  ur2 = 1.05
  Gama = 6e7
  J = 2e4
  Br = 1.07
  # Okrajove podminky
  addboundary("Fiktivni hranice nekonecna", "magnetic vector potential", 0, 0)
  # Materialy
  addmaterial("Vzduch", 0, 0, ur0, 0, 0, 0, 0, 0, 0)
  addmaterial("Civka A", -J, 0, ur0, Gama, 0, 0, 0, 0, 0)
  addmaterial("Civka B", J, 0, ur0, Gama, 0, 0, 0, 0, 0)
  addmaterial("Kotva-Zelezo", 0, 0, ur1, 0, 0, 0, 0, 0, 0)
  addmaterial("Magnet levy", 0, 0, ur2, 0, Br, 90, 0, 0, 0)
  addmaterial("Magnet pravy", 0, 0, ur2, 0, Br, 270, 0, 0, 0)
  # Hrany
  addedge(0.2, 0, 0.2, 0.6, 0, "none")
  addedge(0, 0.6, 0, 0, 0, "none")
  addedge(0, 0, 0.2, 0, 0, "none")
  addedge(0.6, 0, 0.8, 0, 0, "none")
  addedge(0.8, 0, 0.8, 0.6, 0, "none")
  addedge(0.6, 0.6, 0.6, 0, 0, "none")
  addedge(0.6, 0.6, 0.4, 0.8, 90, "none")
  addedge(0.4, 0.8, 0.2, 0.6, 90, "none")
  addedge(0.4, 1, 0, 0.6, 90, "none")
  addedge(0.8, 0.6, 0.4, 1, 90, "none")
  addedge(0, -0.2, 0, -0.4, 0, "none")
  addedge(0, -0.4, 0.8, -0.4, 0, "none")
  addedge(0.8, -0.4, 0.8, -0.2, 0, "none")
  addedge(0.8, -0.2, 0, -0.2, 0, "none")
  addedge(0.4, 1.201, 0.3, 1.101, 90, "none")
  addedge(0.5, 1.101, 0.4, 1.201, 90, "none")
  addedge(0.4, 0.599, 0.5, 0.699, 90, "none")
  addedge(0.3, 0.699, 0.4, 0.599, 90, "none")
  addedge(0.5, 0.699, 0.4, 0.799, 90, "none")

```

```
addedge(0.4, 0.799, 0.3, 0.699, 90, "none")
addedge(0.3, 1.101, 0.4, 1.001, 90, "none")
addedge(0.4, 1.001, 0.5, 1.101, 90, "none")
addedge(0.4, 2.35, -1.6, 0.35, 90, "Fiktivni hranice nekonecna")
addedge(-1.6, 0.35, 0.4, -1.75, 90, "Fiktivni hranice nekonecna")
addedge(0.4, -1.75, 2.4, 0.35, 90, "Fiktivni hranice nekonecna")
addedge(2.4, 0.35, 0.4, 2.35, 90, "Fiktivni hranice nekonecna")
addedge(0, 0.6, 0.2, 0.6, 0, "none")
addedge(0.6, 0.6, 0.8, 0.6, 0, "none")
# Znacky oblasti
addlabel(0.4, 1.1, 0, 0, "Civka A")
addlabel(0.4, 0.7, 0, 0, "Civka B")
addlabel(0.4, 0.9, 0, 0, "Kotva-Zelezo")
addlabel(0.4, 0.3, 0, 0, "Vzduch")
addlabel(0.1, 0.3, 0, 0, "Magnet levy")
addlabel(0.7, 0.3, 0, 0, "Magnet pravy")
addlabel(0.4, -0.3, 0, 0, "Kotva-Zelezo")
# Nastaveni defaultniho zobrazeni
zoombestfit()
solve()
showgrid(0)
showgeometry(1)
showvectors(0)
showinitialmesh(0)
showsolutionmesh(0)
showcontours(0)
showscalar("none", "magnetic`vector`potential`real", "scalar",-0.2,0.2)
# Vysledky pro magnety 90-270 -J +J
# Sit
zoomregion(-1.4, -0.8, 2.15, 1.65 )
showinitialmesh(1)
showsolutionmesh(1)
saveimage("90-270-Sit-pocatecni-resena.png")
showinitialmesh(0)
showsolutionmesh(0)
# Vektory
zoomregion(-0.3, -0.5, 0.9, 1.3 )
showvectors(1)
saveimage("90-270-vektory.png")
showvectors(0)
```

```
# Silocary
zoomregion(-0.3, -0.5, 0.9, 1.3 )
showcontours(1)
saveimage("90-270-kontury.png")
showcontours(0)
# Vektorovy magneticky potencial
zoomregion(-0.3, -0.5, 0.9, 1.3 )
showscalar("scalar", "magnetic vector potential real", "scalar", -0.2, 0.2)
saveimage("90-270-mag-potencial.png")
showscalar("none", "magnetic vector potential real", "scalar", -0.2, 0.2)
# Magneticka indukce
zoomregion(-0.3, -0.5, 0.9, 1.3 )
showscalar("scalar", "magnetic flux density real", "magnitude", -1, 1)
saveimage("90-270-mag-indukce.png")
showscalar("none", "magnetic flux density real", "magnitude", -1, 1)
# 90-90 -J +J
# Uprava praveho magnetu
message("Nyni bude obracena polarita praveho magnetu")
modifymaterial("Magnet pravy", 0, 0, ur2, 0, Br, 90, 0, 0, 0)
zoombestfit()
solve()
# Vysledky pro magnety 90-90 -J +J
# Sit
zoomregion(-1.4, -0.8, 2.15, 1.65 )
showinitialmesh(1)
showsolutionmesh(1)
saveimage("90-90-Sit-pocatecni-resena.png")
showinitialmesh(0)
showsolutionmesh(0)
# Vektory
zoomregion(-0.3, -0.5, 0.9, 1.3 )
showvectors(1)
saveimage("90-90-vektory.png")
showvectors(0)
# Silocary
zoomregion(-0.3, -0.5, 0.9, 1.3 )
showcontours(1)
saveimage("90-90-kontury.png")
showcontours(0)
# Vektorovy magneticky potencial
```

```

zoomregion(-0.3, -0.5, 0.9, 1.3 )
showscalar("scalar", "magnetic vector potential real", "scalar", -0.2, 0.2)
saveimage("90-90-mag-potencial.png")
showscalar("none", "magnetic vector potential real", "scalar", -0.2, 0.2)
# Magneticka indukce
zoomregion(-0.3, -0.5, 0.9, 1.3 )
showscalar("scalar", "magnetic flux density real", "magnitude", -1, 1)
saveimage("90-90-mag-indukce.png")
showscalar("none", "magnetic flux density real", "magnitude", -1, 1)
# 270-90 -J +J
# Uprava obou magnetu
message("Nyni bude obracena polarita obou magnetu")
modifymaterial("Magnet levy", 0, 0, ur2, 0, Br, 270, 0, 0, 0)
modifymaterial("Magnet pravy", 0, 0, ur2, 0, Br, 90, 0, 0, 0)
zoombestfit()
solve()
# Vysledky pro magnety 270-90 -J +J
# Sit
zoomregion(-1.4, -0.8, 2.15, 1.65 )
showinitialmesh(1)
showsolutionmesh(1)
saveimage("270-90-Sit-pocatecni-resena.png")
showinitialmesh(0)
showsolutionmesh(0)
# Vektory
zoomregion(-0.3, -0.5, 0.9, 1.3 )
showvectors(1)
saveimage("270-90-vektory.png")
showvectors(0)
# Silocary
zoomregion(-0.3, -0.5, 0.9, 1.3 )
showcontours(1)
saveimage("270-90-kontury.png")
showcontours(0)
# Vektorovy magneticky potencial
zoomregion(-0.3, -0.5, 0.9, 1.3 )
showscalar("scalar", "magnetic vector potential real", "scalar", -0.2, 0.2)
saveimage("270-90-mag-potencial.png")
showscalar("none", "magnetic vector potential real", "scalar", -0.2, 0.2)
# Magneticka indukce

```

```
zoomregion(-0.3, -0.5, 0.9, 1.3 )
showscalar("scalar", "magnetic flux density real", "magnitude", -1, 1)
saveimage("270-90-mag-indukce.png")
showscalar("none", "magnetic flux density real", "magnitude", -1, 1)
# 270-90 +J -J
# Uprava obou magnetu
message("Nyni bude zamenen smer toku proudu civkou")
modifymaterial("Civka A", J, 0, ur0, Gama, 0, 0, 0, 0)
modifymaterial("Civka B", -J, 0, ur0, Gama, 0, 0, 0, 0)
zoombestfit()
solve()
# Vysledky pro magnety 270-90 +J -J
# Vektory
zoomregion(-0.3, -0.5, 0.9, 1.3 )
showvectors(1)
saveimage("270-90+J-J-vektory.png")
showvectors(0)
# Silocary
zoomregion(-0.3, -0.5, 0.9, 1.3 )
showcontours(1)
saveimage("270-90+J-J-kontury.png")
showcontours(0)
# Vektorovy magneticky potencial
zoomregion(-0.3, -0.5, 0.9, 1.3 )
showscalar("scalar", "magnetic vector potential real", "scalar", -0.2, 0.2)
saveimage("270-90+J-J-mag-potencial.png")
showscalar("none", "magnetic vector potential real", "scalar", -0.2, 0.2)
# Magneticka indukce
zoomregion(-0.3, -0.5, 0.9, 1.3 )
showscalar("scalar", "magnetic flux density real", "magnitude", -1, 1)
saveimage("270-90+J-J-mag-indukce.png")
showscalar("none", "magnetic flux density real", "magnitude", -1, 1)
```


Příloha H

Model feromagnetického aktuátoru pomocí Python skriptu

H.1 Python skript

Zadání modelu bude stejné jako pro předchozí model feromagnetického aktuátoru řešený bez využití Python skriptu. V tomto příkladě ale využijeme skriptu napsaného v programovacím jazyce Python. Tento skript se postará o vytvoření celého modelu a vygeneruje nám výsledky, při jeho tvorbě budou využity vybrané příkazy z nápovědy programu. Seznam všech příkazů je uveden v nápovědě programu pod *Scripting / Command* a seznam všech klíčových slov je též v nápovědě pod *Scripting / Keyword_list*.

H.2 Formulace problému

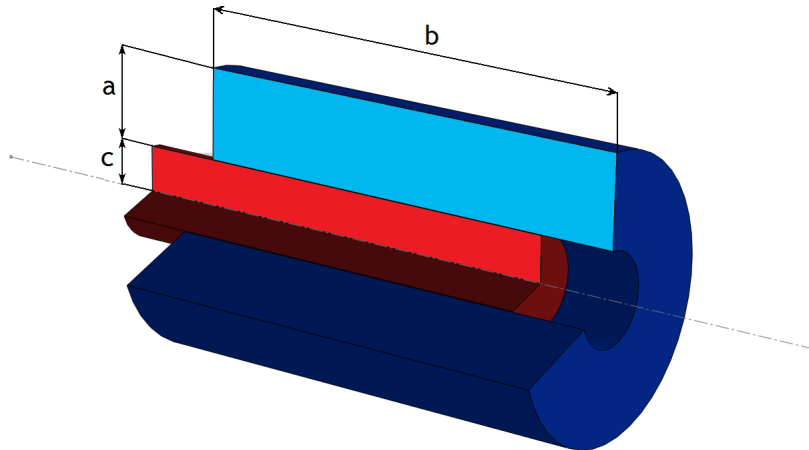
Je dán feromagnetický aktuátor, který je tvořen cívkou s ocelovým jádrem, obklopený vzduchem viz obr. H.1. Kotva je považována za lineární ocel, která má relativní permeabilitu μ_{r1} a její rozměry jsou poloměr c a délka, která je stejná jako u cívky. Cívka i okolní vzduch mají relativní permeabilitu μ_{r0} . Cívka má rozměry: šířka a , délka b . Cívkou teče proudová hustota J_{ext} . Model bude řešen jako 2D osově symetrická úloha, plocha pro řešení je řez na obr. H.1, který je vyznačen světlejšími barvami.

$$\mu_{r0} = 1 \quad \mu_{r1} = 1500 \quad J_{ext} = 1000 \frac{\text{A}}{\text{m}^2} \quad a = 0,1 \text{ m} \quad b = 0,5 \text{ m} \quad c = 0,05 \text{ m}$$

H.3 Popis využitých příkazů

Základní příkazy použité pro nakreslení modelu, vytvoření okrajových podmínek a materiálů a práci s výsledky byly popsány v předchozí kapitole v části G.3 a zde nebudou znovu vypisovány. Zde budou vysvětleny významy příkazů, které jsou použity nově.

Následující příkaz vybere všechny hrany, které jsou uvedeny číslem v jeho parametru.



Obr. H.1: 3D model cívky (modrá) s částečně vsunutým jádrem (červená)

selectedge(číslo hrany, ...)

další příkaz posune vybrané hrany v zadaném směru, může je také duplikovat.

moveselection(posun X, posun Y, duplikování)

Duplikování může být ano nebo ne {True, False}.

Výsledky pro vyjmenované hrany v parametrech příkazu.

result = surfaceintegral(číslo hrany, ...)

Do proměnné r uložíme výsledek výpočtu v závorce.

r.append(0.15 - (j*posun))

Do proměnné F uložíme výslednou maxwellovskou sílu ve směru osy z "Fy".

F.append(result["Fy"])

Výsledky postupně tiskneme do konzole následujícím příkazem

print(r[-1], F[-1], j)

Výsledky pro vyjmenované plochy v parametrech příkazu.

result = volumeintegral(0,1,2)

Do proměnné W uložíme výslednou energii magnetického pole "Wm".

W.append(result["Wm"])

Příkaz **for** vytvoří cyklus pro postupné posouvání kotvy v cívce, jako v ostatních známých programovacích jazycích.

Pro vykreslení vypočítaných hodnot do grafu využijeme následující kus kódu.

import pylab as pl - importuje potřebné knihovny

pl.plot(rw, Fw) - zadá proměnné pro jednotlivé osy

pl.grid(1) - zobrazí mřížku

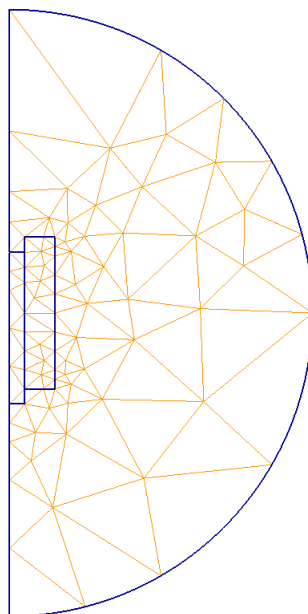
pl.xlabel("Posun (m)") - popíše osu X

pl.ylabel("F (N)") - popíše osu Y

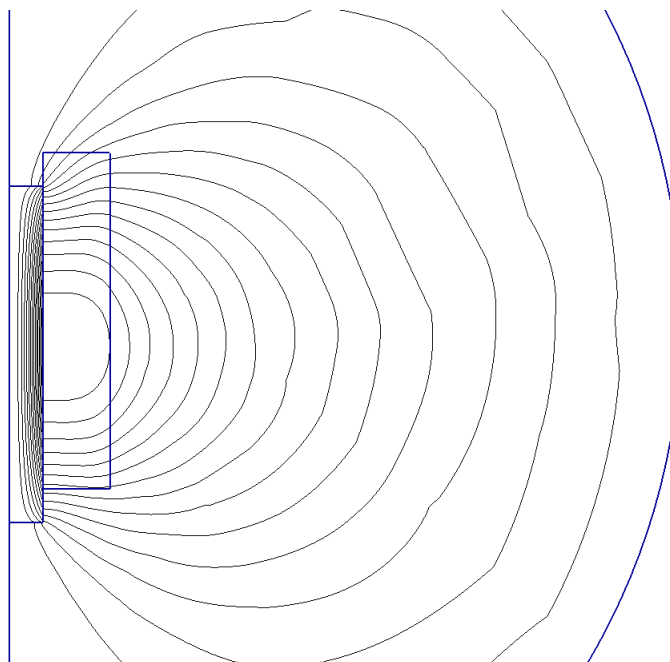
pl.show() - zobrazí graf

H.3.1 Vygenerované výsledky

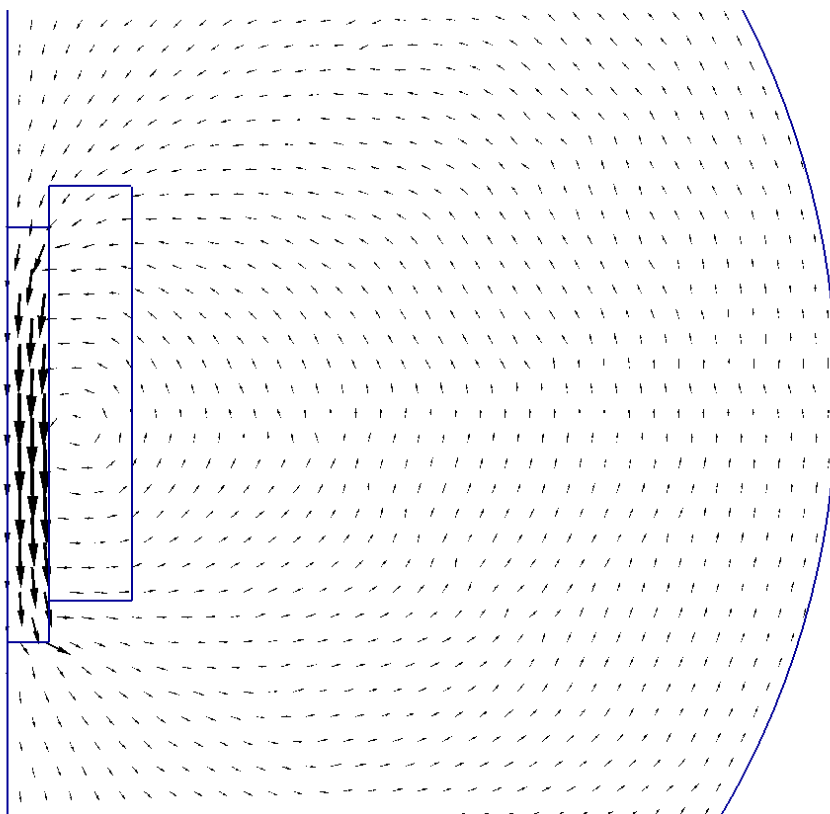
Script nám vytvoří několik obrázků a uloží je do složky, ve které se sám nachází. Na obr. H.2 je znázorněna počáteční síť, která je vytvořena zdiskretizováním definiční oblasti. Zobrazení siločár vidíme na obr. H.3 a vektorů na obr. H.4. Skalární zobrazení indukce pro částečně vsunutou kotvu je zobrazeno na obr. H.5.



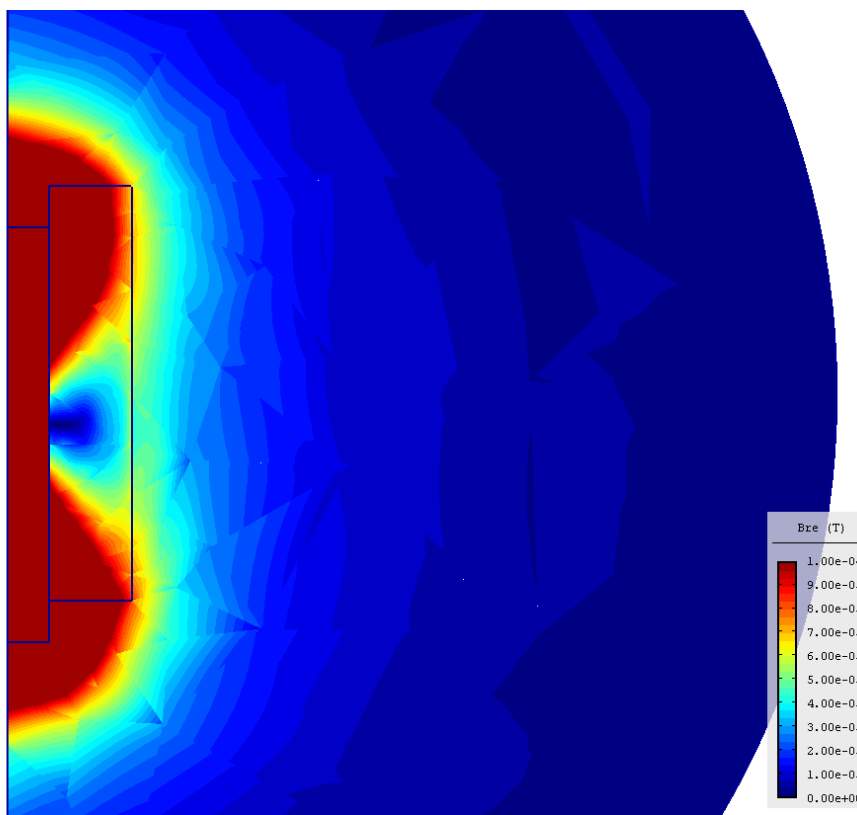
Obr. H.2: Vykreslení počáteční sítě



Obr. H.3: Zobrazení siločár magnetického pole



Obr. H.4: Zobrazení vektorů magnetického pole

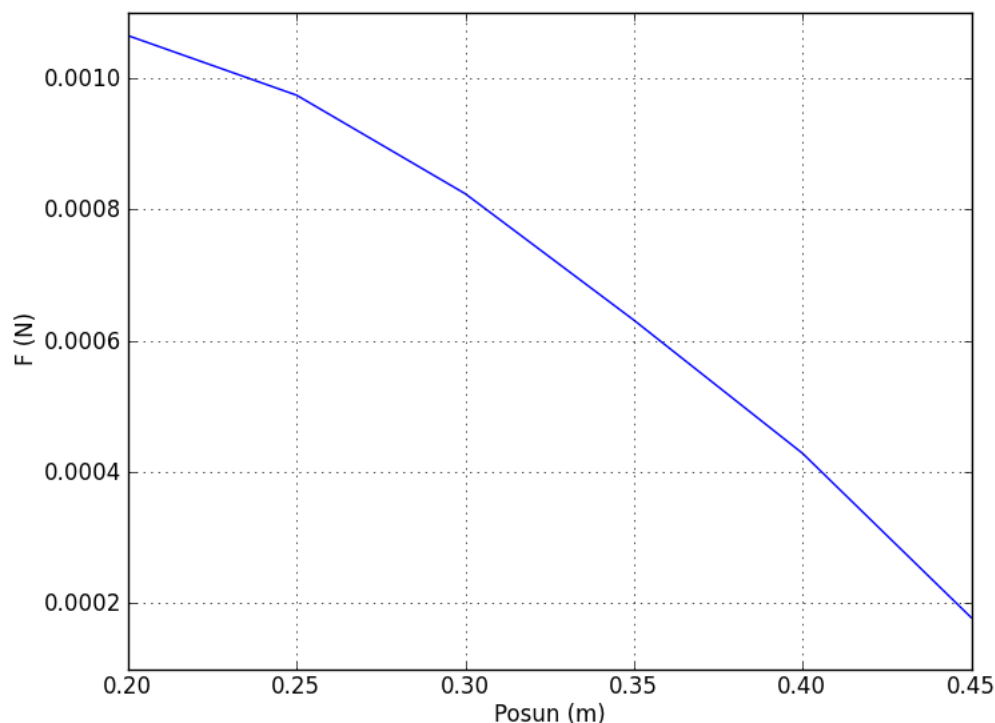


Obr. H.5: Rozložení skalárního pole magnetické indukce

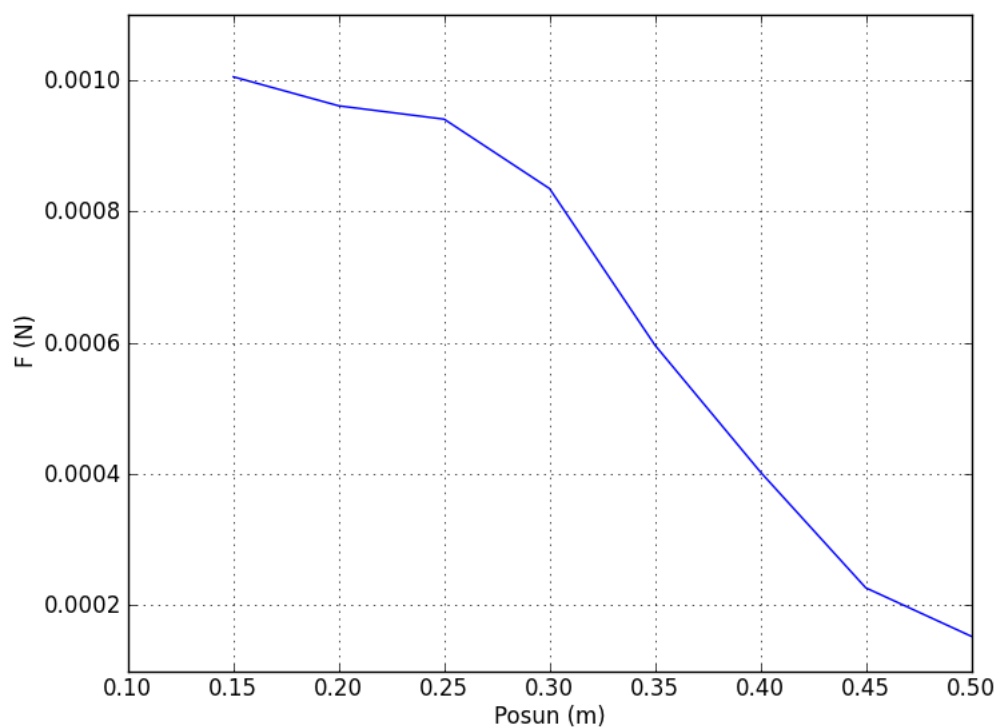
H.3.2 Výpočet síly působící na kotvu

Script nám hlavně dokázal ušetřit spoustu práce při výpočtu síly působící na částečně vsunutou kotvu. Nemuseli jsme ručně posouvat všechny body kotvy a pro každé vyřešené pole odečítat hodnoty, ze kterých bychom pak vytvořili graf jako v části 4.4.3. Vše potřebné obstaral script. My jsme si jen po výpočtu uložili zobrazené grafy v potřebném formátu. První je zobrazen na obr. H.6 a byl vypočten pomocí energie magnetického pole podle vzorečku H.1. Tímto vzorcem vypočteme sílu působící mezi body 1 a 2, protože však chceme mít hodnoty na porovnání ve stejných posuvech, jako při druhém způsobu výpočtu využijeme toho, že použitím bodů 1 a 3 získáme hodnoty síly působící v bodě 2. Druhý byl vypočítán pomocí Maxwellovské síly a je zobrazen na obr. H.7. Když oba výsledky porovnáme zjistíme, že se výrazně neliší. Ale přesto jsou jejich hodnoty mírně odlišné, v měřítku grafu to dokonce vypadá, že se liší výrazně. Hodnoty vypočtené pomocí magnetické síly můžeme považovat za přesnější, protože výpočet Maxwellovské síly na jednotlivé hrany kotvy v programu není tak přesný.

$$F = \frac{W_{m2} - W_{m1}}{\text{posun}_2 - \text{posun}_1} \quad [N] \quad (\text{H.1})$$



Obr. H.6: Velikost síly působící na kotvu získané z energie magnetického pole



Obr. H.7: Velikost síly působící na kotvu získané z Maxwellovské síly

H.4 Kompletní script pro model feromagnetického aktuátoru

```
# Model feromagnetického aktuátoru
newdocument("Python-aktuator", "axisymmetric", "magnetic", 0, 1, "hp-adaptivity",
20, 1, 0, "steadystate", 1.0, 1.0, 0.0)
# Zadane hodnoty
ur1 = 1500
Jext = 1000
posun = -0.05
# Okrajove podminky
addboundary("Antisymetrie", "magnetic vector potential", 0, 0)
# Materialy
addmaterial("Vzduch", 0, 0, 1, 0, 0, 0, 0, 0, 0)
addmaterial("Civka", Jext, 0, 1, 0, 0, 0, 0, 0, 0)
addmaterial("Kotva", 0, 0, ur1, 0, 0, 0, 0, 0, 0)
# Hrany
addedge(0.05, -0.25, 0.15, -0.25, 0, "none")
addedge(0.15, -0.25, 0.15, 0.25, 0, "none")
addedge(1, 0, 0, 1, 90, "Antisymetrie")
```

```
addedge(0, -1, 1, 0, 90, "Antisymetrie")
addedge(0.05, 0.25, 0.15, 0.25, 0, "none")
addedge(0.05, 0.25, 0.05, 0.2, 0, "none")
addedge(0.05, 0.2, 0.05, -0.25, 0, "none")
addedge(0.05, -0.25, 0.05, -0.3, 0, "none")
addedge(0.05, -0.3, 0, -0.3, 0, "none")
addedge(0, -0.3, 0, 0.2, 0, "Antisymetrie")
addedge(0, 0.2, 0.05, 0.2, 0, "none")
addedge(0, 1, 0, 0.2, 0, "Antisymetrie")
addedge(0, -0.3, 0, -1, 0, "Antisymetrie")
# Znacky oblasti
addlabel(0.1, 0.05, 0, 0, "Civka")
addlabel(0.2, 0.05, 0, 0, "Vzduch")
addlabel(0.025, 0.05, 0, 0, "Kotva")
# Nastaveni defaultniho zobrazeni
zoombestfit()
solve()
showgrid(0)
showgeometry(1)
showvectors(0)
showinitialmesh(0)
showsolutionmesh(0)
showcontours(0)
showscalar("none", "magnetic vector potential real", "scalar", -0.2, 0.2)
# Vysledky pole
# Sit
zoombestfit()
showinitialmesh(1)
showsolutionmesh(1)
saveimage("Sit.png")
showinitialmesh(0)
showsolutionmesh(0)
# Vektory
zoomregion(-0.2, -0.4, 1, 0.35 )
showvectors(1)
saveimage("Vektory.png")
showvectors(0)
# Silocary
zoomregion(-0.2, -0.4, 1, 0.35 )
showcontours(1)
```

```
saveimage("Kontury.png")
showcontours(0)
# Magneticka indukce
zoomregion(-0.2, -0.4, 1, 0.35 )
showscalar("scalar", "magnetic flux density real", "magnitude", 0, 0.0001)
saveimage("Mag-indukce.png")
showscalar("none", "magnetic flux density real", "magnitude", -1, 1)
# Posunuti kotvy na vychozi pozici
zoombestfit()
selectlabel(2)
moveselection(0, 2*posun, False)
selectededge(6, 7, 8, 10)
moveselection(0, 2*posun, False)
selectnode(0)
moveselection(0, 2*-posun, False)
# Promenne
r = []
F = []
W = []
Fw = []
# Posouvani kotvy
print("F = f(r) (N):")
for j in range(8):
    if j < 0:
        selectededge(6, 7, 8, 10)
        moveselection(0, posun, False)
        selectnode(0)
        moveselection(0, -posun, False)
        selectlabel(2)
        moveselection(0, posun, False)
    solve()
    result = surfaceintegral(6, 7, 8, 10)
    r.append(0.15 - (j*posun))
    F.append(result["Fy"])
    print(r[-1], F[-1], j)
    result = volumeintegral(0, 1, 2)
    W.append(result["Wm"])
# Vypocet sily z energie
rw = [0.2, 0.25, 0.3, 0.35, 0.4, 0.45]
```



```
Fw = [10*(W[0]-W[2]),10*(W[1]-W[3]),10*(W[2]-W[4]),10*(W[3]-W[5]),10*(W[4]-W[6]),10*(W[5]-W[7])]
```

```
# Vykreslení výsledku do grafu z energie
```

```
try:
```

```
import pylab as pl
```

```
pl.plot(rw, Fw)
```

```
pl.grid(1)
```

```
pl.xlabel("Posun (m)")
```

```
pl.ylabel("F (N)")
```

```
pl.show()
```

```
except ImportError as err:
```

```
print("Chyba skriptu: " + err.message)
```

```
# Vykreslení výsledku do grafu z maxwelovské síly
```

```
try:
```

```
import pylab as pl
```

```
pl.plot(r, F)
```

```
pl.grid(1)
```

```
pl.xlabel("Posun (m)")
```

```
pl.ylabel("F (N)")
```

```
pl.show()
```

```
except ImportError as err:
```

```
print("Chyba skriptu: " + err.message)
```