VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky

BAKALÁŘSKÁ PRÁCE

2011 Miroslav Urban

VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra aplikované matematiky

Vizualizace diskretizovaných polí ve 3 dimenzích

Visualization of Discretized Fields over Tetrahedra

2011 Miroslav Urban

Text na této stránce nahradit zadáním.

Prohlášení studenta

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně.

Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

Dne: 6.5.2011 ……………………….

Podpis

Poděkování

Rád bych poděkoval vedoucímu diplomové práce **Ing. Lukáš Dalibor, Ph.D.** za odbornou pomoc a konzultaci při vytváření této práce.

Abstrakt

Účelem této bakalářské práce je vytvořit software s přívětivým grafickým rozhraním (GUI), který zobrazuje prostorová pole. Tyto 3D pole jsou zadány konstantními funkcemi nad danou diskretizací do čtyřstěnů. Pole vizualizujeme v řezech čtyřstěnů danou rovinou. Barvy řezů odpovídají funkci. Výsledná obarvená mapa n řezů je vizualizace prostorové diskretizace.

Klíčová slova

diskretizace, GUI (grafické uživatelské rozhraní)

Abstract

The purpose of this bachelor thesis is to develop software with a friendly graphical user interface (GUI), which displays three dimensional fields. These 3D fields are specified by constant functions over the discretization into tetrahedra. The field is displayed in cuts of tetrahedra and a given plane. Colors of cuts correspond to the function. The final colored map of n cuts is visualization of the three dimensional discretization.

Key words

Discretization, GUI (graphical user interface)

Seznam použitých symbolů a zkratek

GUI	grafické uživatelské rozhraní (Graphical User Interface)
(u_1, u_2)	skalární součin vektorů u_1, u_2
$u_1 \times u_2$	vektorový součin vektorů u_1, u_2

Seznam obrázků

Obrázek 1 – Rovina ρ
Obrázek 2 – Průsečík I roviny s úsečkou
Obrázek 3 – Čtyřstěn ABCD
Obrázek 4 – Řez čtyřstěnu rovinou (ve 3 a 4 bodech)
Obrázek 5 – Nesetříděné body
Obrázek 6 – Body setříděné podle úhlů
Obrázek 7 – Znaménka sinových a kosinových složek v jednotlivých kvadrantech
Obrázek 8 – Přechod Modrá → Zelená → Červená
Obrázek 9 – Lineární funkce G → B a B → G
Obrázek 10 – Lineární funkce R → G a G → R
Obrázek 11 – Layout editor
Obrázek 12 – Property Inspector
Obrázek 13 – Rozvrhnutí komponentů GUI
Obrázek 14 – Aktivované GUI
Obrázek 15 – Vizualizace analytické funkce 1
Obrázek 16 – Vizualizace analytické funkce 2
Obrázek 17 – Elektromagnet
Obrázek 18 – Vizualizace diskretizace elektromagnetu 1
Obrázek 19 – Vizualizace diskretizace elektromagnetu 2

Obsah

1 Úvod 2
2 Vizualizace skalárních polí 3
2.1 Průsečík roviny s úsečkou 4
2.2 Řez čtyřstěnu rovinou 5
2.3 Seřazení rohových bodů 6
2.4 Tvorba barevné mapy 9
3 GUI 12
3.1 Tvorba GUI (MATLAB) 12
3.2 Vlastní tvorba GUI 14
4 Příklady vizualizovaných polí 16
4.1 Vizualizace analytické funkce 16
4.2 Zobrazení magnetického pole elektromagnetu 17
5 Závěr 20

1 Úvod

Modelování fyzikálních polí parciálními diferenciálními rovnicemi vede na prostorové diskretizace (do sítě trojúhelníků, čtyřstěnů). Po vyřešení zůstává netriviální otázkou jak zobrazit vypočtené pole. Tato práce se bude zabývat algoritmy potřebnými pro vizualizaci diskretizovaných rovinných polí. Diskretizace prostorových polí je náhrada spojitého prostředí systémem diskrétních bodů (uzlových bodů), v kterých se soustředí fyzikální parametry popisující stav nebo vlastnosti příslušného místa spojitého prostředí.
Těchto diskretizovaných polí můžeme využít např. pro numerickou metodu konečných prvků (MKP), pro řešení fyzikálních úloh (vedení tepla, pružnost) a podobně.
Jádrem práce je způsob jak dané diskretizované prostorové pole efektivně vizualizovat. Pro jeho zobrazení je třeba provést základní geometrické operace. Hlavní potíže spočívají v odladění těchto geometrických operací v programu tak, aby fungoval pro všechny možné případy, co mohou nastat.

Na začátku této práce v kapitole 2 jsou uvedeny algoritmy za pomoci, kterých probíhá vizualizace diskretizovaných prostorových polí. V kapitole 3 se popisuje tvorba GUI v programovém prostředí MATLAB. V kapitole 4 jsou ukázky vizualizovaných prostorových polí.

2 Vizualizace skalárních polí

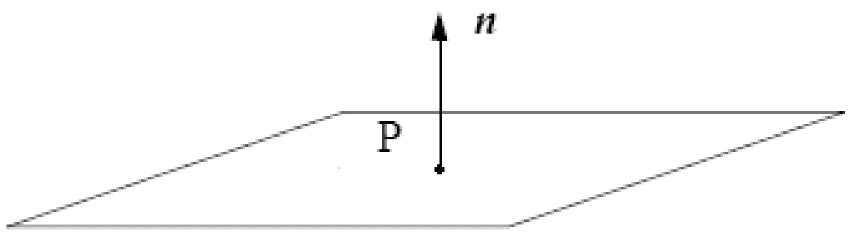
Mějme rozdělení oblasti $\Omega \in R^3$ do disjunktních čtyřstěnů.

$$\overline{\Omega} = \bigcup_{i=1}^{n} \overline{K}_i \,, \qquad K_i \bigcap K_j = \emptyset \;\; \text{pro } i \neq j$$

Chceme vizualizovat skalární pole $u(\bar{x})$, které je po částech konstantní nad touto sítí čtyřstěnů, tj. je reprezentováno vektorem $\bar{u} \in R^n$.

$$u(\bar{x}) = u_i \quad pro\ \bar{x} \in K_i$$

Vizualizovat dané skalární pole $u(\bar{x})$ budeme v řezu danou rovinou ρ, která je definována normálovým vektorem $n = [n_1, n_2, n_3]$ a bodem $P = [p_1, p_2, p_3]$.



Obrázek 1 – Rovina ρ

Obecná rovnice roviny má tvar:
$ax + by + cz + d = 0,$
kde koeficienty a, b, c nejsou současně nulové a jsou to koeficienty normálového vektoru roviny $n = [n_1, n_2, n_3]$. Proměnné x, y, z jsou souřadnice bodu ležícího v rovině. Pro daný bod $P = [p_1, p_2, p_3]$ dopočteme zbývající koeficient d

$$d = -n_1 p_1 - n_2 p_2 - n_3 p_3 \quad .$$

Obecná rovnice roviny ρ je pak v následujícím tvaru:

$$\rho\colon\ n_1 x + n_2 y + n_3 z = n_1 p_1 + n_2 p_2 + n_3 p_3$$

Vizualizace skalárního pole dosáhneme:

- nalezením průsečíků (rohových bodů), kde rovina ρ řízne čtyřstěn K_i,
- seřazením rohových bodů řezu,
- a vytvořením barevné mapy v RGB pro reprezentaci hodnot skalárního pole $u(\bar{x})$.

Výsledná obarvená mapa řezů n čtyřstěnů je vizualizace prostorové diskretizace.

2.1 Průsečík roviny s úsečkou

Má-li úsečka s rovinou právě jeden společný bod, pak je úsečka různoběžná s rovinou ρ, jejich společný bod nazýváme **průsečíkem** roviny s úsečkou.

Úsečka AB je část přímky definovaná mezi dvěma krajními body $A = [a_1, a_2, a_3]$ a $B = [b_1, b_2, b_3]$.

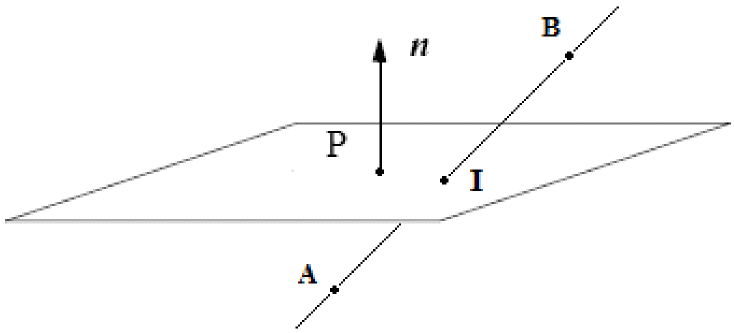
Body x, y, z ležící na úsečce AB jsou popsány **parametrickou rovnicí úsečky AB:**

$$x = a_1 + (b_1 - a_1)t$$

$$y = a_2 + (b_2 - a_2)t$$

$$z = a_3 + (b_3 - a_3)t$$

$$t \in \langle 0,1 \rangle$$



Obrázek 2 – Průsečík I roviny s úsečkou

Rovnice úsečky a roviny upravíme tak, abychom mohli vytvořit soustavu S čtyř rovnic o čtyřech neznámých. Jestliže je daná soustava **regulární** ($|\det S| \neq 0$) má právě jedno řešení. Singularitu matice soustavy testujeme podmínkou s danou numerickou přesností eps = 10^{-12} následovně: $|\det S| < eps$.

Soustava rovnic S pro nalezení průsečíku úsečky AB s rovinou ρ vypadá následovně:

$$x \qquad\qquad - (a_1 - b_1)t = a_1$$

$$y \qquad - (a_2 - b_2)t = a_2$$

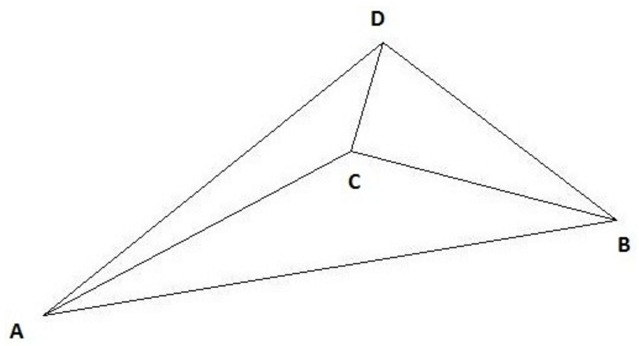
$$z - (a_3 - b_3)t = a_3$$

$$n_1x + n_2y + n_3z \qquad = n_1p_1 + n_2p_2 + n_3p_3$$

Vyřešením soustavy rovnic získáme bod **I** = $\begin{bmatrix} x \\ y \\ z \end{bmatrix}$, který je průsečíkem pokud $t \in \langle 0,1 \rangle$.

2.2 Řez čtyřstěnu rovinou

Při hledání rohových bodů řezu hledáme průsečík roviny ρ s jednotlivými úsečkami, z kterých se skládá čtyřstěn ABCD.
Čtyřstěn ABCD se skládá z šesti hran (z šesti úseček) - AB, AC, AD, BC, BD, CD.

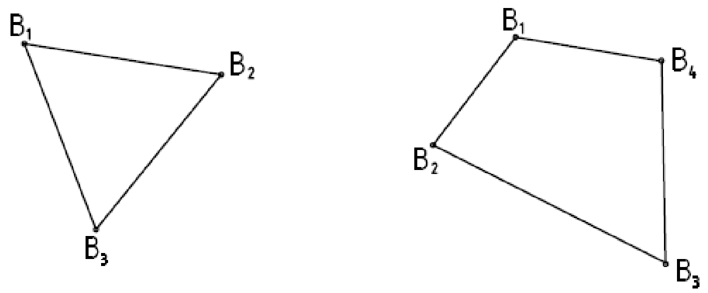


Obrázek 3 – Čtyřstěn ABCD

Dohromady musíme pro nalezení všech rohových bodů čtyřstěnu ABCD vytvořit 6 soustav rovnic pro nalezení průsečíků úseček AB, AC, AD, BC, BD, CD s rovinou ρ. Vyřešením všech soustav rovnic dostaneme rohové body řezu čtyřstěnu ABCD s rovinou ρ.

Případy, které mohou nastat při řezu čtyřstěnu rovinou, jsou tyto:

- rovina ρ řízne pouze vrchol čtyřstěnu,
- rovina ρ řízne hranu čtyřstěnu,
- rovina ρ řízne stěnu čtyřstěnu,
- rovina ρ řízne čtyřstěn ve 3 bodech $B_1,\ B_2,\ B_3$,
- rovina ρ řízne čtyřstěn ve 4 bodech $B_1,\ B_2,\ B_3,\ B_4$.

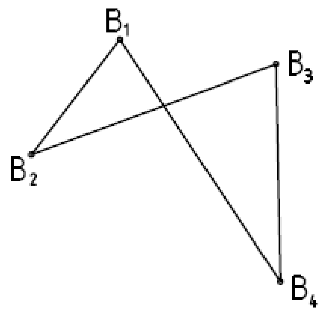


Obrázek 4 – Řez čtyřstěnu rovinou (ve 3 a 4 bodech)

Pro vizualizaci nás zajímají případy, kdy rovina ρ řízne čtyřstěn ABCD ve 3 nebo 4 bodech, ale bohužel i stěnu, kdy je matice soustavy singulární a máme nekonečně mnoho řešení.

2.3 Seřazení rohových bodů

Pro vizualizaci potřebujeme, aby nám vznikly z vypočtených rohových bodů řezů mnohoúhelníky (trojúhelníky a čtyřúhelníky). Tři rohové body B_1, B_2, B_3 řezu není třeba seřazovat, protože jejich spojením vždy vznikne trojúhelník. Pokud máme 4 rohové body B_1, B_2, B_3, B_4, může nastat případ (obrázek 5), kdy spojením bodů nevznikne čtyřúhelník, proto je třeba body setřídit následujícím postupem.



Obrázek 5 – Nesetříděné body

Body B_1, B_2, B_3, B_4 se nacházejí v jedné rovině (rovině řezu). Je možné body setřídit podle úhlů. Potřebujeme nějaký bod, který se nachází v jeho geometrickém středu, proto vypočteme těžiště T bodů B_1, B_2, B_3, B_4.

$$T = \frac{B_1 + B_2 + B_3 + B_4}{4}$$

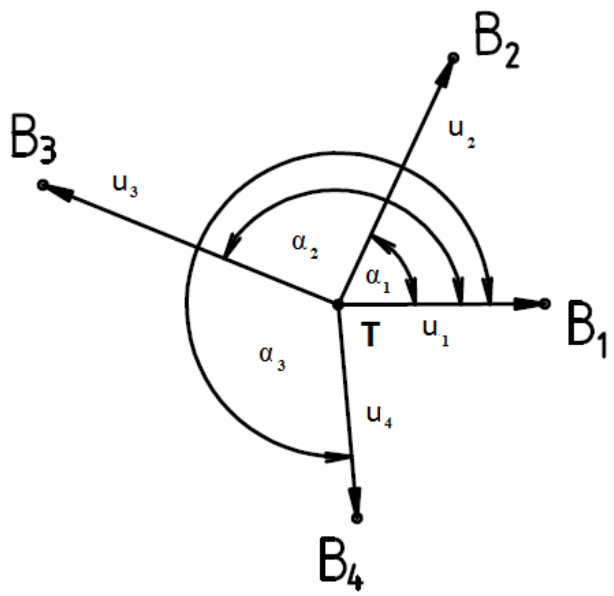
Vektory u_1, u_2, u_3, u_4 vedené z těžiště T do bodů B_1, B_2, B_3, B_4 .

$$u_1 = B_1 - T, \qquad u_2 = B_2 - T, \qquad u_3 = B_3 - T, \qquad u_4 = B_4 - T$$

Zvolíme si vektor u_1 jako **referenční vektor** a od něho budeme zjišťovat, jaké svírá úhly s vektory u_2, u_3, u_4 .

$$\alpha_1 = \measuredangle u_1 u_2 , \qquad \alpha_2 = \measuredangle u_1 u_3, \qquad \alpha_1 = \measuredangle u_1 u_4$$

K tomu abychom mohli správně určit velikosti úhlů, musíme znát kvadranty úhlů. Využijeme skalární součin pro zjištění znaménka kosinové složky úhlů α_1 , α_2 , α_3. a vektorový součin pro zjištění znaménka sinové složky úhlu velikosti úhlů α_1 , α_2 , α_3 .



Obrázek 6 – Body setříděné podle úhlů

Skalární součin využijeme pro zjištění znaménka kosinové složky úhlu svíraného dvojicí vektorů.

$$zc_1 = sgn\,(u_1, u_2)$$

$$zc_2 = sgn\,(u_1, u_3)$$

$$zc_3 = sgn\,(u_1, u_4)$$

Vektorový součin využijeme pro zjištění znaménka sinové složky úhlu svíraného dvojicí vektorů. Jelikož výsledkem vektorového součinu je vektor, zajímá nás jen sinová složka.

$$zs_{1(sin)} = \mathrm{sgn}(\,(u_1 \times u_2) * n\,)$$

$$zs_{2(sin)} = \mathrm{sgn}(\,(u_1 \times u_3) * n)$$

$$zs_{3(sin)} = \mathrm{sgn}(\,\big((u_1 \times u_4\big) * n)$$

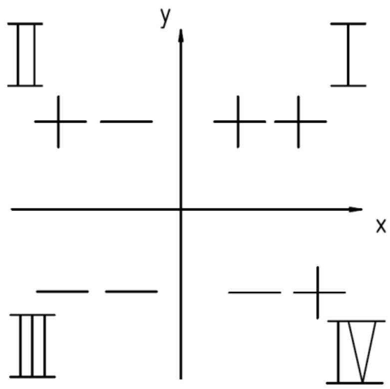
Vztah pro výpočet odchylky vektorů se skládá ze skalárního součinu vektorů v čitateli a součinu velikostí vektorů ve jmenovateli, odchylka vektorů je maximálně **180 stupňů**, tj. $\boldsymbol{\pi}$ v obloukové míře. Jsou-li u_1, u_2, u_3, u_4 libovolné nenulové vektory, pak pro úhly α_1, α_2, α_3 platí vzorce:

$$\cos\alpha_1 = \frac{(u_1, u_2)}{\| u_1 \|\| u_2 \|}, \qquad \alpha_1 \in \langle 0, \pi \rangle$$

$$\cos\alpha_2 = \frac{(u_1, u_3)}{\| u_1 \|\| u_3 \|}, \qquad \alpha_2 \in \langle 0, \pi \rangle$$

$$\cos\alpha_3 = \frac{(u_1, u_4)}{\| u_1 \|\| u_4 \|}, \qquad \alpha_3 \in \langle 0, \pi \rangle$$

Podle znaménka sinové a kosinové složky (Obrázek 7) určíme, do kterého kvadrantu úhel patří.



Obrázek 7 – Znaménka sinových a kosinových složek v jednotlivých kvadrantech

Například postup pro určení úhlu $\boldsymbol{\alpha_1 = \measuredangle u_1 u_2}$:

$\alpha_1 = \arccos(\frac{(u_1, u_2)}{\| u_1 \| \| u_2 \|})$, $\quad \alpha_1 \in \langle 0, \pi \rangle$

Uvažujme $\mathbf{sin}\ \boldsymbol{\alpha_1}$ jako sinovou složku úhlu α_1 a $\mathbf{cos}\ \boldsymbol{\alpha_1}$ jako kosinovou složku úhlu α_1. V následujícím algoritmu je uveden postup jak určit úhel α_1.

Algoritmus 1

if $\sin\alpha_1 > 0 \wedge \cos\alpha_1 > 0$, I. kvadrant
úhel $= \alpha_1$

elseif $\sin\alpha_1 > 0 \wedge \cos\alpha_1 < 0$, II. kvadrant
úhel $= \alpha_1$

elseif $\sin\alpha_1 < 0 \wedge \cos\alpha_1 < 0$, III. kvadrant
úhel $= 2\pi - \alpha_1$

elseif $\sin\alpha_1 < 0 \wedge \cos\alpha_1 > 0$, IV. kvadrant
úhel $= 2\pi - \alpha_1$

elseif $\sin\alpha_1 = 0 \wedge \cos\alpha_1 > 0$
úhel $= 0$

elseif $\sin\alpha_1 = 0 \wedge \cos\alpha_1 < 0$
úhel $= \pi$

elseif $\sin\alpha_1 > 0 \wedge \cos\alpha_1 = 0$
úhel $= \frac{\pi}{2}$

$\boldsymbol{elseif}\ \sin\alpha_1 < 0 \land \cos\alpha_1 = 0$

$$\text{úhel} = \frac{3\pi}{2}$$

end.

Tímto způsobem zjistíme i velikosti α_2 , α_3. Jakmile známe všechny úhly α_1 , α_2 , α_3 můžeme podle jejich velikosti seřadit body B_1, B_2, B_3, B_4 tak, aby po jejich spojení vznikl čtyřúhelník.

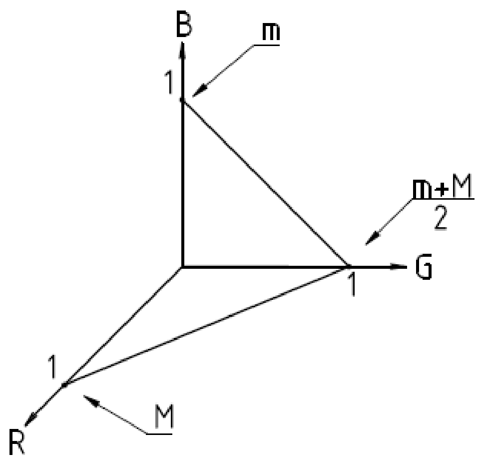
2.4 Tvorba barevné mapy

Abychom mohli dané řezy čtyřstěnů obarvit, a tím reprezentovat hodnoty skalárního pole $u(\bar{x})$ vytvoříme barevnou mapu v RGB.
Barevný model RGB (Red, Blue, Green) je metoda pro reprodukci barev. Je to aditivní způsob míchání barev. Kombinací dvou základních barev vzniká barva sekundární.

Chceme vytvořit barevnou mapu, která bude přecházet z modré barvy na zelenou barvu a pak na barvu červenou.
Uvažujme barvu určenou uspořádanou trojicí (R, G, B), kde $\mathbf{R}, \mathbf{G}, \mathbf{B} \in \langle \mathbf{0}, \mathbf{1} \rangle$ jsou složky červené, zelené a modré barvy.
Pokud známe minimální hodnotu a maximální hodnotu skalárního pole $u(\bar{x})$, vytvoříme barevnou mapu, kterou realizujeme pomocí lineárních funkcí.

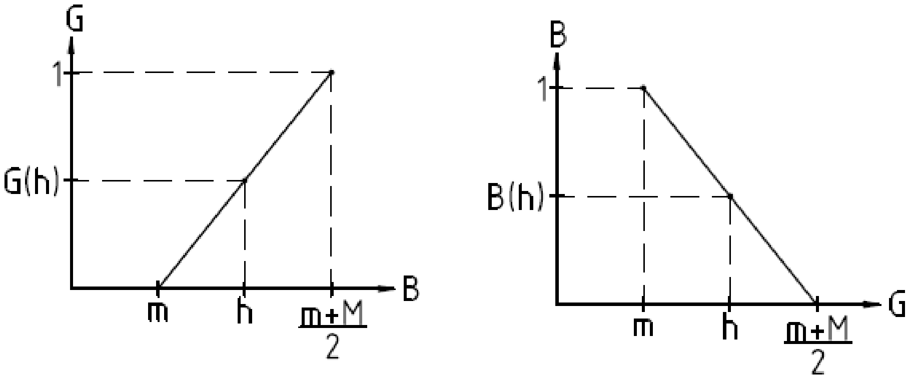
- $m = \min\ (u(\bar{x}))$ odpovídá modré barvě (RGB = [0,0,1])
- $M = \max\ (u(\bar{x}))$ odpovídá červené barvě (RGB = [1,0,0])
- $\frac{m+M}{2}$ odpovídá zelené barvě (RGB = [0,1,0])

Tyto hodnoty skalárního pole $u(\bar{x})$ převedeme lineárními funkcemi na uspořádanou trojici (R, G, B).

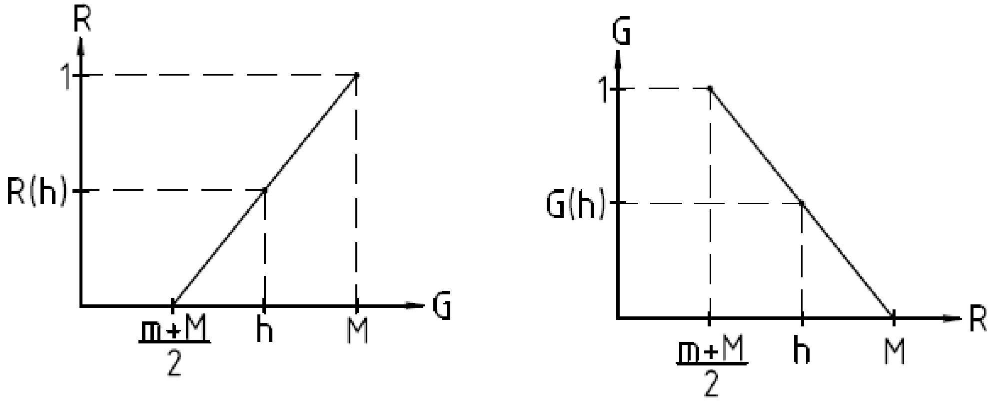


Obrázek 8 – Přechod Modrá → Zelená → Červená

Pomocí barevné mapy přiřadíme hodnotě h skalárního pole $u(\bar{x})$ barvu v RGB. Pokud se hodnota h nachází v první polovině intervalu ($h \leq \frac{m+M}{2}$), pak její barva bude kombinací modré a zelené. Je-li hodnota h v druhé polovině intervalu ($h > \frac{m+M}{2}$), pak bude barva přecházet od barvy zelené po barvu červenou.



Obrázek 9 – Lineární funkce G → B a B → G



Obrázek 10 – Lineární funkce R → G a G → R

V následujícím algoritmu je uveden postup jak vytvořit dané lineární funkce pro přechod barev v RGB.

Algoritmus 2

if $h \leq \frac{m+M}{2}$

$R = 0$

$G = \dfrac{2}{(M-m)} * (h-m)$

$B = -\frac{2}{(M-m)} * (h - \frac{m+M}{2})$

else

$B = 0$

$R = \frac{2}{(M-m)} * (h - \frac{m+M}{2})$

$G = -\frac{2}{(M-m)} * (h - M)$

end.

Takto vytvořená barevná mapa nám umožní lineární (plynulý) přechod barev a bude reprezentovat hodnoty skalárního pole $u(\bar{x})$.

V této chvíli máme vše potřebné proto, abychom mohli dané skalární pole vizualizovat.

3 GUI

Grafické uživatelské rozhraní (**G**raphical **U**ser **I**nterface) je prostředkem, který umožňuje ovládat napsaný program pomocí grafických ovládacích prvků. Na monitoru počítače jsou zobrazena okna, ve kterých programy zobrazují svoje výsledky (grafy, číselné hodnoty, tabulky). Uživatel používá k ovládání grafické vstupní prvky, jako jsou menu, tlačítka, posuvníky, tabulky, formuláře a podobně.

3.1 Tvorba GUI (MATLAB)

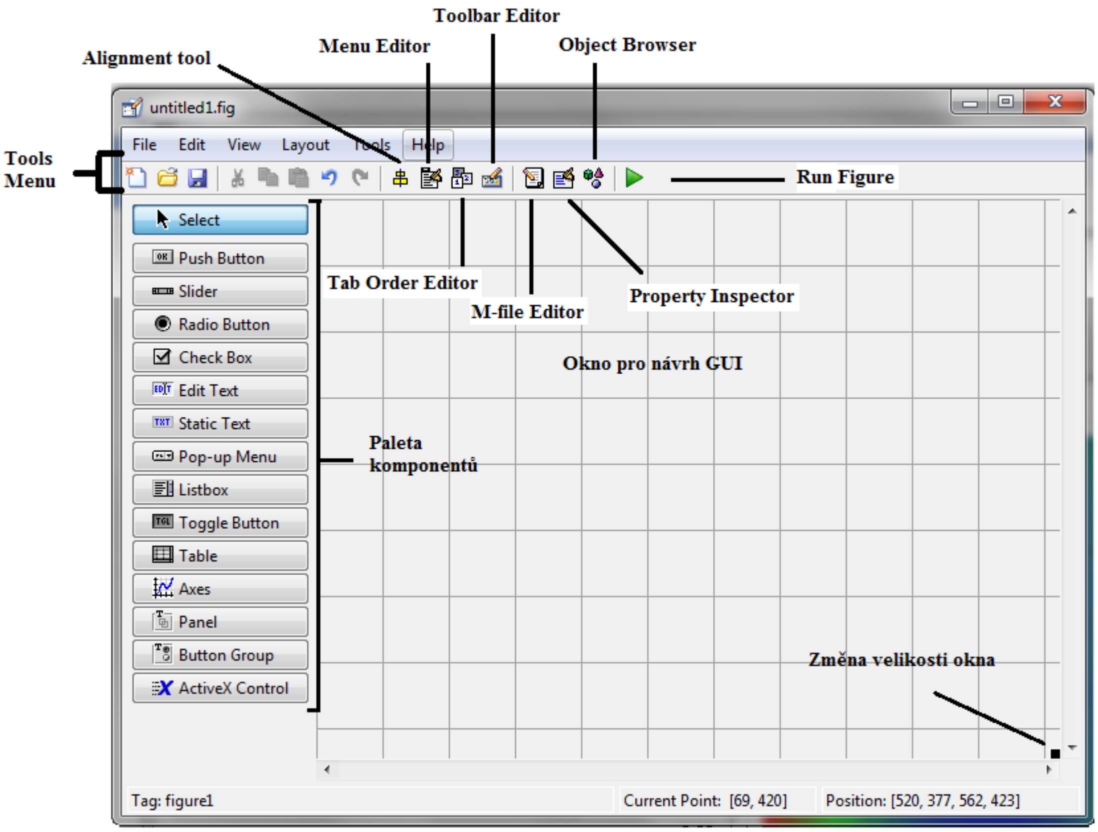
Tvorba GUI je součástí vývojových a programovacích nástrojů Matlabu a provádí se v matlabovském vývojovém prostředí **GUIDE** (Graphical User Interface Development Environment), které umožňuje vytvářet a editovat uživatelské rozhraní, a to prostřednictvím následujících komponentů (aktivních ovládacích prvků), například to jsou:

- **push buttons –** tlačítka,
- **sliders –** posuvníky,
- **radio buttons –** přepínače,
- **check boxes –** zaškrtávací políčka,
- **edit texts –** editovací texty,
- **static texts –** statický texty,
- **pop-up menus –** vyskakovací menu,
- **tables –** tabulky a
- **axes –** osy.

GUIDE obsahuje:

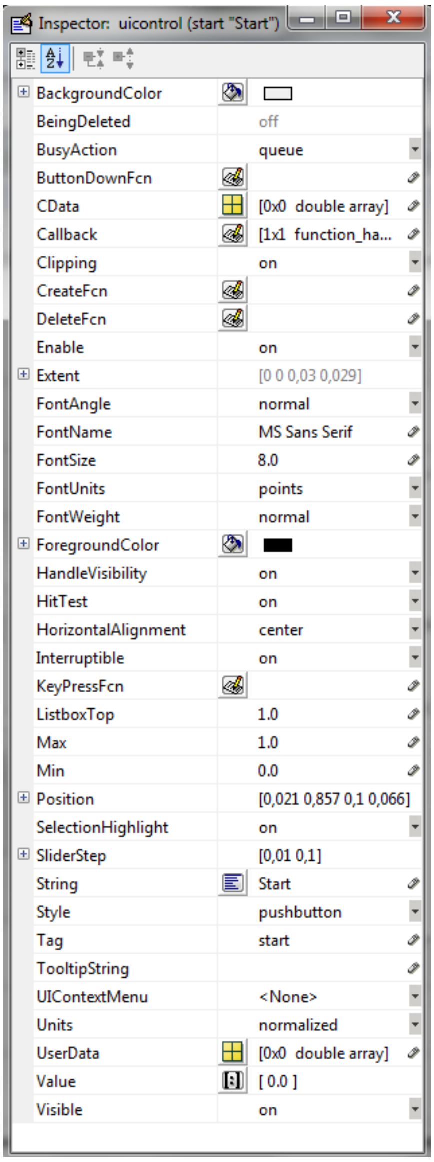
- **Layout Editor –** přidání a základní uspořádání objektů v okně návrhu
- **Object Browser –** sledování hierarchické struktury Handle Graphics objektů
- **Property Inspector –** nastavení hodnot objektů
- **Alignment Tool –** zarovnání a rozmístění objektů
- **Menu Editor –** vytváření menu
- **Toolbar Editor –** vytváření lišty s nástroji

Základní ovládací nastroj pro tvorbu GUI je **Layout Editor** (obrázek 11), který umožňuje vybírat GUI komponenty z palety a uspořádat je v okně pro návrh GUI. K jeho startu slouží příkaz *guide,* který zadáme v hlavním příkazovém okně Matlabu.



Obrázek 11 – Layout editor

Funkční GUI se generuje aktivováním návrhu vytvořeného v Layout Editoru vybráním **Run Figure** (zelené tlačítko na obrázku 11) položky v Tools menu. Matlab se tím postará o uložení souboru typu ***.fig** (fig-soubor), kde jsou uložené grafické objekty, jejich rozměr a poloha v rámci aplikace. Funkční stránka aplikace se ukládá do druhého souboru. Ten má příponu ***. m** (m-soubor) a nacházejí se v něm všechny funkce, které se starají o správný chod aktivních prvků za pomocí funkcí a skriptů.
Aktivní ovládací prvky mají množinu vlastností jako poloha, barva, popis, velikost písma, název (Tag), rozsah a podobně. Pro nastavení těchto vlastností slouží **Property Inspector** (obrázek 12). Jeho okno vyvoláme dvojklikem na příslušný aktivní prvek. Důležitá vlastnost každého použitého objektu je název (Tag), který jednoznačně identifikuje aktivní prvek. Pod tímto názvem se automaticky generuje kód v m-souboru.
Pro každý aktivní prvek, který jsme umístili v oknu pro návrh GUI se generuje v m-souboru funkce s příponou ***_Callback***. Prefix funkce tvoří právě název aktivního prvku, např. *Start _Callback*. V této funkci se nastavuje to, co se provede po aktivování daného prvku.
Odevzdávání dát mezi jednotlivými funkcemi probíhá pomocí datové struktury ***handles***. K této speciální proměnné je možné přistupovat v každé funkci. Ke každému vlastnosti prvku a k její hodnotě je možné přistupovat takovou konstrukcí *handles.NázevObjektu (názevVlastnosti).*
Díky těmto základním postupům můžeme vytvořit funkční GUI v matlabu.

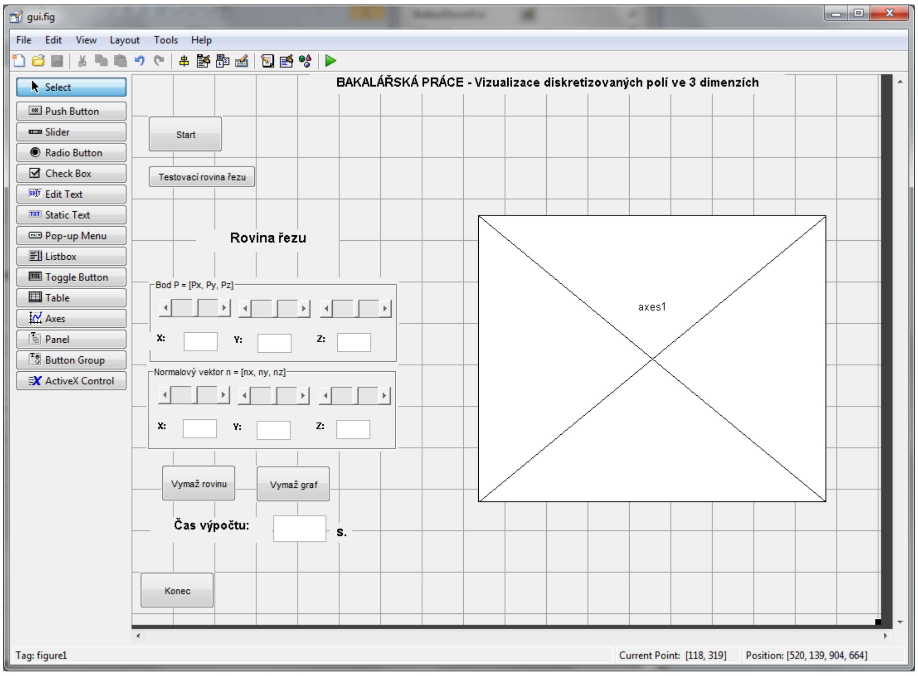


Obrázek 12 – Property Inspector

3.2 Vlastní tvorba GUI

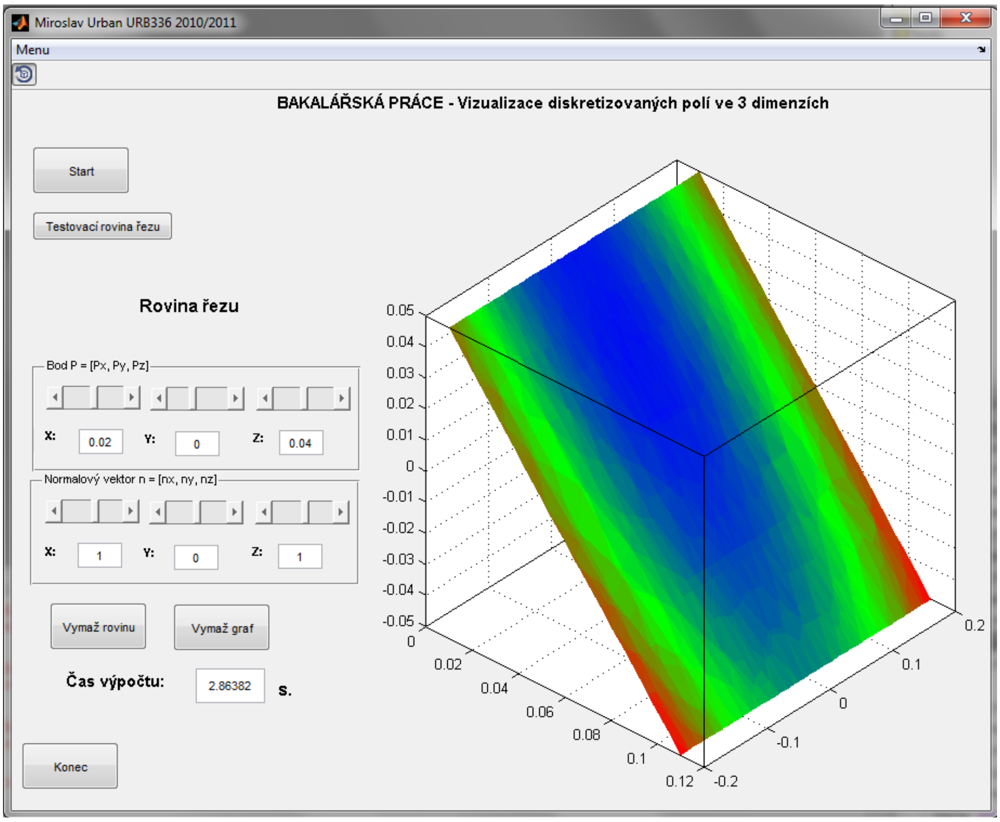
GUI pro vizualizaci diskretizovaných polí se skládá z:

- **tlačítek** – pro start programu, pro nastavení hodnot roviny, pro vymazaní hodnot roviny, pro vymazání grafu a pro ukončení GUI,
- **posuvníků** – pro nastavení hodnot roviny,
- **editovacího textu** – pro výpis času výpočtu a zobrazení hodnot posuvníků,
- **statických textu** – pro popisy jednotlivých částí GUI a
- a z **os** – pro zobrazení diskretizovaných polí.



Obrázek 13 – Rozvrhnutí komponentů GUI

Takto vytvořené GUI pak po aktivování vypadá následovně:



Obrázek 14 – Aktivované GUI

4 Příklady vizualizovaných polí

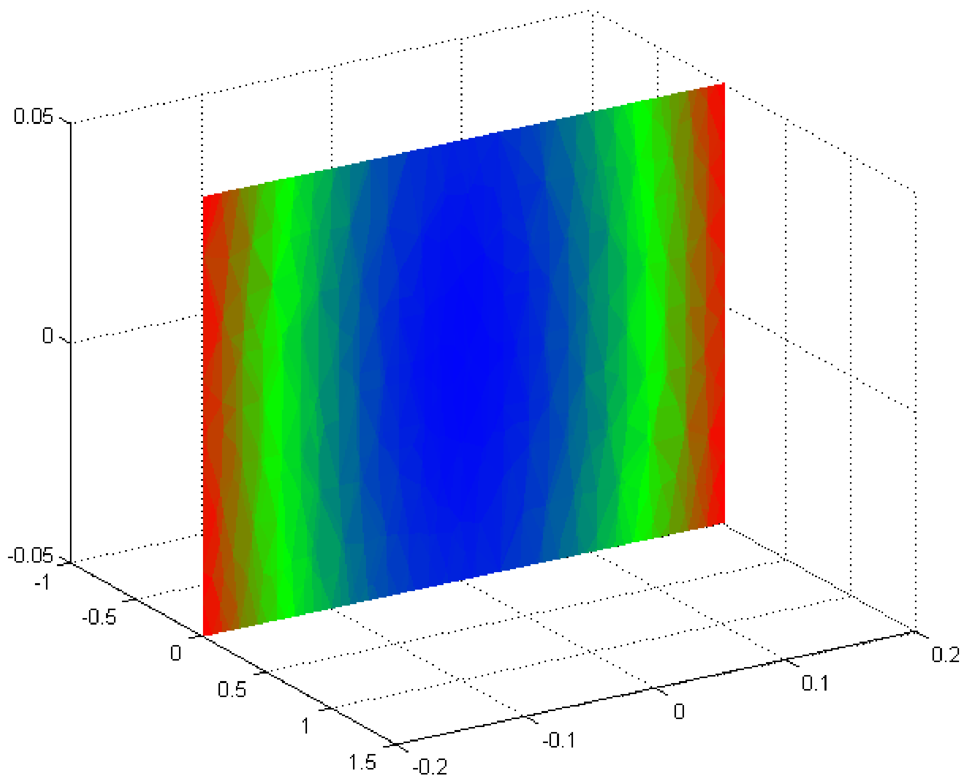
V této kapitole si ukážeme dva příklady vizualizovaných polí.

4.1 Vizualizace analytické funkce

Mějme sít čtyřstěnů, která tvoří krychli $\Omega := (0,1)^3$ a nad touto sítí si zobrazíme funkci, která má následující tvar:

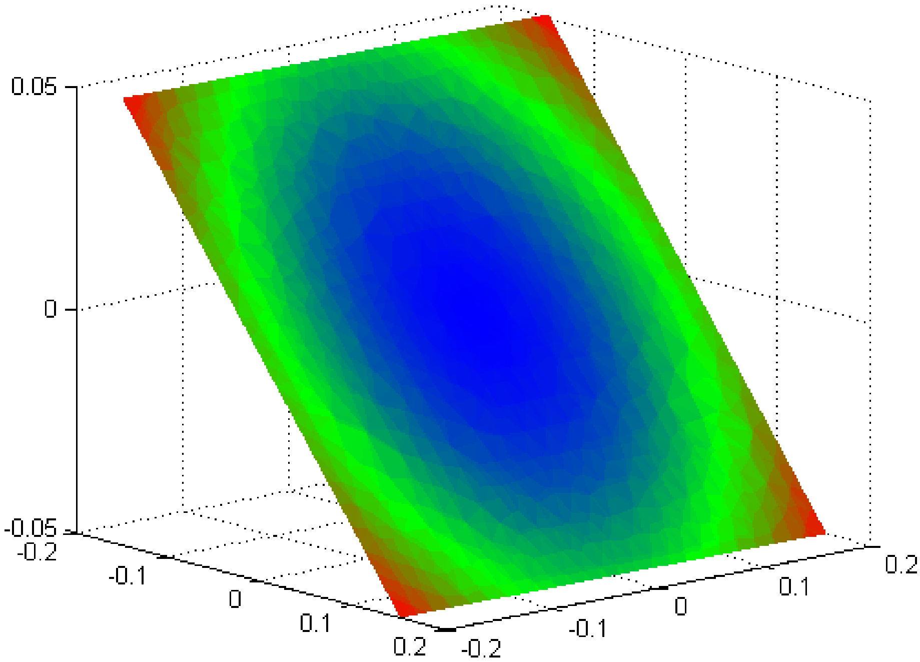
$$f(x,y,z) = x^2 + y^2 + z^2$$

Na obrázku 15 je řez rovinou ρ, která je zadána bodem $P = [0.02, 0, 0]$ a normálovým vektorem $n = [1, 0, 0]$.



Obrázek 15 – Vizualizace analytické funkce 1

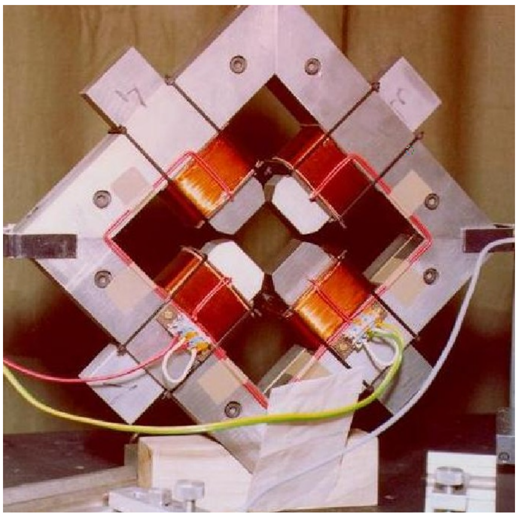
Na obrázku 16 je řez jinou rovinou ρ, která je zadána bodem $P = [0, 0, 0]$ a normálovým vektorem $n = [1, 0, 3]$.



Obrázek 16 – Vizualizace analytické funkce 2

4.2 Zobrazení magnetického pole elektromagnetu

Mějme diskretizaci elektromagnetu z obrázku 17.

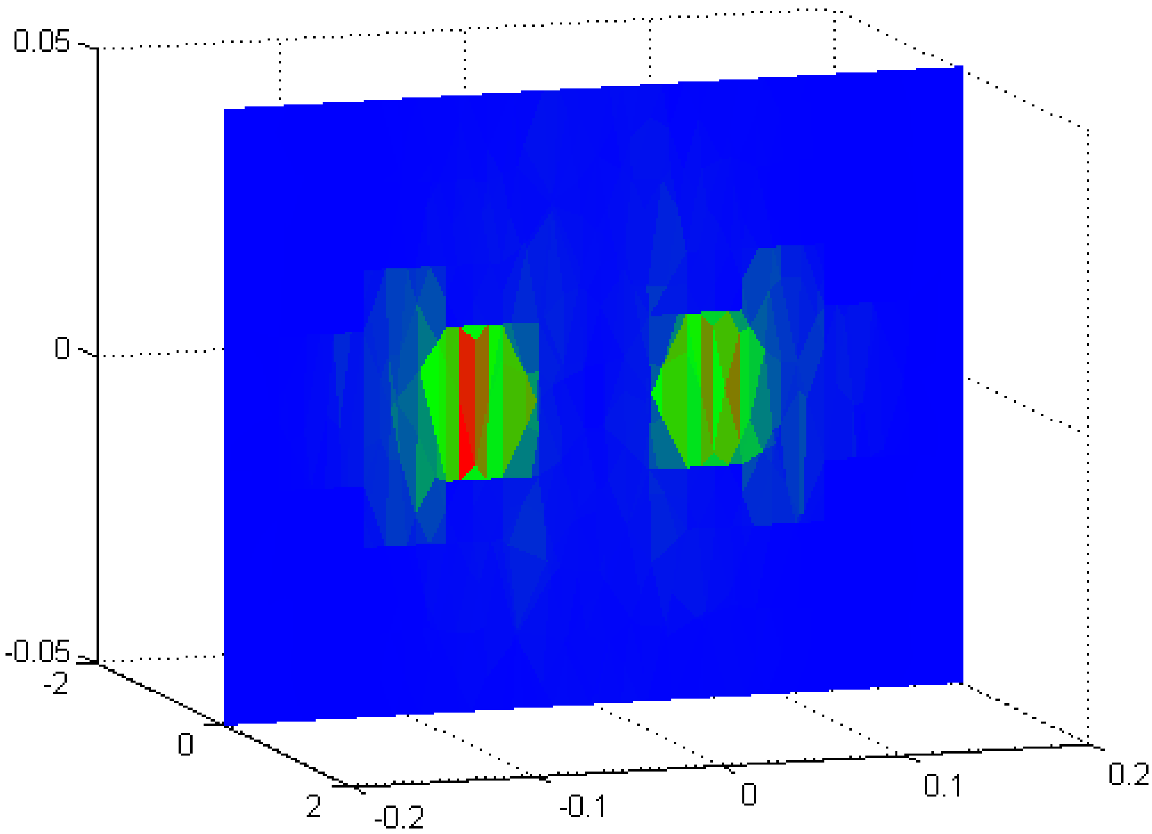


Obrázek 17 – Elektromagnet

Pro každý čtyřstěn K máme danou magnetickou indukci $\overline{B^k} = \left(B_x^k,\ B_y^k,\ B_z^k\right)$ z níž vypočteme po čtyřstěnech konstantní velikost $B^k = \sqrt{\left(B_x^k\right)^2 + \left(B_y^k\right)^2 + \left(B_z^k\right)^2}$.

Na obrázku 18 je vykreslena funkce B v řezu rovinou, která je zadána:

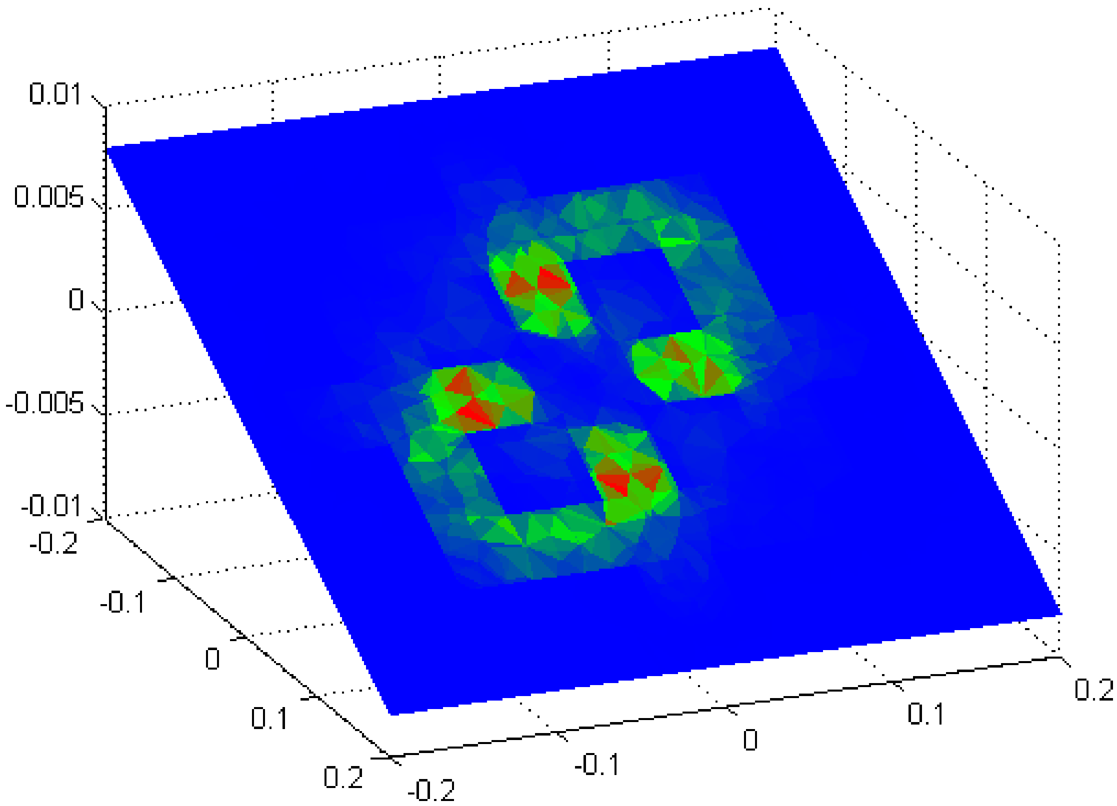
$$P = [0, 0, 0] \text{ a } n = [1, 0, 0].$$



Obrázek 18 – Vizualizace diskretizace elektromagnetu 1

Na obrázku 19 je vykreslena funkce B v řezu rovinou, která je zadána:

$$P = [0, 0, 0] \text{ a } n = [0.4, 0, 10].$$



Obrázek 19 – Vizualizace diskretizace elektromagnetu 2

Takto zobrazené diskretizace elektromagnetu nám ukazují silové účinky elektromagnetického pole, které určuje elektromagnetická indukce B.

5 Závěr

Podařilo se nám vytvořit software v programovém prostředí MATLAB pro vizualizaci prostorových diskretizovaných polí, který slouží pro zobrazení výsledků praktických úloh.

V bakalářské práci jsme formulovali algoritmy potřebné pro vizualizaci prostorových polí a popsali jsme tvorbu grafického uživatelského rozhraní s aplikací na náš software.

Literatura

1. MATLAB® 7 Creating Graphical User Interfaces
< http://www.mathworks.com/help/pdf_doc/matlab/buildgui.pdf >.

2. JIRÁSEK, František; BENDA, Josef . *Matematika pro bakalářské studium*. [s.l.] : Ekopress, 2006.

Adresářová struktura přiloženého disku

/bc	bakalářská práce ve formátu pdf.
/matlab	zdrojové kódy v programu Matlab.