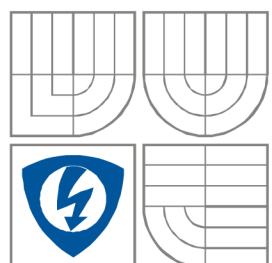


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ

ÚSTAV RADIOELEKTRONIKY

FACULTY OF ELECTRICAL ENGINEERING AND
COMMUNICATION

DEPARTMENT OF RADIO ELECTRONICS

INTERNETOVÁ UČEBNICE: METODA KONEČNÝCH DIFERENCÍ V ČASOVÉ OBLASTI

INTERNET TEXTBOOK: TIME DOMAIN FINITE DIFFERENCES

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

Michal Martinů

VEDOUCÍ PRÁCE
SUPERVISOR

prof. Dr. Ing. Zbyněk Raida

BRNO, 2009

ABSTRAKT

V bakalářské práci jsem se zaměřil na vytvoření programu pro demonstraci principů metody konečných diferencí v časové oblasti. Principy metody jsou demonstrovány skriptem pro programové prostředí Matlab. Skript je doplněn standardním uživatelským rozhraním Windows. Programátorský a uživatelský popis skriptu má formu internetových stránek. Mým úkolem bylo vytvořit část internetové učebnice se všemi vrstvami a zaručit kompatibilitu se stávající verzí. Jak v českém, tak i anglickém jazyku.

KLÍČOVÁ SLOVA

Konečné diference v časové oblasti, Matlab, Internetová učebnice.

ABSTRACT

The project is focused on the development of the computer program demonstrating principles of the time-domain finite difference method. Principles of the method are demonstrated by a script for Matlab. The script is completed by the standard user's interface of Windows. Programmer's guide and user's guide are of the form of Internet pages. I have to create a part of internet textbook with all of layers and warrant a compatibility with existing version. In both languages, Czech and English.

KEYWORDS

Time domain finite differences, Matlab, Internet textbook.

BIBLIOGRAFICKÁ CITACE

MARTINŮ, M. *Internetová učebnice: metoda konečných diferencí v časové oblasti*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2009. 32 s. Vedoucí semestrální práce prof. Dr. Ing. Zbyněk Raida

Prohlášení

Prohlašuji, že svoji bakalářskou práci na téma [Internetová učebnice: metoda konečných diferencí v časové oblasti](#) jsem vypracoval samostatně pod vedením vedoucího semestrálního projektu 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 uvedeného semestrálního projektu dále prohlašuji, že v souvislosti s vytvořením tohoto projektu 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 § 152 trestního zákona č. 140/1961 Sb.

V Brně dne [5. června 2009](#)

.....
podpis autora

Poděkování

Děkuji vedoucímu semestrálního projektu [prof. Dr. Ing. Zbyňkovi Raidovi](#) za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mého semestrálního projektu.

V BRNĚ DNE [5. ČERVNA 2009](#)

.....
PODPIS AUTORA

OBSAH

Seznam obrázků.....	6
1 Úvod.....	7
1.1 Popis internetové učebnice.....	8
2 Teorie programování v HTML.....	9
2.1 Teorie skriptů.....	10
2.2 Kaskádové styly a jejich použití.....	11
2.3 Dědičnost v CSS.....	12
2.4 PHP skripty a jejich historie.....	13
3 Popis metody.....	15
4 Popis vrstev.....	22
4.1 Vrstva A.....	22
4.2 Vrstva B.....	23
4.3 Vrstva C.....	23
4.4 Vrstva D.....	24
5 Popis uživatelského rozhraní.....	25
5.1 Stručný popis grafického rozhraní.....	25
5.2 Editace parametrů.....	26
5 Závěr.....	27
6 Literatura.....	28

Seznam obrázků

- Obr. 1.1 Typická stránka internetové učebnice
- Obr. 3.1 Sestavení soustavy rovnic do matic a přidělení souřadnic uzlům
- Obr. 3.2 Výběr vhodné množiny uzlů.
- Obr. 3.3 Buňka prostorové diskretizační sítě FDTD v kartézské souřadné soustavě.
- Obr. 4.1 Ukázková stránka vrstvy A.
- Obr. 4.2 Ukázková stránka vrstvy B.
- Obr. 4.3 Ukázková stránka vrstvy C.
- Obr. 4.4 Ukázková stránka vrstvy D.
- Obr. 4.5 Ukázková stránka vrstvy E.
- Obr. 4.6 Ukázková prostředí Netbeans.
- Obr. 4.7 Ukázková stránka vrstvy F.
- Obr. 5.1 Úvodní vzhled uživatelského rozhraní.
- Obr. 5.2 Vzhled spuštěné aplikace pro vid TE.

1 Úvod

Bakalářská práce se zabývá rozšířením *Multimediální učebnice elektromagnetických vln a mikrovlnné techniky*

<http://www.feec.vutbr.cz/~raida/multimedia>

Na této učebnici pracuje řada odborníků pod vedením prof. Dr. Ing Zbyňka Raidy.

Učebnice je založena na názornosti a jednoduchém ovládní. Učebnici může snadno používat i člověk, který nemá zkušenosti s multimediální internetovou vzdělávací literaturou.

Učebnice si klade za cíl jednoduchou a dobře pochopitelnou formou vysvětlit čtenářům základy řešení Maxwellových rovnic pro nejčastěji používané elektromagnetické struktury. Ne každý totiž ovládá použití Maxwellových rovnic v diferenciálním nebo v integrálním tvaru a jejich aplikaci.

2.1 Difrakce na rovinné pohlcující překážce
Základní teorie

Budeme se zabývat **difrakcí** vln v následující situaci. Mezi zdrojem vlnění V a místem příjmu P je tenká rovinná přepážka, která elektromagnetické vlny nepropouští ani neodráží (tzv. pohlcující přepážka). V přepážce je vyříznuto okénko nebo jiný otvor, kterým může vlnění procházet do oblasti místa příjmu. Přepážka může být také polorovina, jak je nakresleno na obr. 2.1A.1 (okénkem je pak celá horní polorovina). Úkolem je vypočítat intenzitu pole v bodu P .

Popsaná situace je aproximací některých v praxi se vyskytujících úloh, např. šíření vln přes horský hřbet. Připomeňme, že při aplikacích v radiotechnice v souvislosti s šířením vln v terénu se místo termínu *přepážka* používá termín *překážka* (terénní překážka). Zde budeme používat oba termíny ve stejném významu.

Takto formulovanou úlohu řešil poprvé A.J. Fresnel (čti "Frenel", 1788 - 1827) poměrně jednoduchým způsobem, který nyní vysvětlíme. Pro Fresnelův způsob řešení difrakční úlohy se někdy používá termín *Fresnelova difrakce*.

Rovinu přepážky si označme S a budeme předpokládat, že je kolmá na spojnici VP . Konkrétně budeme sledovat difrakci na přepážce - polorovině (obr. 2.1A.1), ale postup lze

Obr. 2.1A.1 Difrakce na polorovině

Obr. 1.1 Typická stránka internetové učebnice.

Učebnice se snaží na konkrétních příkladech vysvětlovat praktický význam složitých matematických a fyzikálních jevů. Učebnice chce názorně ukázat podstatu zkoumaných jevů a z ní vyplývající principy fungování běžných elektromagnetických struktur.

1.1 Popis internetové učebnice

Typická stránka Multimediální učebnice je znázorněna na obr. 1.1. K základní práci s učebnicí slouží pět ovládacích tlačítek v pravém horním rohu okna:

- Tlačítkem CZ/EN přepínáme mezi českou a anglickou verzí učebnice. Anglická verze učebnice slouží zejména zahraničním studentům a čtenářům, kteří se chtějí zdokonalit v odborné angličtině.
- Tlačítka ZPĚT a DOPŘEDU jsou obdobou tlačítek, která známe z internetových prohlížečů. Po stisknutí se vrátíme na předchozí stránku nebo se posuneme na stránku, kterou jsme sledovali před chvílí.
- Stisknutím tlačítka INDEX se vrátíme na úvodní stránku učebnice
- Tlačítko PDF verze převede aktuální stranu učebnice do formátu PDF.

Hlavní menu se nachází v levé části okna (viz obr. 1.1). Kliknutím na menu rozbalujeme jednotlivé vrstvy vybraného tématu. Na výběr má čtenář nejprve základní nabídku, následně lze vybírat z jednotlivých vrstev kapitoly.

Každá kapitola, je rozdělena do šesti vrstev:

- A obsahuje jednoduchý, slovní popis probíraného jevu.
- B prezentuje složitější matematický popis probíraného jevu.
- C je uživatelskou příručkou skriptů Matlabu, které si může čtenář stáhnout, aby jejich prostřednictvím lépe pochopil studované jevy.
- D je programátorskou příručkou programů.
- E obsahuje simulační programy ve formě javovských appletů.
- F obsahuje pětice kontrolních otázek, na kterých si může čtenář ověřit správné pochopení látky.

Mým úkolem je obohatit multimediální učebnici o metodu konečných diferencí v časové oblasti. Je tedy zapotřebí vytvořit uživatelské rozhraní k m-souborům v programu MATLAB a sestavit odpovídající vrstvy C a D.

2 Teorie programování v HTML

Tím, čím se internetové dokumenty výrazně odlišují od ostatních dokumentů, je používání hypertextu. Ten umožňuje provázat internetové stránky mezi sebou. Pokud tedy aktivujeme odkaz na jedné stránce, zobrazí se nám jiný, související dokument, ale třeba i obrázek, zvukový soubor, animace nebo soubor určený ke stažení. Odkazem může být nejčastěji text, obrázek nebo jiný grafický objekt.

K vytváření hypertextových dokumentů se používá jazyk HTML (Hypertext Markup Language). Základem stránky je prostý text. Teprve vložením určitých příkazů do textu dosáhneme jeho správného umístění, barvy či velikosti. Právě tak můžeme vytvářet i odkazy a dokumenty propojovat. Soubory HTML mají standardně koncovku *.html nebo častěji *.htm. Kratší verze se prosadila až později díky konvencím používaným v operačním systému Windows.

Hypertextové dokumenty lze otevřít v prohlížeči (browseru), který příkazy interpretuje do grafické podoby. Protože prohlížeče nabízí více firem, není ani interpretace stránek v těchto prohlížečích jednotná. Nehledě k tomu, že často obsahují některé výraznější odlišnosti nebo nadstandardní funkce, které jiné postrádají. A také proto, že se jazyk HTML vyvíjí spolu s technickou úrovní výpočetní techniky, vyvíjí se i prohlížeče.[5.1]

Důsledkem je to, že uživatelé používají různé prohlížeče v různých verzích, navíc v různém rozlišení obrazu při různém nastavení svých browserů. Není jednoduché napsat stránku tak, aby byla zobrazena ve všech případech tak, jak autor zamýšlel. Proto je důležité, vyvarovat se používání těchto nestandardních funkcí a konstrukcí.

I proto by při tvorbě stránky měl být kladen důraz na její logickou strukturu. Nehledě k tomu, že stránky mohou být prohlíženy negrafickými browsery (např. i pro nevidomé). Proto je důležitější aby prohlížeč poznal, že se jedná např. o hlavní nadpis ap. Podle úrovně nadpisu se orientují také i internetové vyhledávače. Strukturu stránky a její formátování zajišťují speciální příkazy. Prohlížeč (browser) stránky musí umět rozlišit text určený k formátování od těchto příkazů. Pro odlišení jsou příkazy - značky ohraničeny tzv. šipovými závorkami "< >". (Ty lze jednodušeji na klávesnici napsat pomocí zkratky <Alt> + <, > nebo <Alt> + <.>).

Některé příkazy - tagy [tegy] - značky, jsou párové (ohraničují tak vybraný text) a jiné ne. Párové příkazy jsou ukončeny, resp. uzavírací příkaz obsahuje lomítko. Např.: <tag> formátovaný text </tag>

Můžete se setkat taky s tím, že jsou někdy uzavírací znaky vynechány. To je sice v zásadě možné (např. tabulkové příkazy, ukončení odstavce), ale může nastat případ, kdy to nedopadne dobře. To záleží také na "inteligenci" prohlížeče. Proto raději všechna pravidla dodržujte a předejete tak možným chybám a různým interpretacím v různých prohlížečích[5.3].

Některé tagy mají navíc různé atributy, které se zapisují dovnitř příkazu. Těmito atributy jsou nejčastěji příkazy pro zarovnání, barvu text, jeho velikost ap. Hodnota atributu se zapisuje mezi uvozovky, které jsou nezbytné především u víceslovných hodnot. Např.: <P align="right"> libovolný text </P>

Toto znamená, že text začíná na novém odstavci (P), a je zarovnán (align) napravo (right), pro další text to již neplatí(</P>) a je zarovnáván jako obvykle vlevo. Atributy se nemusí psát všechny (obvykle je jich víc), stačí jen ty, které potřebujeme.

Strukturu či sazbu textu tedy ovlivňují speciální příkazy (párové, nepárové, s atributy). Pokud použijeme následující konstrukci příkazu: <!-- libovolný text, příkaz, komentář -->, dosáhneme toho, že je text nebo příkaz ignorován. Je totiž interpretován jako komentář. To se dá využít např. při ladění stránky, nebo pro zapsání poznámek. Musí se však dát pozor na správné zakončení příkazu.

Příkazy ani jejich atributy nejsou "case sensitiv", to znamená, že je celkem jedno, jestli se budou psát velkými nebo malými písmeny. Ale pro lepší přehlednost je lépe psát příkazy velkými písmeny. Je důležité si uvědomit, že prohlížeče ignorují zarovnání textu tak jak je ve zdrojovém souboru. Ignoruje tedy odstavce a odsazení. Zalomení řádku "entrem" (konec řádku), prázdné řádky, dvojité a vícenásobné mezery jsou interpretovány jako jedna mezera.

2.1 Teorie skriptů

Pomocí skriptovacích jazyků lze stránky oživit, umožňují uživateli ovlivnit jejich obsah, naprogramovat jednoduché hry, atd. Stránky tak přestávají být pouze statickým objektem, ale stávají se dynamickými interaktivními stránkami reagující na volby a potřeby uživatele. Skriptovací programy mohou být součástí zdrojového kódu stránky, a pak se program vykonává přímo na stránkách u uživatele, nebo mohou být použity programy, které jsou uloženy na serveru. V druhém případě program na serveru vyhodnocuje data zasláná uživatelem a zpět odesílá již speciálně sestavené stránky a data pro klientův prohlížeč. Nejčastějším případem takovýchto programů je zpracovávání dat z vyplněných formulářů a práce s databázemi na serveru. Neznámější programy jsou **ASP** (Active Server Pages), **CGI** (Common Gateway Interface) a **PHP** (Hypertext Preprocessor). [7.5]

Mezi běžnými uživateli se ale nejvíce používá první skupina programovacích prostředků. Tedy ty skripty, které jsou součástí zdrojového kódu stránky. Nejrozšířenější jsou Visual Basic Script, ale hlavně Java Script, který nás bude zajímat především. Jedná se o celkem nenápadný prostředek, ale má rozsáhlé možnosti. Pomocí skriptu lze manipulovat s obsahem textu na stránkách, používat matematické výpočty, pracovat s grafikou, ale i ovlivňovat chování prohlížeče, jako je zobrazování různých informativních a varovných oken, ovlivňování zobrazení nástrojových a funkčních lišt, nebo otvírání a zavírání nových oken prohlížeče s určeným obsahem ap. Zde je jeden malý příklad. Funkce použitá v odkazu otevře nové okno s uvedeným obsahem podle definice ve skriptu.

Nové okno bude bez lišty "toolbar", bude mít rozměry 600x80 a bude 50 pixelů od levého okraje.

```
<SCRIPT language="JavaScript">
<!--//
function openWindow() {
window.open("", "nove", 'toolbar=0,scrollbars=1,width=600,height=80,resizable=1,left=50');
}
//-->
</SCRIPT>
<A HREF="sample.html" onClick="openWindow()" target="_blank">odkaz</A>
```

2.2 Kaskádové styly a jejich použití

Kaskádové styly (angl. Cascading Style Sheets, zkratka CSS) jsou nadstavbou jazyka HTML určenou k definování vzhledu WWW stránek. V současnosti je platnou normou verze CSS2 a alespoň její jádro je podporováno většinou nových grafických prohlížečů (MSIE 5+, Netscape 6+, Mozilla, Opera 5+ a další). Většina prohlížečů ale podporuje jen určité funkce CSS2, případně pouze CSS1; obecně ještě není CSS2 plně podporováno nikde a v práci s CSS obsahují navíc prohlížeče mnoho chyb.[6.2.8]

CSS umožňují přiřadit každému prvku v HTML určité vlastnosti (vzhled, způsob zobrazení či jiné prezentace, umístění atd.), a to v závislosti na druhu zařízení, na němž je WWW dokument prezentován (obrazovka počítače, TV, tiskárna, kapesní počítač, mobilní telefon, slepecký výstup, zvukový výstup atd.). Dále umožňují definovat vlastnosti pro třídy a identifikátory - objekty v HTML dokumentu, jejichž třída nebo identifikátor odpovídají těm z definic v CSS, potom tyto vlastnosti přebírají.[6.6.4]

Způsoby použití:

Definice CSS mohou být součástí HTML dokumentu nebo mohou být v samostatném souboru. K HTML dokumentu jsou potom připojeny během jeho načítání do prohlížeče. Možnosti definování stylů jsou:

1. Přimo v HTML dokumentu, v části označené párovou značkou <STYLE> v sekci <HEAD>:

```
<HEAD> ... <STYLE type="text/css">seznam pravidel</STYLE> ... </HEAD>
```

2. V konkrétní značce (tagu) v HTML dokumentu:

```
<XXX ... style="seznam vlastností">
```

3. V externím souboru; soubor se načítá pomocí <LINK> v sekci <HEAD>:

```
<HEAD> ... <LINK rel="stylesheet" type="text/css" href="cesta_k_souboru"> ... </HEAD>
```

4. V externím souboru; soubor se načítá CSS direktivou @import v sekci <STYLE> (pouze v CSS2):

```
<STYLE type="text/css">@import url("cesta_k_souboru"); ... </STYLE>
```

V externím souboru je přítom seznam pravidel, stejně jako v sekci <STYLE>.

2.3 Dědičnost v CSS

Některé vlastnosti se mohou v rámci stromu dokumentu dědit. To znamená, že pokud není řečeno jinak, vlastnost definovanou pro jeden prvek přebírají i všechny objekty uvnitř tohoto prvku. Nedědí se ale všechny vlastnosti - každá vlastnost má definováno, zda její hodnota je či není dědičná [6].

Příklad:

Vlastnost color (barva textu) je dědičná. Pokud v tomto příkladu:

```
<H1>Toto je <EM>důležitý</EM> titulek</H1>
```

nebude mít prvek EM nijak specifikovanou vlastnost color, převezme ji od rodičovského prvku H1 (z hlediska stromu dokumentu je zde EM potomkem H1, neboť prvek EM je obsažen uvnitř prvku H1) - celý text titulku bude tedy zobrazen stejnou barvou textu.

Kaskáda:

Styly, ovlivňující jeden dokument, mohou pocházet ze tří různých zdrojů: od autora, od uživatele a z klientu. Autor (tvůrce dokumentu) specifikuje styly přímo v dokumentu (či v externích souborech k němu připojených - viz dříve). Uživatel si v některých prohlížečích může zvolit své vlastní styly (např. připojit svůj vlastní soubor se styly nebo nastavit některé vlastnosti přímo v prohlížeči). A v neposlední řadě každý klient (prohlížeč) má za povinnost mít definovanou *výchozí tabulku stylů*, která předchází všem dříve jmenovaným a definuje tak výchozí hodnoty podle možností dané klientem.

Všechny tyto styly se navzájem ovlivňují a tvoří výsledné hodnoty podle pravidel kaskádování. Kaskáda ohodnotí každý styl - přidělí mu tzv. *váhu*. Pokud je pro danou vlastnost více pravidel, přednost dostane to, které má větší váhu. Standardně mají styly autora větší váhu než styly uživatele; pouze při použití direktivy !important (viz dále) je tomu naopak. Styly autora i uživatele ale mají vždy větší váhu než výchozí styly klientu. Váha také záleží na pořadí a způsobu načtení: styly z dané tabulky stylů mají větší váhu a přepíší styly importované z jiné tabulky. Aby se mohla určit hodnota dané vlastnosti daného prvku, klient musí všechna pravidla nejprve setřídít podle následujícího třídění kaskád [6.1]:

1. Nejprve najde všechna pravidla, která se vztahují k danému prvku, dané vlastnosti a použitému médiu (výstupní zařízení). Pravidla se použijí, pokud jejich selektor páruje s daným prvkem.
2. Poté setřídí pravidla podle jejich váhy a zdroje - styly autora mají přednost před styly uživatele, výchozí styly klientu mají nejmenší váhu atd.
3. Následně setřídí pravidla podle jejich specifičnosti - specifičtější pravidla mají přednost před obecnějšími. Se pseudo-třídami a pseudo-prvky se zde nakládá, jako by to byly běžné třídy a prvky.
4. V posledním kroku jsou pravidla setříděna podle pořadí v dokumentu. Z pravidel, která mají stejnou váhu, zdroj i specifičnost, se použije to, které je definováno nejpozději. Importované styly se považují za definované dříve než styly popsané přímo v dané tabulce.

2.4 PHP skripty a jejich historie

Jedná se o interpretovaný skriptovací jazyk (jakýsi hybrid jazyka C), tzn. že pro spuštění skriptů (zdrojových textů) je nutný externí program (tzv. interpreter), který je vykoná. Samotné skripty jsou vepsány v HTML souboru, kde pak interpreter PHP na místo původního skriptu zapíše jeho výsledek, tzn. že klientovi se dostane vždy čistý HTML kód, částečně či dokonce celý vyprodukovaný skriptem[8].

Je to nástroj poměrně mladý. Současné proslulosti se mu dostalo až ve verzi 3.0, která byla uvolněna teprve v polovině roku 1998 (systém vznikl již v roce 1994, avšak jeho tehdejší podoba se od té současné značně liší).

Jeho autorem a jedním ze současných vývojářů je Rasmus Lerdorf a nyní je PHP dostupný pro platformy Linux, 32bitová Windows (WIN '95/'98, NT) a OS Macintoshe. Nutno podotknout, že všechny verze PHP jsou uvolněny **ZCELA ZDARMA**, a za zmínku stojí i to, že v PHP běží i populární emailový server Email.cz, www.centrum.cz a další.

Nejnovější počín od vývojářů je současný release PHP v4 - Candidate 1 (tedy 1. kandidát na ostrou verzi PHP v4), které se dostalo značného vylepšení, především na platformě 32 bitových Windows, kde je PHP dostupný jako modul (dynamicky linkovaná knihovna - soubor .dll) pro rozšíření ISAPI (Microsoft Internet Server API - spolupráce web-server<->aplikace), což značně urychluje jeho činnost, neboť se interpreter nemusí při každém volání znovu zavádět do paměti, nýbrž je v ní trvale zavaden.

PHP kromě těchto variant existuje ještě jako modul pro Linuxovský web-server **Apache**, kde je podobně jako u ISAPI interpreter permanentně zaveden v paměti. To je ale samozřejmě vázáno s nutností použít právě tohoto web-serveru, což degraduje celou myšlenku PHP - a sice kompatibilitu s jakýmkoliv web-serverem. Je to případ právě konkurenčního skriptovacího jazyka ASP, jenž je vázán právě pouze na web-server firmy Microsoft, která se samozřejmě s podporou pro jiné OS než Windows nijak nezabývá. Bohužel výrazné změny na platformě 32bitových Windows s sebou nenesou jen klady - příkladem toho je, že zde nebudou již použitelné knihovny DLL, používané v PHP3[8.3].

Je možné si jej stáhnout adresy www.php.net nebo ještě lépe z českého portálu www.php.cz. Vyberete si pouze požadovanou platformu a už můžete stahovat - instalace se provede prostým rozbalením archivu do určeného adresáře na disku (myšlena klasická distribuce interpreteru - nikoli modul pro Apache či ISAPI, tam je to drobet o něčem jiném) a zkopírováním souboru php3.ini do adresáře s Windows (obvykle C:\Windows\).

Jak jsem již zmínil, PHP je skriptovací jazyk, který se vykonává na serveru, pro naši další práci je tedy nutný přístup k nějakému webserveru. Jistě by nebylo moudré připojovat se při každém našem požadavku na vyzkoušení skriptu na nějaký vzdálený web server, tam soubor uploadovat a pak teprve browserem downloadovat výsledné HTML.

Proto je více než záhodno nainstalovat si web-server na svém vlastním počítači, už jen proto, že tak budou moci PHP používat i ti, kdož nejsou připojeni k Internetu či dokonce na svém

počítači, ani nemají instalovanu žádnou síťovou kartu či modem. Ti využijí možnosti tzv. loopbacku, což je přesměrování na váš lokální počítač (IP adresa 127.0.0.* se považuje jako localhost - tedy adresa lokálního počítače).

V tomto případě je ve Windows nutné nainstalovat ovladač telefonního adaptéru (dial-up adapter) (i přesto, nejste-li jeho majiteli) a poté samozřejmě TCP/IP protokolu (jako IP adresu pak můžete nastavit libovolnou). Poté se již můžete poohlédnout po nějakém web-serveru.

K dostání je dnes opravdu velké množství kvalitních web-serverů, takže není žádný problém vybrat si podle svého gusta. Pro operační systém Linux je volba myslím jednoznačná - Apache je jistě tou nejlepší volbou.

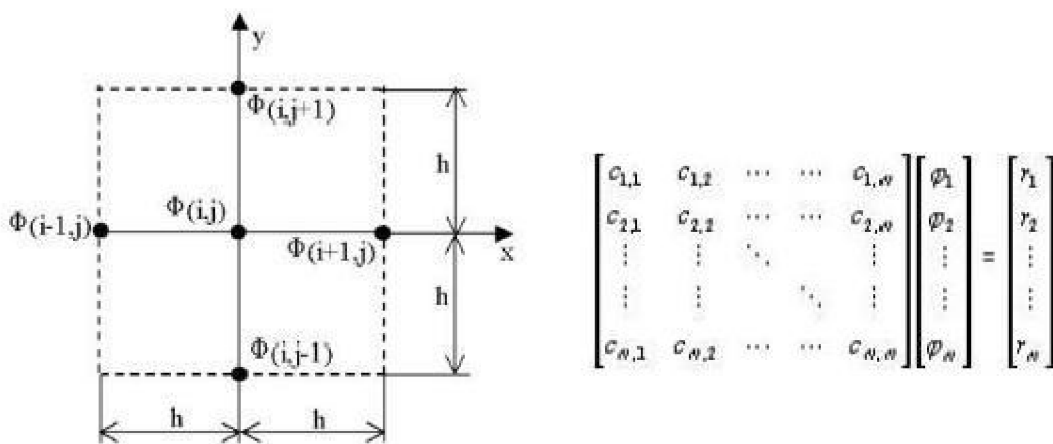
3 Popis metody

Metoda konečných diferencí v časové oblasti (*Finite-Difference Time-Domain, FDTD*) je založena na tom, že původně spojitá hledaná funkce je nahrazena sadou diskrétních funkčních hodnot. Nehledáme již tedy průběh funkce, ale jen sadu jejích hodnot v uzlových bodech diskretizační sítě [3.2.1].

Za podmíněk, že bychom znali současný stav libovolné soustavy a také mechanismy, kterými se řídí, mohli bychom dopředu určit situace v této soustavě v budoucnosti. Ovšem v celém vesmíru je přípiš mnoho zákonů a jeho rozlehlost je tak přípiš obrovská, že jeho komplexní popis je naprosto nemožný. Ovšem v případě elektromagnetického pole to dokážeme. Výpočet obvykle požadujeme v omezeném prostoru, který není pojen s okolím. Veličiny máme většinou dvě vektor E a vektor H .

Postup řešení:

1. ...výběr vhodné množiny uzlů (výběr sítě)
2. ...volba vzdálenosti mezi uzly
3. ...aproximace diferenciálního operátoru diferenčním
4. ...sestavení soustavy rovnic (okrajové podmínky)
5. ...řešení soustavy rovnic (Gaussova eliminace, vlastní čísla-vektory, iterační metody...)



Obr. 3.1 Sestavení soustavy rovnic do matic a přidělení souřadnic uzlům.

Elektromagnetické pole můžeme popsat Maxwellovými rovnicemi [3.2.9], které vyjadřují souvislost časové změny elektromagnetického pole a jeho prostorového rozložení. Při výpočtu pole bereme v úvahu hodnoty pole v konečném počtu bodů a časových okamžiků, a navíc jen s konečnou přesností.

Při řešení problému většinou můžeme vyjít z podmínky nulových hodnot veličin pole v počátečním čase. Pro nás to znamená, že známe pole v celém vyšetřovacím objemu. Všechny tyto faktory hovoří pro možnost výpočtu elektromagnetických polí v časové oblasti.

Nesmíme ovšem zapomenout, že při realizaci výpočtu pole v časové oblasti můžeme vždy brát v úvahu hodnoty pole v konečném počtu bodů a časů. Výsledkem je, že námi vypočtená pole budou od těch skutečných mírně odlišná. Tento nedostatek se odrazí v chybě numerického řešení [3.2.20].

Při metodě konečných diferencí v časové oblasti vycházíme z Maxwellových rovnic v diferenciálním tvaru. Derivace podle času i prostorových souřadnic nahrazujeme při výpočtech metodou FDTD diferencemi. Derivace lze nahradit diferencemi tří typů [2.1, 2.2, 2.3]:

- dopřednými

$$\frac{\partial F}{\partial x} \approx \frac{F(x + \Delta) - F(x)}{\Delta} \quad (3.1)$$

- zpětnými

$$\frac{\partial F}{\partial x} \approx \frac{F(x) - F(x - \Delta)}{\Delta} \quad (3.2)$$

- centrálními

$$\frac{\partial F}{\partial x} \approx \frac{F(x + \Delta / 2) - F(x - \Delta / 2)}{\Delta} \quad (3.3)$$

Ve výše uvedených vztazích je $F(x)$ je diferencovaná funkce a Δ je malý posuv, který slouží k odhadu derivace. Čím je Δ menší, tím více se difference blíží hodnotě derivace.

V praxi se nejčastěji používají centrální difference. Vhodným rozmístěním prostorových uzlů pro tři složky intenzity elektrického pole E_x , E_y a E_z a pro tři složky intenzity pole magnetického H_x , H_y a H_z můžeme velmi efektivně modelovat operátory rotace těchto veličin v první a druhé Maxwellovy rovnici. Nejčastěji se používá prostorové rozmístění uzlů, které koncem 60. let navrhl profesor Yee (obr. 2.1), viz [3.2.3].

Rovněž derivace podle času, které se nacházejí na pravých stranách první a druhé Maxwellovy rovnice, jsou nahrazeny centrálními diferencemi. Vztah mezi časovým krokem a prostorovým krokem, který zajišťuje bezpodmínečnou stabilitu FDTD, je nazýván Courantovu podmínkou.

V celé řadě případů je výpočet prováděn mimo vodiče, v prostředí bez elektrických i magnetických vlastností ($\mu = \mu_0$, $\varepsilon = \varepsilon_0$). Počet numerických operací lze pak snížit zavedením měřítka pro jednu z intenzit pole. Tím lze ušetřit tři násobení při přepočtu každé buňky. Zavedeme-li například měřítka pro intenzitu elektrického pole viz. [2.2.26]

$$E' = \frac{\Delta t}{\mu \Delta x} E \quad (3.4)$$

pro stejný diskretizační krok ve všech prostorových souřadnicích $\Delta x = \Delta y = \Delta z$ získáme upravené Maxwellovy rovnice, pro které platí: viz. [3.2.33]

$$C \frac{(\Delta t)^2}{\mu \varepsilon (\Delta x)} \quad (3.5)$$

Význam symbolů ve výše uvedených vztazích:

$F(x)$ – diferencovaná funkce

Δ – posuv

E_x, E_y a E_z – složky intenzity el. pole

H_x, H_y a H_z – složky intenzity mag. pole

μ – permitivita

t – čas

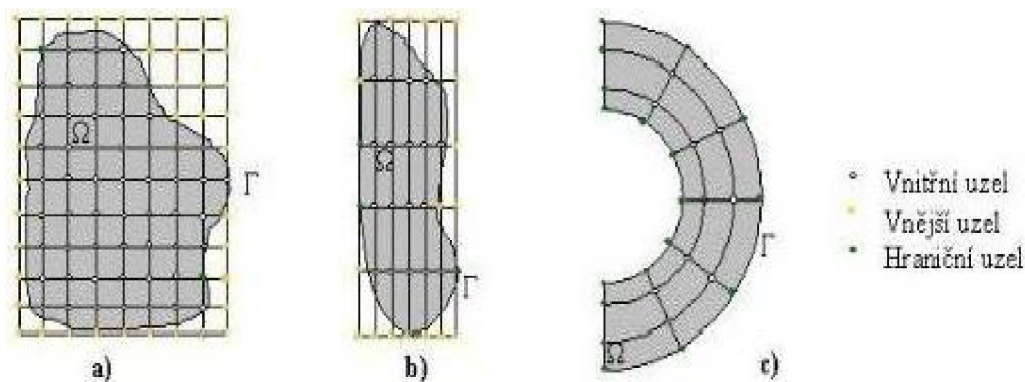
Výhody:

1. ...jednoduchost při programování a numerické realizaci
2. ...relativní jednoduchost v nelineárních matematických modelech

Nevýhody:

1. ...problém s aproximací okrajových podmínek na jednotlivých částech hranic, které nejsou vhodně použitelné na rozdílně husté síť
2. ...zhoršení přesnosti aproximovaného řešení pro síť s různým odstupem uzlů
3. ...nezbytnost relativně hustého časového kroku

Derivace podle času i prostorových souřadnic nahrazujeme při výpočtech metodou FDTD výše zmíněnými diferencemi, které nahrazují derivace. Pro náhradu prostorových derivací můžeme najít takové uspořádání diskretizačních uzlů pro jednotlivé složky pole, aby potřebné derivace byly vždy počítány z diferencí centrálních které aproximují hodnotu derivace nejvěrneji.



Obr. 3.2 Výběr vhodné množiny uzlů.

Pro náhradu derivací v čase se nepoužívají zpětné diference, protože při jejich použití dochází k divergenci výpočtu pro libovolnou velikost časového kroku. Dopředné diference nám umožňují najít početní postupy bezpodmínečně stabilní pro volitelnou délku kroku. Ovšem stabilita je zde podmíněna nutností inverze matice v každém kroku výpočtu. To činí tuto metodu poměrně náročnou na výpočet. Pokud je námi zvolený systém stabilní, nemusí to zaručovat přesnou hodnotu výpočtů. V dnešní době se nejčastěji používají pro metodu diskretizace v čase centrální diference, které nahrazují časové derivace [3.2.2].

Diskretizace při vhodném rozdělení složek pole, přičemž některé jsou diskretizovány v časech $n\Delta\tau$ a jiné v časech $(1/2 + n)\Delta\tau$, je stabilní, pokud časový krok $\Delta\tau$ nepřekročí jistotu mez, tzv. *Courantovu podmínku*. Tím, že jsme diskretizovali původně spojité pole a nahradili jsme derivace diferencemi, jsme vytvořili jisté chyby. Ovšem pokud známe vztah mezi velikostí diskretizačního kroku a chybou výpočtu, můžeme zpětně stanovit hustoty diskretizační sítě pro získání potřebné přesnosti. *Kritická mez (Courantova podmínka)*[3.2.2.4]:

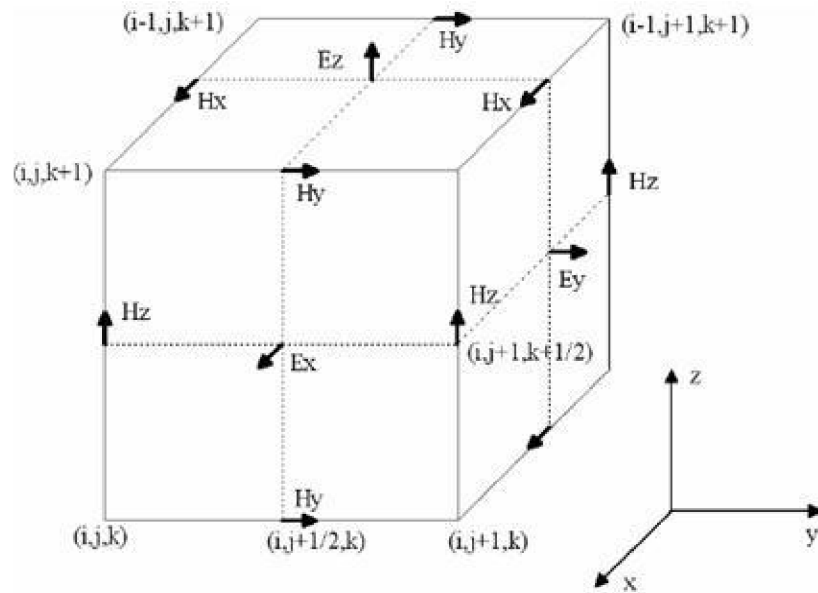
$$\Delta t \leq \frac{1}{c} \cdot \left(\frac{1}{(\Delta x)^2} + \frac{1}{(\Delta y)^2} + \frac{1}{(\Delta z)^2} \right)^{-1/2} \quad (3.6)$$

Proto postupujeme jakoby od konce, nejprve určíme počet diskretizačních řetězců v prostoru a potom úměrně zvolené diskretizaci určíme velikost časového kroku podle (3.6). Diskretizační řetězce nastavujeme tak, aby na jednu délku vlny v opticky nejhustším prostředí připadalo alespon 10 diskretizačních řetězců. Vlnová délka je tu chápána jako délka vlny o největší frekvenci, na níž chceme hodnotit vlastnosti obvodu.

V kartézské souřadné soustavě byla metoda metoda FDTD použita poprvé ve frekvenční oblasti. Elektromagnetické pole má šest složek, E_x, E_y, E_z, H_x, H_y a H_z . Odvození pro složku E_x :

$$E_{x,i,j,k}^{n+1} = \left(\frac{1 - \frac{\sigma\Delta t}{2\varepsilon}}{1 + \frac{\sigma\Delta t}{2\varepsilon}} \right) \cdot E_{x,i,j,k}^n + \left(\frac{\Delta t}{\varepsilon} \right) \cdot \left(\frac{H_{z,i,j+1/2,k}^{n+1/2} - H_{z,i,j-1/2,k}^{n+1/2}}{\Delta y} - \frac{H_{y,i,j,k+1/2}^{n+1/2} - H_{y,i,j,k-1/2}^{n+1/2}}{\Delta z} \right) \quad (3.7)$$

$E_{x,i,j,k}$ je zde myšlená hodnota složky intenzity elektrického pole ve směru x v uzlu sítě s inexty i, j, k . Horní index se používá pro vyznačení času E na n značí hodnotu veličiny E v n -tém časovém kroku, přesněji v čase $t=n\Delta\tau$. Musíme si ovšem uvědomit, že posun diskretizace v čase vede na neceločíselné hodnoty indexu n a posun jednotlivých veličin v prostoru. Také posun jednotlivých veličin v prostoru vede k neceločíselným hodnotám indexů i, j a k .



Obr. 3.3 Buňka prostorové diskretizační sítě FDTD v kartézské souřadné soustavě.

Pro zjednodušení můžeme použít měřítko pro intenzitu elektrického pole:

$$E' = \frac{\Delta t}{\mu \Delta x} \quad (3.8)$$

Pokud budeme brát stejné diskretizační kroky ve všech třech prostorových souřadnicích, můžeme upravit takto a dojde k velikému zjednodušení výpočtu (Pro intenzitu H_x):

$$H_{x,i,j,k}^{n+1/2} = H_{x,i,j,k}^{n-1/2} + E'_{y,i,j,k+1/2}{}^n - E'_{y,i,j,k-1/2}{}^n + H_{x,i,j,k}^n - E'_{y,i,j+1/2,k}{}^n \quad (3.9)$$

Soustavy rovnic (3.7) umožňují počítat časový vývoj elektromagnetického pole ze známého počátečního stavu. Počáteční podmínkou je ve většině řešených úloh stanovena nulová hodnota vektorů pole v celé vyšetřované oblasti. Musíme ovšem doplnit buzení tak, že vnutíme do některých uzlů sítě v určitém čase nenulové hodnoty pole. Můžeme použít například harmonický budící signál, který se budou numerickou sítí šířit postupně a po určitém čase se ustálí. V případech lineárních soustav se však častěji používá postup takový, že získáme pomocí FDTD časovou odezvu lineární soustavy na vhodný vstupní signál a pak transformujeme pomocí *Fourierovy transformace* do frekvenční oblasti [3.2.2].

Nejčastěji se provádí buzení Gaussovým impulzem:

$$S(t_n) = \exp\left(\frac{t_n - t_c}{N}\right)^2 \quad (3.10)$$

nebo Blackmannův-Harrisův v časovém okně:

$$S(t_n) = 0,35875 + 0,48829 \cos\left(\frac{\pi(t_n - t_c)}{N}\right) + 0,14128 \cos\left(\frac{2\pi(t_n - t_c)}{N}\right) + 0,01168 \cos\left(\frac{3\pi(t_n - t_c)}{N}\right) \quad (3.11)$$

Pro přehlednost, to je doba příslušící n -tému časovému kroku, t_c je poloviční šířka impulzu a N je počet časových kroků. Tento signál vnutíme do příslušných uzlů sítě. Musíme ovšem zajistit, aby signál trval nejvýše tak dlouho, než k uzlům dorazí vlna odražená od obvodu. Jinak by došlo k odrazu zpět a tím ke zkreslení. Také nesmíme zapomenout aplikaci vhodného rozložení pole vlny ve vedení, kterým celou soustavu napájíme. Pro dominantní vid TEM na vstupní a výstupní bráně, který má radiální směr použijeme Gaussův šum a vnucujeme hodnoty a za výstupy vln na jednotlivých branách bereme :

$$E_r(r, t) = \frac{1}{2} \exp\left(\frac{t_n - t_c}{N}\right)^2 \quad a \quad S(t_n) = \sum_i E_r(r_i) \cdot \frac{1}{r_i} \quad (3.12)$$

Tento postup je hojně využíván, pokud řešíme vlnovody s kovovým pláštěm, protože je velmi jednoduché toto rozložení odvodit. Na závěr se ještě podíváme na okrajové podmínky. Pokud chceme najít řešení pro určité elektromagnetické pole potřebujeme Maxwellovy rovnice v diferenciálním tvaru doplnit o okrajové podmínky. Hodnota jedné veličiny je počítána ze dvou sousedních hodnot jiné veličiny. Uzly na okraji struktury takové sousedící body nemají, a proto musíme zvolit jiný postup. Můžeme zavést elektrické a magnetické stěny, na kterých budeme chtít nulovou hodnotu tečné složky příslušné intenzity pole popřípadě nulovou hodnotu derivace podle normály. Následně už jenom chceme nulovou hodnotu tečné složky, nebo nulovou derivaci[3.2.2.4].

Další možnost je poněkud obtížnější a cílem je zavést absorpční podmínky, které zaručí bezodrazovost zakončení sítě. Pokud je nám znám směr vlny dopadající na stěnu sítě, můžeme do hraničních buněk dosadit hodnoty ze sousedních vnitřních buněk, které vystihneme v předešlém okamžiku, známe samozřejmě rychlost s jakou se šíří vlna ven z obvodu.

$$M = \frac{\Delta x}{c \Delta t} \quad (3.13)$$

Ve většině úloh ovšem neznáme předem směr, ze kterého bude vlna na rozhraní dopadat. Postupem času se vyvinuly další metody. *Engquist* a *Majda* zkoumali tzv. *jendosměrné vlnové rovnice*:

$$\left(k_x - \sqrt{\frac{\omega^2}{c^2} - k_y^2 - k_z^2}\right) \cdot \left(k_x + \sqrt{\frac{\omega^2}{c^2} - k_y^2 - k_z^2}\right) \cdot U = 0 \quad (3.14)$$

Nebo také pomocí dvojčleného Taylorova rozvoje operátoru můžeme použít jednosměrnou rovnici umožňující šíření vln pouze jedním směrem k okraji sítě[3.2.2.4].

$$L U = \frac{\delta^2 U}{\delta x \delta t} - \frac{1}{c} \cdot \frac{\delta^2 U}{\delta t^2} + \frac{c}{2} \cdot \frac{\delta^2 U}{\delta y^2} + \frac{c}{2} \cdot \frac{\delta^2 U}{\delta z^2} = 0 \quad (3.15)$$

Vyšší tlumení odražené vlny můžeme dosáhnout pomocí tzv. *dokonale přizpůsobené vrstvy (PML)*. Dopadá-li elektromagnetická vlna na rozhraní dvou prostředí, dojde většinou k odrazu, pokud nejsou vlnové impedance obou prostředí shodné.

Vlna většinou dopadá na hranici zkoumané oblasti z prostředí s malými nebo nulovými ztrátami (prostředí s reálnou impedancí). Pokud chceme dosáhnout tlumení vlny, musí být toto prostředí ztrátové. Tento problém lze vyřešit, pokud prostředí má nejen komplexní složku ale i permeabilitu. Tímto způsobem se sestavují pohlcující materiály na kovové konstrukce. Dochází tedy ke vzniku tepla při prepolarizování jakékoliv z intenzit pole. Impedanci prostředí získáme ze vzorce:

$$Z_v = \sqrt{\frac{j\omega\mu + \sigma'}{j\omega\varepsilon + \sigma}} \quad (3.16)$$

4 Popis vrstev

4.1 Vrstva A

Vrstva A je souvislým textem, obsahujícím zejména slovní popis probíraného jevu. Čtenář zde není zatěžován složitými matematickými vztahy a náročnými odvozeními. Cílem je co nejjednodušeji dobrat se podstaty. Tato vrstva je určena především pro studenty bakalářského studijního programu. Pokud Vás při četbě určitá pasáž zaujme a budete ji chtít prostudovat hlouběji, můžete přejít pomocí menu do druhé vrstvy učebnice (vrstva B).

The screenshot shows a web browser displaying a page from a digital textbook. The page title is "2.1 Metoda konečných diferencí" (2.1 Finite-Difference Method). The left sidebar contains a navigation menu with items like "Úvod", "2.1 Metoda konečných diferencí", and "2.1A Základní teorie". The main content area is titled "Základní teorie" and discusses the Finite-Difference Time-Domain (FDTD) method. It explains that the method approximates continuous fields by discrete values on a grid. A diagram (Obr. 2.1A.1) shows a 2D grid with nodes labeled $\Phi_{(i,j)}$, $\Phi_{(i,j-1)}$, $\Phi_{(i,j+1)}$, and $\Phi_{(i-1,j)}$, with grid spacing h . Below the diagram is a system of linear equations (Obr. 2.1A.2) representing the discretized equations:
$$\begin{bmatrix} c_{1,1} & c_{1,2} & \dots & c_{1,n} \\ c_{2,1} & c_{2,2} & \dots & c_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ c_{n,1} & c_{n,2} & \dots & c_{n,n} \end{bmatrix} \begin{bmatrix} \Phi_1 \\ \Phi_2 \\ \vdots \\ \Phi_n \end{bmatrix} = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_n \end{bmatrix}$$
 The page also includes sections for "Postup řešení" (Solution procedure), "Výhody" (Advantages), and "Nevýhody" (Disadvantages). At the bottom, there are three equations for numerical derivatives:
$$\frac{\partial F}{\partial x} \approx \frac{F(x+\Delta) - F(x)}{\Delta} \quad (2.1A.1)$$

$$\frac{\partial F}{\partial x} \approx \frac{F(x) - F(x-\Delta)}{\Delta} \quad (2.1A.2)$$

$$\frac{\partial F}{\partial x} \approx \frac{F(x+\Delta/2) - F(x-\Delta/2)}{\Delta} \quad (2.1A.3)$$

Obr. 4.1 Ukázková stránka vrstvy A.

4.2 Vrstva B

Vrstva B nabízí navíc veškerá matematická odvození a jejich podrobnou diskusi. Rovněž druhá vrstva má charakter souvislého textu, avšak textu náročného, psaného pro zasvěceného čtenáře na magisterské úrovni.

[GZREN](#) [Přechod](#) [Index](#) [Návod](#) [PDF verze](#)

Multimediální učebnice

1. Úvod

2. Šíření EM vln

2.1 Metoda konečných diferencí

2.1A Základní teorie

2.1B Podrobnější popis

2.1C Program v Matlabu

2.1D Vývoj programu

2.1E Java aplik

2.1F Kvíz

2.1 Metoda konečných diferencí

Podrobnější popis

Pro náhradu derivací v čase se nepoužívají zpětné diference, protože při jejich použití dochází k divergenci výpočtu pro libovolnou velikost časového kroku. Dopředné diference nám umožňují najít početní postupy bezpodmínečně stabilní pro volitelnou délku kroku. Ovšem stabilita je zde podmíněna nutností inverze matice v každém kroku výpočtu. To činí tuto metodu poměrně náročnou na výpočet. Pokud je nám zvolený systém stabilní, nemusí to zaručovat přesnou hodnotu výpočtu. V dnešní době se nejčastěji používají pro metodu diskretizace v čase centrální diference, které nahrazují časové derivace.

Diskretizace při vhodném rozdělení složek pole, přičemž některé jsou diskretizovány v časech n delta t a jiné v časech $(1/2 + n)$ delta t , je stabilní, pokud časový krok delta t nepřekročí jistou mez, tzv. Courantovu podmínku. Tím, že jsme diskretizovali původně spojité pole a nahradili jsme derivace diferencemi, jsme vytvořili jisté chyby. Ovšem pokud známe vztah mezi velikostí diskretizačního kroku a chybou výpočtu, můžeme zpětně stanovit hustotu diskretizační sítě pro získání potřebné přesnosti, kroměka meze (Courantova podmínka):

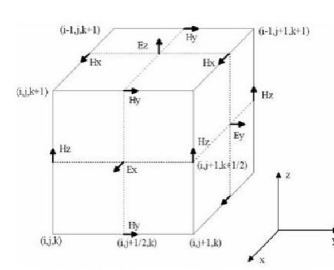
$$\Delta t \leq \frac{1}{c} \left(\frac{1}{(\Delta x)^2} + \frac{1}{(\Delta y)^2} + \frac{1}{(\Delta z)^2} \right)^{-1/2} \quad (2.18.1)$$

Proto postupujeme jakoby od konce, nejprve určíme počet diskretizačních řetězců v prostoru a potom úměrně zvolené diskretizaci určíme velikost časového kroku podle VZORCE Diskretizační řetězce nastavujeme tak, aby na jednu délku vlny v optiky nej hustším prostředí připadalo alespoň 10 diskretizačních řetězců. Vlnová délka je tu chápána jako délka vlny o největší frekvenci, na níž chceme hodnot vlastnosti obvodu.

V kartézské souřadné soustavě byla metoda metoda FDTD použita poprvé ve frekvenční oblasti, Elektromagnetické pole má šest složek, E_x , E_y , E_z , H_x , H_y a H_z . Odvození pro složku E_x :

$$E_{x,i,j,k}^{n+1} = \left(\frac{1 - \frac{\sigma \Delta t}{2\epsilon}}{1 + \frac{\sigma \Delta t}{2\epsilon}} \right) E_{x,i,j,k}^n + \left(\frac{\Delta t}{\epsilon} \right) \left(\frac{H_{z,i,j-1/2,k}^{n+1/2} - H_{z,i,j+1/2,k}^{n+1/2}}{\Delta y} - \frac{H_{y,i,j,k-1/2}^{n+1/2} - H_{y,i,j,k+1/2}^{n+1/2}}{\Delta z} \right) \quad (2.18.2)$$

$E_{x,i,j,k}$ je zde myšlená hodnota složky intenzity elektrického pole ve směru x v uzlu sítě s indexy i, j, k . Horní index se používá pro vyznačení času E na n -tém časovém kroku, přesněji v čase $t = n$ delta t . Musíme si ovšem uvědomit, že posun diskretizace v čase vede na necelocíslné hodnoty indexů n a posun jednotkových veličin v prostoru. Také posun jednotkových veličin v prostoru vede k necelocíslným hodnotám indexů i, j a k .



Obr. 2.1B.1. Buňka prostorové diskretizační sítě FDTD v kartézské souřadné soustavě.

Pro zjednodušení můžeme použít měřítka pro intenzitu elektrického pole,

$$E' = \frac{\Delta t}{\epsilon} \quad (2.18.3)$$

Obr. 4.2 Ukázková stránka vrstvy B.

4.3 Vrstva C

Vrstva C je uživatelskou příručkou skriptů Matlabu, které si může čtenář stáhnout, aby jejich prostřednictvím lépe pochopil studované jevy.

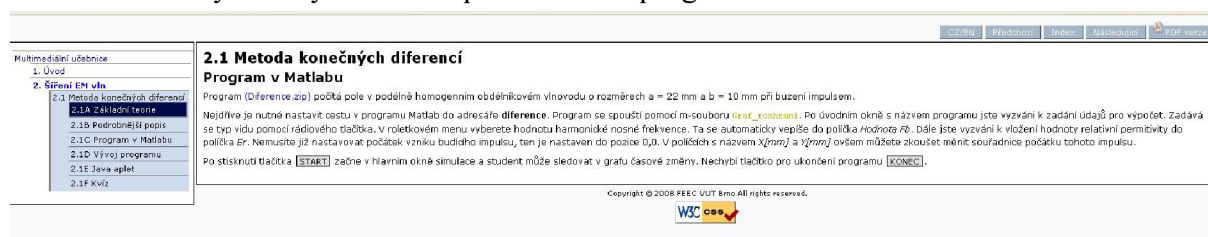
Program ([diference.zip](#)) počítá pole v podélně homogenním obdélníkovém vlnovodu o rozměrech $a = 22$ mm a $b = 10$ mm při buzení impulsem.

Nejdříve je nutné nastavit cestu v programu Matlab do adresáře **diference**. Program se spouští pomocí m-souboru [Graf_rozhrani](#).

Po úvodním okně s názvem programu jste vyzváni k zadání údajů pro výpočet. Zadává se typ vidu pomocí rádiového tlačítka. V roletkovém menu vyberete hodnotu harmonické nosné frekvence. Ta se automaticky vepíše do políčka [Hodnota Fb](#).

Dále jste vyzváni k vložení hodnoty relativní permitivity do políčka [Er](#). Nemusíte již nastavovat počátek vzniku budícího impulsu, ten je nastaven do pozice 0,0. V políčkách s názvem [X\[mm\]](#) a [Y\[mm\]](#) ovšem můžete zkusit měnit souřadnice počátku tohoto impulsu.

Po stisknutí tlačítka **START** začne v hlavním okně simulace a student může sledovat v grafu časové změny. Nechybí tlačítko pro ukončení programu **KONEC**.



Obr. 4.3 Ukázková stránka vrstvy C.

4.4 Vrstva D

Vrstva D je v podstatě programátorskou příručkou.

Rozměry vlnovodu, změny se v závislosti na mřížce.

$$a = 22e-3;$$
$$b = 10e-3;$$

Alokujeme pole proměnných o rozměrech mřížky. Promennými budou všechny prostorové složky intenzit.

$$Nx_i = \text{round}(Nx/2);$$
$$Ny_i = \text{round}(Ny/2);$$

$$Hz = \text{zeros}(Nx-1, Ny-1); \quad \% \text{ matice pro intenzity}$$
$$Ex = \text{zeros}(Nx, Ny-1);$$
$$Ey = \text{zeros}(Nx-1, Ny);$$

Práce s materiálovými konstantami

```
Ca = 1;
Cb = [ones( Nx-1, Ns)*dt/(eps*epr*dl), ones( Nx-1, Ny-Ns-1)*dt/(eps*dl)];
Da = 1;
Db = dt/(mi*dl);
```

Hlavní cyklus, spočívá ve snímání elektrických složek pole z lineárních souřadnic.

```
for t=1:round(steps/10)
    % cyklus
    if t<=imst
        Hz( Nxi-5, Nyi) = h(t);
    end;
```

Nakonec dopočítáme podélné složky

```
Ex(2:Nx-1,:) = Ca*Ex(2:Nx-1,:) +...
    Cb(2:Nx-1,:) .* Hz(2:Nx-1,:) - Cb(1:Nx-2,:) .* Hz(1:Nx-2,:);
Ey(:,2:Ny-1) = Ca*Ey(:,2:Ny-1) +...
    Cb(:,1:Ny-2) .* Hz(:,1:Ny-2) - Cb(:,2:Ny-1) .* Hz(:,2:Ny-1);

Hz = Da* Hz(1:Nx-1,1:Ny-1) + ... % dopocitani podelne slozky
    Db*(Ex(2:Nx,:) - Ex(1:Nx-1,:) + Ey(:,1:Ny-1) - Ey(:,2:Ny));
```

2.1 Metoda konečných diferencí
Vývoj programu
 Program počítá pole v podélné homogenním obdélníkovém vinovodu při buzení impulsem.
 Rozměry vinovodu, změni se v závislosti na mířce:

```
a = 2e-3;
b = 1e-3;
```

kde a je délka a b je šířka vinovodu.
 Alokujeme pole proměnných o rozměrech mřížky. Proměnnými budou všechny prostorové složky intenzity:

```
Nxi=round(Nx/2);
Nyi=round(Ny/2);
```

Maticy pro intenzity:

```
Hx = zeros( Nx-1, Ny-1);
Ex = zeros( Nx, Ny-1);
Ey = zeros( Nx-1, Ny);
```

Práce s materiálovými konstantami:

```
Ca = 1;
Cb = [ones( Nx-1, Ns)*dt/(eps*epr*dl), ones( Nx-1, Ny-Ns-1)*dt/(eps*dl)];
Da = 1;
Db = dt/(mi*dl);
```

Hlavní cyklus, spočívá ve snímání elektrických složek pole z lineárních souřadnic:

```
for t=1:round(steps/10)
    if t<=imst
        Hz( Nxi-5, Nyi) = h(t);
    end
end
```

Nakonec dopočítáme podélné složky Hz:

```
Ex(2:Nx-1,:) = Ca*Ex(2:Nx-1,:) + Cb(2:Nx-1,:) .* Hz(2:Nx-1,:) - Cb(1:Nx-2,:) .* Hz(1:Nx-2,:);
Ey(:,2:Ny-1) = Ca*Ey(:,2:Ny-1) + Cb(:,1:Ny-2) .* Hz(:,1:Ny-2) - Cb(:,2:Ny-1) .* Hz(:,2:Ny-1);
Hz = Da* Hz(1:Nx-1,1:Ny-1) + Db*(Ex(2:Nx,:) - Ex(1:Nx-1,:) + Ey(:,1:Ny-1) - Ey(:,2:Ny));
```

Pozn.: Volání programů:
[dici_TE.a](#) - [dici_TH.a](#)

Copyright © 2008 FEEC VUT Brno. All rights reserved.
 WSC CSS

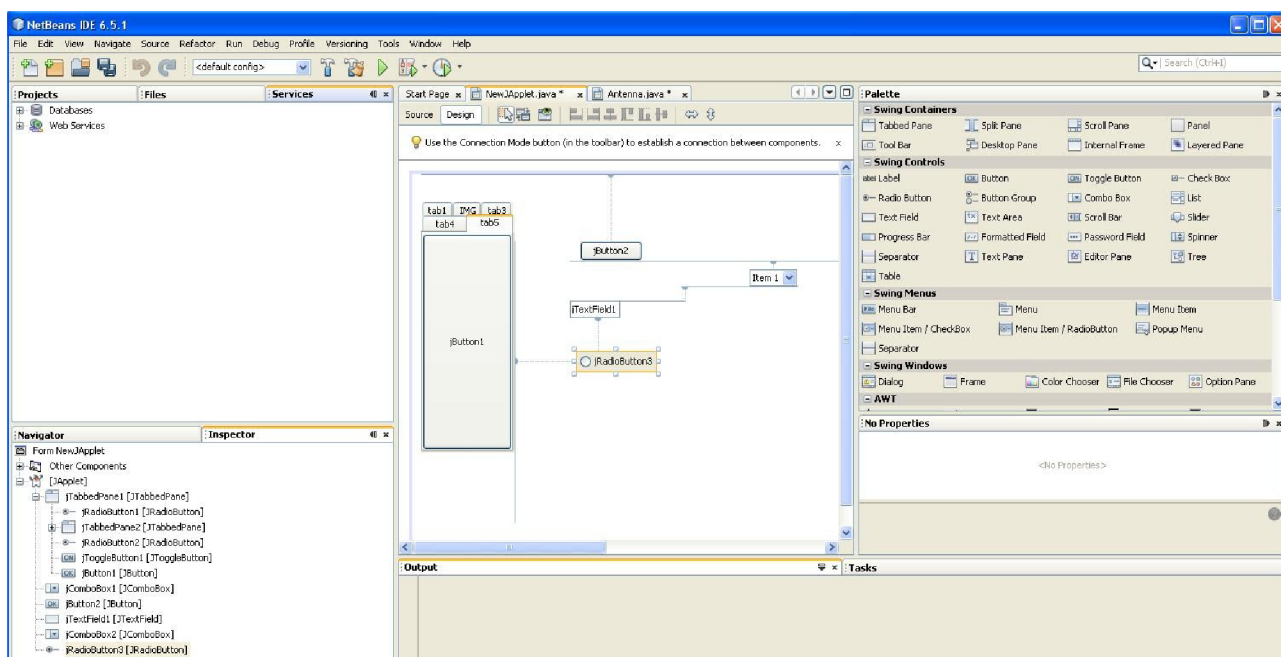
Obr. 4.4 Ukázková stránka vrstvy D.

4.5 Vrstva E

Vrstva E nabízí numerické modely studovaných jevů a struktur ve formě javovských appletů. Zdrojové kody jsou přiloženy, tato část není filálně dokončena.



Obr. 4.5 Ukázková stránka vrstvy E.



Obr. 4.6 Ukázková prostředí Netbeans.

4.6 Vrstva F

Umožňuje studentům ověřit si správné chápání studované látky. Jsou zde zveřejněny pětice otázek, čtenář u každé otázky vybírá správnou odpověď. Jakmile pětici otázek čtenář projde, je vyhodnocena jejich správnost. U nesprávných odpovědí a u nezodpovězených otázek je student informován o správném řešení. Na závěr kvízu je zobrazeno celkové hodnocení, které stejně jako pdf formát vygeneruju php skript umístěný na serveru.

2.1 Metoda konečných diferencí
Kvíz

Zodpovězením těchto otázek získáte zpětnou informaci o tom, jak jste dané problematice porozuměli. Pouze jediná z možností je správná. Nemusíte odpovídat na všechny otázky. Pokud neznáte u dané otázky odpověď, nechte ji nezodpovězenou (předem vykrááno "na tuto otázku neodpovím"). Váš výsledek pak **nebude** tržba otázkou ovlivněn. Správná odpověď **přidá 2 body** k celkovému hodnocení, při špatné odpovědi **ztrácíte 1 bod**. Otázky jsou děleny do skupin po pěti.
Po zodpovězení otázek klikněte na **Zobrazit výsledek**.
Zobrazuji otázky č. 1 až 5 z celkového počtu 5 otázek:

Otázka č.1
Metoda konečných diferencí v časové oblasti nahrazuje parciální derivace intenzit polí ...

Možné odpovědi pro otázku č.1:

- ... konečnými diferencemi (vzdálenost mezi uzly je konečná).
- ... konečnými diferencemi (diference vyjadřuje konečnou energii).
- ... vhodně zvolenou aproximační funkcí.
- Na tuto otázku neodpovím.

Otázka č.2
Courantova podmínka říká, jak velký musí být ...

Možné odpovědi pro otázku č.2:

- ... prostorový diskretní krok, aby chyba numerického řešení nepřekročila 1%.
- ... počet diskretních kroků na vinovou délku, aby metoda konvergovala.
- ... časový diskretní krok, aby metoda byla stabilní.
- Na tuto otázku neodpovím.

Otázka č.3
Pokud analyzovaná struktura obsahuje rozhraní dvou dielektrik ...

Zadání otázky číslo 3 ...

- ... buňka Yee automaticky splňuje podmínky na rozhraní.
- ... metodu konečných diferencí nemůžeme použít.
- ... metodu konečných diferencí je třeba vhodně modifikovat.
- Na tuto otázku neodpovím.

Otázka č.4
Absorpční okrajové podmínky a dokonale přizpůsobené vrstvy použijeme ...

Zadání otázky číslo 4 ...

- ... potřebujeme-li modelovat šíření vlny volným prostorem.
- ... pro modelování útlumových dělánek ve vlnovodech.
- ... pro modelování pronikání vlny do vodivého materiálu.
- Na tuto otázku neodpovím.

Otázka č.5
Při analýze vlnovodu metodou konečných diferencí v časové oblasti ...

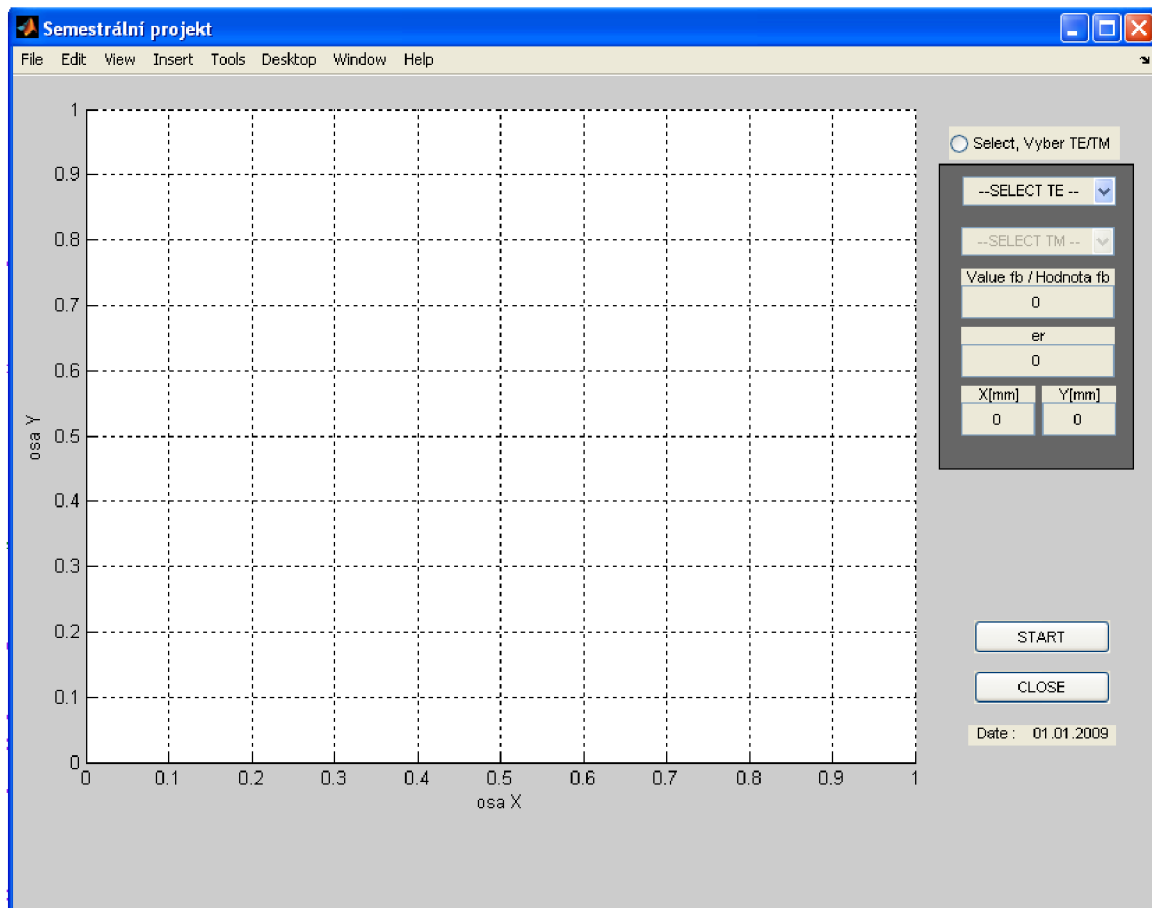
Zadání otázky číslo 5 ...

- ... nemůžeme studovat buňky, pokud existuje ustálená harmonická pole.

Obr. 4.7 Ukázková stránka vrstvy F.

5 Popis uživatelského rozhraní

Mým cílem bylo vytvořit uživatelské rozhraní jako výukovou aplikaci, která má za úkol, přiblížit studentům metodu konečných diferencí v časové oblasti. Z tohoto důvodu jsem se snažil o vytvoření rozhraní s jednoduchým ovládáním, které by v sobě zahrnovalo možnost eliminace určitých chyb, které by mohly nastat nesprávnou manipulací.



Obr. 5.1 Úvodní vzhled uživatelského rozhraní.

5.1 Stručný popis grafického rozhraní

Matlab nabízí několik možností tvorby grafických aplikací v závislosti na použité verzi programu. Moje aplikace byla vytvořena ve verzi 7.4.0 (R2007a). V většině případů nebývá problém s kompatibilitou s vyššími verzemi, avšak v nižších verzích mohou nastat určité problémy v zobrazování grafického rozhraní.

Toto prostředí je programováno systémem Handle Graphics, spočívající v ručním definování všech chování, stavů a pozic objektů. Systém HandleGraphics, ostatně asi jako každý jiný má určitou hierarchii, kde nejvýše postavená je obrazovka, pak formulář, a pod ním jsou objekty do něj vložené (axes, uicontrol, image...). Je tedy zřejmé, že musíme v tomto pořadí také programovat. Po definici funkce `function Graf_rozhrani(vstpar)` zjišťujeme rozlišení obrazovky a definujeme první objekt, kterým je samotný formulář (neboli figure).

```
F1=figure('Units','Pixels','Color',[1 0.518 0.035],'Position',...
    [0.05*Monitor(4) 0.05*Monitor(4) 820 600],...
    'Name','Semestrální práce BB2E','NumberTitle','off',...
    'Tag','Figure_1','Resize','on','Visible','on');
```

Každý objekt přiřazujeme jakoby nějaké proměnné, kterou můžeme pak vypsat (např. `get(F1)`). Jako první zde nastavujeme jednotky v pixelech, dále barvu RGB tvořenou třemi čísly od nuly do jedné, pozici x,y levého spodního rohu a rozměr, vypínáme automatický popisek, markujeme tag s hodnotou „Figure 1“, povolujeme měnit velikost a viditelnost. Tag je jakási vysačka, kterou si můžeme označit daný objekt. Je to velmi užitečná syntaxe a dále bude hojně využívána. V objektech obecně je možno nastavovat spousta dalšího, ale tyto příkazy si lze snadno už podle jejich názvu spojit s jejich významem, takže je tu nemá cenu vysvětlovat.

Neboť zde využíváme funkce, nabízí se programovat metodou Switched Board Programing, která využívá vynikající vlastnosti funkcí volat samy sebe s různými parametry. Dále jsou používány příkazy `switch` a `case`. Ukážeme si to na příkladu:

```
function Graf_rozhrani(vstpar) %vstupní parametry

if nargin==0
    :
h3=uicontrol('Units','Normalized','Position',[0.83 0.87 0.135 0.04],...)
    'Style','Popup','Tag','popup','Enable','off',...
    'String',' TE1| TE2| TE3| TE4| TE5| -SELECT-',...
    'Callback','Graf_rozhrani fb','Value',6);
:
else
    switch(vstpar)
        case('fb')

            TE=get(findobj('Tag','popup'),'Value')

            global fb;
            if TE==1
                fb=1.5e10;
                set(findobj('Tag','edit2'),'String','TE1 1.5e10 Hz');

            else
                if TE==2
                    fb = 3.0e10;
                    set(findobj('Tag','edit2'),'String','TE2 3.0e10 Hz');
                else
                    :

                    if TE==6
                        fb = 0;
                        set(findobj('Tag','edit2'),'String','0');
                    end
```


⋮
end

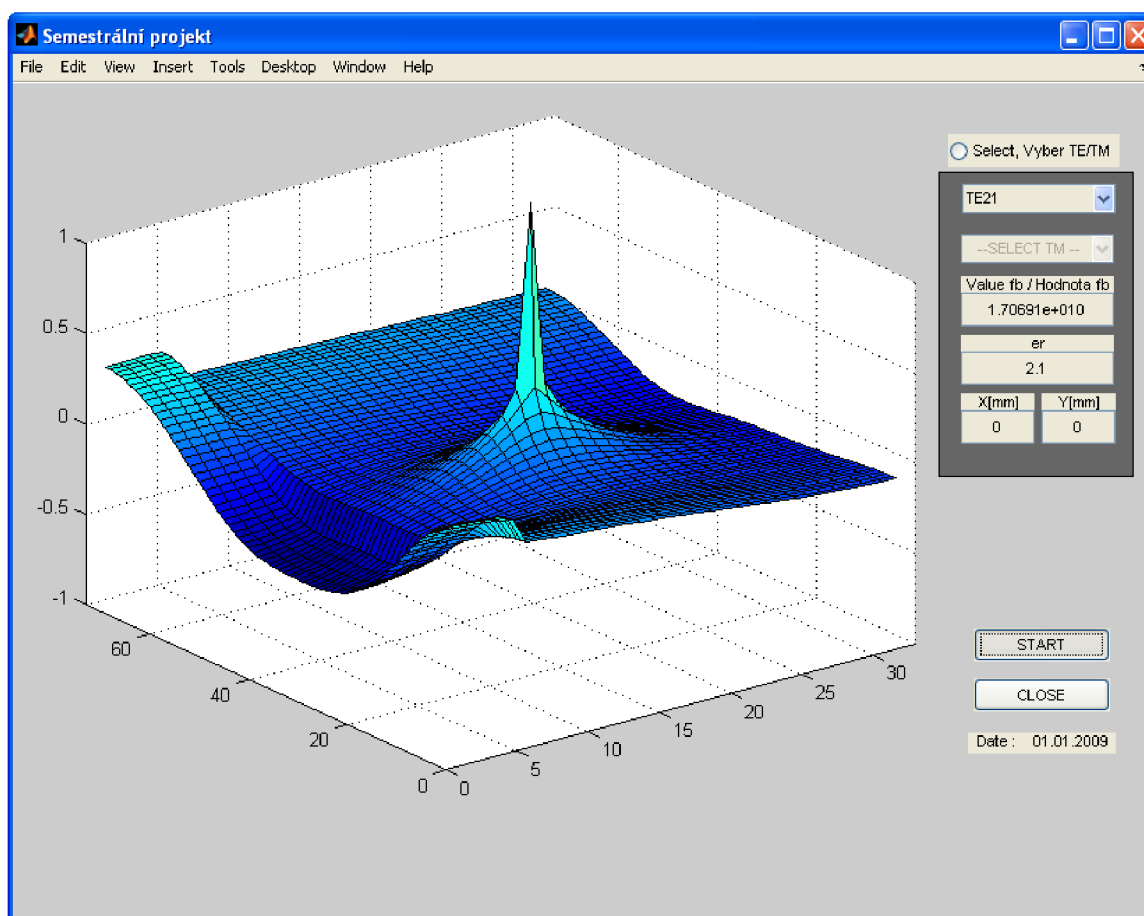
Po založení celé funkce následuje řádek `if nargin==0`, který říká: pokud funkce nemá žádné vstupní hodnoty, proved' následující instrukce. V objektu `h3` (pop-up menu), je pak krom jiných definic také `'CallBack'`, `'Graf_rozhrani fb'`, volající při jakékoli změně (výběru některé z položek) znovu tuto funkci, tentokrát ovšem se vstupní hodnotou! Neprojde přes podmínku `nargin` a pokračuje až za `else` příkazem `switch`, kde hledá `case` s příslušným parametrem `CallBacku` (`'fb'`). Zde pak můžeme pokračovat programováním a nastavováním vlastností. Tady se nám právě hodí `Tag`, díky němuž se můžeme odkazovat na jakékoli další objekty, získávat a dávat jim hodnoty.

Handle Graphic je grafický systém implementovaný v MATLABu. S jeho pomocí lze efektivně pracovat s grafickými objekty. Obsahuje příkazy pro dvoj- a trojrozměrnou vizualizaci dat, zpracování signálů a grafiku obecně.

Při tvorbě jsem využil výše zmíněnou metodu pro vyšší názornost a možnosti lepšího pochopení činnosti a tvorby objektu. Přímá realizace spouštění grafické aplikace je zabezpečena příkazem `Graf_rozhrani` v příkazovém okně Matlabu.

5.2 Editace parametrů

Uživatel má možnost přímého zadávání konkrétních parametrů pro tvorbu grafu.



Obr. 5.2 Vzhled spuštěné aplikace pro vid TE.

Zadané parametry jsou přímo zpracovány rovnicemi v m-souborech. Jedná se většinou o editační místa pro různé typy parametru. Také se zde nachází radiové tlačítko a dvě roletková menu. Kombinací těchto parametrů lze relativně rychle a názorně demonstrovat metodu konečných diferencí v časové oblasti.

1. Uživatel si vybere, zda chce vykreslit graf pro TE vid nebo TM vid v podélně homogenním obdélníkovém vlnovodu. Tím se zablokuje roletkové menu druhého vidu, aby uživatel nemohl udělat chybu a byl naveden ke správnému roletkovému menu.
2. Po zvolení typu vidu má uživatel na výběr z pěti nabízených hodnot vidu.
3. Zvolená hodnota vidu se automaticky zobrazí v políčku *Hodnota Fb*, což je hodnota harmonické nosné.
4. Uživatel nastaví podle libosti relativní permitivitu.
5. Poslední dvě políčka *X* a *Y* jsou souřadnice počátku budícího signálu.
6. Po nastavení parametrů uživatel stiskne tlačítko START a začne simulace. Výpočet není náročný a simulace probíhá okamžitě. Student může sledovat v grafu časové změny.
7. Tlačítko KONEC zavře celou aplikaci.

6 Závěr

V závěru své práce bych rád zhodnotil dosažené výsledky stanovených cílů, které jsem zmiňoval v úvodu.

Cílem mé bakalářské práce bylo vytvořit novou část internetové učebnice. Vytvořit vrstvy A, B, C, D, E a F a vytvořit uživatelské rozhraní v programu MATLAB a v také java applet. Všechny části měly být v českém a anglickém jazyce. Všechny vytvořené části musejí být kompatibilní se současnou verzí internetové učebnice.

Zdrojové kódy a vzhled mojí části učebnice je umístěn na:

<http://www.butchers.cz/>

Dle mého názoru se kompatibilita učebnice podařila a webové stránky je možno zaimplementovat do stávající verze učebnice. Vrstvy jsou schopny pracovat společně a fungují mezi nimi vztahy typu: další, předchozí, index a přepínání jazykové verze. Pdf formát tvoří generátor v php skriptu umístěném na serveru a nebyl součástí mé práce.

Uživatelské rozhraní v programu MATLAB je propracováno do drobných detailů a podmínek. Pomocí radio tlačítka a roletkových menu, nemůže uživatel při manipulaci s programem vytvořit kolizní stav programu.

Nepodařilo se mi ovšem dotáhnout do konce java applet, přiloženy jsou 2 varianty postupu řešení. První spočívá programování v prostředí Netbeans a druhé je založeno na komunikaci zdrojových kódů Matlabu a Javy.

7 Literatura

- [1] ZAPLATÍLEK, K., DONAR, B. *Matlab pro začátečníky*, 2.vydání. Praha: BEN – Nakladatelství technické literatury, 2005. 152 stran. ISBN 80-7300-175-6
- [2] ZAPLATÍLEK, K., DONAR, B. *Matlab tvorba uživatelských aplikací*, 1. vydání. Praha: BEN – Nakladatelství technické literatury, 2005. 216 stran. ISBN 80-7300-133-0
- [3] RAIDA, Z. et al. *Analýza mikrovlnných struktur v časové oblasti*. Brno: VUTIUM Publishing, 2003.
- [4] TAFLOVE, A. *Computational electrodynamics: The finite-difference time-domain method*. Boston: Artech House, 1995.
- [5] <http://www.jakpsatweb.cz/zaklady-html.html>
- [6] <http://www.jakpsatweb.cz/css/>
- [7] <http://www.jakpsatweb.cz/javascript/>
- [8] <http://www.phpfusion.cz/news.php>
- [9] http://html.gimiweb.net/page/js_cz.php
- [10] <http://html-history.wz.cz/ramce.html>
- [11] <http://java.sun.com/docs/books/tutorial/deployment/applet/properties.html>
- [12] <http://cz.sun.com/> nejvíce kvalitní zdroj
- [12] <http://www.feec.vutbr.cz/~raida/multimedia>