

Bakalářská práce



České
vysoké
učení technické
v Praze

F3

Fakulta elektrotechnická
Katedra elektromagnetického pole

FDTD analýza zářičů/odražečů zatížených nelineárním prvkem

Josef Růžek

Vedoucí práce: doc. Ing. Milan Polívka, Ph.D.
Studijní program: Elektronika a komunikace
Květen 2024

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Růžek** Jméno: **Josef** Osobní číslo: **507595**
Fakulta/ústav: **Fakulta elektrotechnická**
Zadávající katedra/ústav: **Katedra elektromagnetického pole**
Studijní program: **Elektronika a komunikace**

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

FDTD analýza zářičů/odražečů zatížených nelineárním prvkem

Název bakalářské práce anglicky:

FDTD Analysis of Radiators/Scatterers Loaded by Nonlinear Element

Pokyny pro vypracování:

Metodou konečných diferencí v časové oblasti (FDTD) [1] analyzujte impedanční a zářivé vlastnosti alespoň dvou různých zářičů/odražečů (např. líniový dipól, dvoupásmová anténa pro harmonický transpondér [2]) zatížených nelineárním prvkem, diodou.

Struktury vybudte mono-frekvenčním nebo úzkopásmovým signálem v okolí základního kmitočtu a s využitím transformace z časové do frekvenční oblasti pomocí DFT a transformace EM pole z blízké do vzdálené oblasti určete frekvenční závislosti jejich nelineárních charakteristik, konverzní ztráty a harmonickou efektivní odraznou plochu (HRCS) [3, 4], v oblasti pokrývající druhou i třetí harmonickou frekvenci. Uvažujte volné prostředí a reálnou vodivost objektů.

Využijte dostupné skripty pro prostředí Matlab [1]. Vyšetřované veličiny graficky znázorněte a porovnejte s hodnotami zjištěnými alternativním způsobem, viz např. [2].

Seznam doporučené literatury:

[1] A. Z. Elsherbeni, V. Demir, The Finite Difference Time-domain Method for Electromagnetics with MATLAB Simulations, Scitech Publishing, 2nd ed., 2015.

[2] M. Polivka, V. Hubata-Vacek and M. Svanda, "Harmonic Balance/Full-Wave Analysis of Wearable Harmonic Transponder for IoT Applications," in IEEE Transactions on Antennas and Propagation, vol. 70, no. 2, pp. 977-987, Feb. 2022.

[3] M. A. Richards, J. A. Scheer, W. A. Holm (ed.), Principles of Modern Radar: Basic Principles, Scitech Publishing, 2010.

[4] G. Storz and A. Lavrenko, "Compact Low-cost FMCW Harmonic Radar for Short Range Insect Tracking," 2020 IEEE International Radar Conference (RADAR), Washington, DC, USA, 2020.

Jméno a pracoviště vedoucí(ho) bakalářské práce:

doc. Ing. Milan Polívka, Ph.D. katedra elektromagnetického pole FEL

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) bakalářské práce:

Datum zadání bakalářské práce: **22.02.2024**

Termín odevzdání bakalářské práce: _____

Platnost zadání bakalářské práce: **15.02.2026**

doc. Ing. Milan Polívka, Ph.D.
podpis vedoucí(ho) práce

podpis vedoucí(ho) ústavu/katedry

prof. Mgr. Petr Páta, Ph.D.
podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Student bere na vědomí, že je povinen vypracovat bakalářskou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v bakalářské práci.

Datum převzetí zadání

Podpis studenta

Poděkování

Děkuji svému vedoucímu práce doc. Ing. Milanu Polívkovi, Ph.D. za trpělivost, cenné rady a podporu při psaní této práce.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze, 28. května 2024

Abstrakt

Tato práce představuje základy metody konečných diferencí v časové oblasti, v angl. Finite-Difference Time-Domain (FDTD), vč. její implementace v prostředí MATLAB a její aplikaci pro analýzu impedančních a směrových vlastností zářičů a odražečů dipolového a flíčkového typu zatížených nelineárním prvkem v mikrovlnné části spektra. Při buzení harmonickým signálem jsou analyzovány napětí, proud a střední hodnota výkonu nejprve několika modelů Schottkyho diody různého stupně složitosti obsahující pouze nelinearitu rezistivní části přechodu kov-polodič, dále pak i kapacitu přechodu a sériovou resistivitu polovodiče a v nejdůležitější variantě i parazitní prvky pouzdra diody. Odpovídajícím způsobem jsou modifikovány rovnice pro časový vývoj polí. Po transformaci veličin z časové do frekvenční oblasti jsou určeny konverzní ztráty jednotlivých modelů a harmonická efektivní odrazná plocha v závislosti na velikosti výkonu dopadajícího na diodu. Konverzní ztráty různých typů diod jsou porovnány s hodnotami získanými alternativními metodami.

Klíčová slova: anténa, dioda, FDTD, harmonický transpondér, odražeč, zářič

Vedoucí práce: doc. Ing. Milan Polívka, Ph.D.

Abstract

This paper presents the basics of the finite-difference time-domain method (FDTD) including its implementation in MATLAB and its application to the analysis of impedance and directional properties of dipole and patch-type radiators and reflectors loaded with a nonlinear element in the microwave spectrum. When excited by a harmonic signal, the voltage, current and mean power are analyzed first for several Schottky diode models of varying complexity containing only the nonlinearity of the resistive part of the metal-semiconductor junction, then the junction capacitance and the series resistivity of the semiconductor, and in the most complete version also the parasitic elements of the diode case. The equations for the time evolution of the fields are modified accordingly. After transformation of the quantities from the time to the frequency domain, the conversion losses of each model and the harmonic radar cross section versus the magnitude of the power incident on the diode are determined. The conversion losses of different diode types are compared with the values obtained by alternative methods.

Keywords: antenna, diode, FDTD, harmonic transponder, scatterer, radiator

Title translation: FDTD Analysis of Radiators/Scatterers Loaded by Nonlinear Element

Obsah

| | |
|--------------------------------------------------------------------------------|-----------|
| Úvod | 1 |
| 1 Úvod do FDTD | 3 |
| 1.1 Metoda konečných diferencí | 3 |
| 1.2 Časový vývoj polí v jedné dimenzi | 3 |
| 1.3 Yeeho buňka | 5 |
| 1.4 Numerická stabilita | 6 |
| 1.4.1 Numerická disperze | 9 |
| 1.5 Hraniční podmínky | 10 |
| 1.5.1 Dokonalý vodič | 11 |
| 1.5.2 Dokonale přizpůsobená vrstva | 11 |
| 1.6 Buzení a transformace z časové do frekvenční oblasti | 12 |
| 1.6.1 Sinusový průběh | 13 |
| 1.6.2 Gaussovský průběh | 13 |
| 1.7 Transformace z blízké do vzdálené oblasti | 14 |
| 1.7.1 Princip ekvivalentního povrchu | 16 |
| 2 Implementace FDTD v prostředí MATLAB | 19 |
| 2.1 Struktura kódu | 19 |
| 2.2 Harmonická efektivní odrazná plocha | 19 |
| 2.3 Dioda v FDTD | 20 |
| 2.3.1 Simulace diody | 21 |
| 2.3.2 Výpočet výkonu na diodě . . . | 25 |
| 2.3.3 Konverzní ztráty na diodě . . . | 25 |
| 2.3.4 Náhradní model diody | 27 |
| 2.3.5 Porovnání FDTD s metodou harmonické rovnováhy | 28 |
| 2.4 Liniový dipól zatížený diodou . . | 28 |
| 2.5 Flíčková anténa zatížená diodou | 31 |
| Závěr | 35 |
| Literatura | 37 |
| Slovník | 39 |
| A Vybrané skripty | 41 |
| B Obecné úpravy skriptů | 45 |
| C Definiční skripty pro simulace diod | 51 |
| D Definiční skripty pro simulaci liniového dipólu zatíženého diodou | 57 |
| E Definiční skripty pro simulaci flíčkové antény zatížené diodou | 65 |

Obrázky

| | |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>1.1 Dopředná diference [2] 4</p> <p>1.2 Zpětná diference [2] 4</p> <p>1.3 Centrální diference [2] 5</p> <p>1.4 Jednodimenzionální FDTD [2] ... 5</p> <p>1.5 Prostor vyplněn Yeeho buňkami [2] 6</p> <p>1.6 Yeeho buňka s definovanými složkami E a H [2] 6</p> <p>1.7 Prostor obsahující schodovitou aproximaci koule [2] 7</p> <p>1.8 Průchod vlny mřížkou pod úhlem 45° [1] 7</p> <p>1.9 Příklad nestability. Maximální složka E_{\max} v jednorozměrné úloze pro dva časové kroky, $\Delta t = 3.3356$ ps a $\Delta t = 3.3357$ ps. [2] 8</p> <p>1.10 Vliv parametru α na numerickou disperzi v jedné dimenzi [1] 10</p> <p>1.11 Vliv numerické disperze na tvar pulzu. [1] 10</p> <p>1.12 Odraz vlny od PEC stěn. Vlevo před odrazem, vpravo po odrazu. [2] 11</p> <p>1.13 Problém ohraničen PML ve dvou dimenzích [2] 12</p> <p>1.16 Gaussovský průběh podle [2] .. 16</p> <p>1.17 Spektrum Gaussovského průběhu podle [2] 16</p> <p>1.18 Gaussovský průběh posunutý o t_0 podle [2] 17</p> <p>1.19 Princip ekvivalentního povrchu [2] 17</p> <p>2.1 Ukázka použití Newtonovy - Raphsonovy metody [2] 22</p> <p>2.2 Simulace diody 23</p> <p>2.3 Napětí a proud vzorkované na diodě 24</p> <p>2.4 Napětí a proud vzorkované na diodě s menším saturačním proudem 24</p> <p>2.5 Proud na diodě buzené menší frekvencí 24</p> <p>2.6 Frekvenční spektra veličin na diodě 26</p> <p>2.7 Srovnání konverzních ztrát na různých diodách metodou FDTD . 27</p> <p>2.8 Náhradní model diody pro základní harmonickou složku f_0 dle [3] 28</p> | <p>2.9 Vliv parazitních prvků na průběh konverzních ztrát diody SMS7630 . 29</p> <p>2.10 Schema náhradního modelu diody ideálně přizpůsobené k bráně s výkonovým rozmítáním prvkem HBTuner simulujícím optimální (komplexně sdruženou) impedanci k impedanci diody. 29</p> <p>2.11 Schema porovnání konverzních ztrát pomocí metod HB a FDTD.. 30</p> <p>2.12 Obvodové schema transpondéru sestávající z dipólové antény, (Schottkyho) diody a ladicí indukčnosti L. [10] 30</p> <p>2.13 Geometrie transpondéru podle [10] bez dipólu 31</p> <p>2.14 Naměřená a simulovaná účinnost transpondéru při $f_0/2f_0 = 2,9/5,8$ GHz s $P_{\text{in}} = -15$ dBm: nejlepší účinnost transpondéru η_{opt} (levá osa y) a odpovídající délka dipólu l_{opt}/λ_0 (pravá osa y) [10] 32</p> <p>2.15 Geometrie transpondéru s flíčkovou anténou podle [12] 33</p> |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

Tabulky

| | |
|------------------------------------------------------------------------------------------------------------------------------------|----|
| 2.1 Srovnání simulované směrovosti, vyzařovací účinnosti a zisku ve volném prostoru HB/Full wave metodou[12] a FDTD | 32 |
|------------------------------------------------------------------------------------------------------------------------------------|----|



Úvod

Obor výpočetního elektromagnetismu (Computational Electromagnetism (CEM)) charakterizuje širokou třídu elektromagnetických problémů, jako jsou odrazy či rozptyl od objektů nebo analýza mikrovlnných obvodů a antén, numerickým řešením Maxwellových rovnic v časové oblasti. Obvykle používané metody CEM se dělí do dvou základních kategorií. První je založena na diferenciálních rovnicích, které aproximují derivace pomocí metody konečných diferencí, druhá na integrálních rovnicích, které jsou aproximovány konečnými sumami [2]. Jednou z velmi populárních metod první kategorie je FDTD, metoda konečných diferencí v časové doméně.

V rámci tohoto projektu se zaměřuji na metodu FDTD, která je dostatečně flexibilní ve smyslu řešení složitých geometrií s omezenými objemy vodičů, dielektrik a současně relativně jednoduchá na implementaci. Simuluje chování šíření vln v časové oblasti tak, že počítá elektrické a magnetické pole přímo [1], a dokáže tedy zobrazit i jejich časový průběh, díky čemuž je velmi intuitivní, protože poskytuje lepší vhled do charakteristik problémů [2]. Nevýhodou je větší výpočetní náročnost pro větší velikost řešené úlohy, nutnost implementovat okrajové podmínky pro vyzařující struktury a použít algoritmy pro transformaci veličin z časové do frekvenční oblasti resp. z blízké do vzdálené oblasti. Tyto algoritmy jsou však velmi dobře zavedeny a správnost jejich výstupů prověřena.

V první části projektu 1 se věnuji teoretickému odvození a principu fungování metody. V druhé části 2 je v prostředí MATLAB představena implementace pro analýzu transpondérů zatíženými nelineárním prvkem, jejíž výsledky jsou porovnány s analytickou analýzou a metodou harmonické rovnováhy [3][4].

Kapitola 1

Úvod do FDTD

1.1 Metoda konečných diferencí

FDTD je metoda konečných diferencí Finite-Difference (FD) implementována v časové oblasti. Myšlenka této metody je, že derivace, tedy změna hodnoty při posunu v definičním oboru o nekonečně malou část, může být pro numerický výpočet nahrazena diferencí. Na rozdíl od derivace, u difference je posun v definičním oboru konečný. Díky tomuto zjednodušení je možné mnoho problémů počítačově simulovat za cenu určité nepřesnosti, která roste s velikostí kroku zvoleného v definičním oboru.

Pokud zjišťujeme hodnotu v bodě x , existují celkem tři možnosti, jak z derivace vyjádřit diferencí. První je použít dopřednou diferencí (viz obr. 1.1)

$$f'(x) \approx \frac{f(x + \Delta x) - f(x)}{\Delta x}, \quad (1.1)$$

zpětnou diferencí (viz obr. 1.2),

$$f'(x) \approx \frac{f(x) - f(x - \Delta x)}{\Delta x}, \quad (1.2)$$

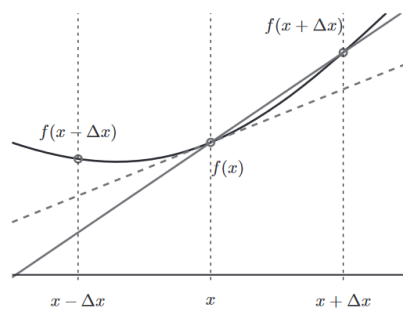
nebo centrální diferencí (viz obr. 1.3)

$$f'(x) \approx \frac{f(x + \Delta x) - f(x - \Delta x)}{2\Delta x}. \quad (1.3)$$

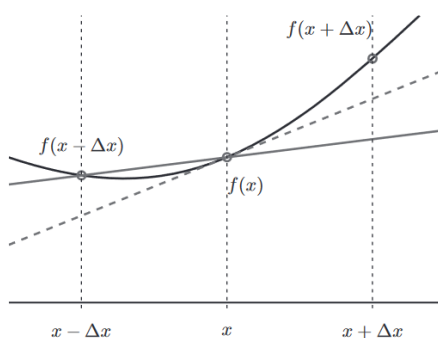
Rozvojem pomocí Taylorovy řady lze snadno zjistit, že chybový člen (rozdíl difference od derivace) je pro dopřednou i zpětnou diferencí řádu jedna, ale pro centrální řádu dva [1]. Proto je užití centrální difference preferované.

1.2 Časový vývoj polí v jedné dimenzi

Princip metody FDTD spočívá ve výpočtu složek elektrického pole E a magnetického pole H v každém časovém kroku. Pro jednoduchost jako vhodná



Obrázek 1.1: Dopředná diference [2]



Obrázek 1.2: Zpětná diference [2]

ukázka může posloužit výpočet polí pro jednodimenzionální problém, analýza šíření vlny po přenosovém vedení [1]. Protože rovnice aktualizující hodnotu \mathbf{E} a \mathbf{H} vycházejí z diferenciálního tvaru prvních dvou Maxwellových rovnic

$$\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t}, \quad (1.4)$$

$$\nabla \times \mathbf{H} = \frac{\partial \mathbf{D}}{\partial t} + \mathbf{J}, \quad (1.5)$$

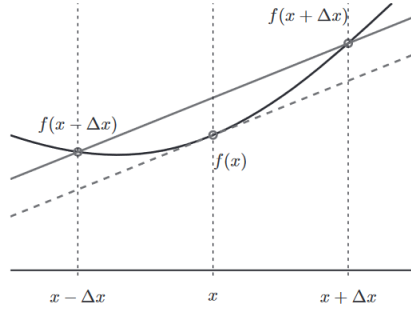
řešení v jednorozměrném zajistí nejen třikrát méně rovnic, ale i jednodušší výrazy, protože výsledek operace rotace bude mít většinu složek rovných nule. Za předpokladu, že vlna TEM¹ se šíří podle osy x a proudová hustota \mathbf{J} je nulová, Maxwellovy rovnice se zjednoduší na:

$$\epsilon_0 \frac{\partial E_y}{\partial t} = -\frac{\partial H_z}{\partial x}, \quad (1.6)$$

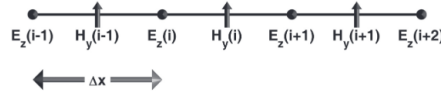
$$\mu_0 \frac{\partial H_z}{\partial t} = -\frac{\partial E_y}{\partial x}. \quad (1.7)$$

Nejprve diskretizujeme prostorovou doménu na buňky o velikosti Δx a časovou na kroky o velikosti Δt . Uzly, ve kterých bude docházet k časové

¹Elektrické a magnetické pole mají pouze složky kolmé na směr šíření.



Obrázek 1.3: Centrální diference [2]



Obrázek 1.4: Jednodimenzionální FDTD [2]

aktualizaci složek elektrického a magnetického pole, si definujeme podle [1] ve středu buněk. Nyní je možné aplikovat algoritmus FDTD pro $x_i = i\Delta x$ a $t_n = (n + 1/2)\Delta t$, kde i je číslo uzlu a n číslo časového kroku. V každém časovém kroku se pomocí FDTD aktualizuje hodnota složek polí ve všech uzlech následovně:

$$\frac{E_y^{(n+1)}(i) - E_y^{(n)}(i)}{\Delta t} = \frac{-1}{\epsilon_0} \frac{H_z^{(n+1/2)}(i + \frac{1}{2}) - H_z^{(n+1/2)}(i - \frac{1}{2})}{\Delta x}, \quad (1.8)$$

$$\frac{H_z^{(n+1/2)}(i + \frac{1}{2}) - H_z^{(n-1/2)}(i + \frac{1}{2})}{\Delta t} = \frac{-1}{\mu_0} \frac{E_y^{(n)}(i + 1) - E_y^{(n)}(i)}{\Delta x}. \quad (1.9)$$

Z rovnic (1.8) a (1.9) vyplývá, že elektrické pole je aktualizováno v čase $n\Delta t$, zatímco magnetické pole v čase $(n + 1/2)\Delta t$, viz obr. 1.4. Toto střídavé aktualizování hodnot (v angl. "leap frog marching" of fields) snižuje výpočetní čas na polovinu. Úpravou předchozích rovnic lze jednoduše získat

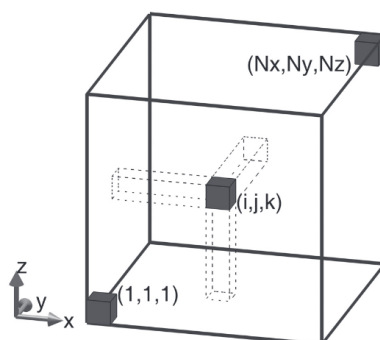
$$E_y^{(n+1)}(i) = E_y^{(n)}(i) + \frac{\Delta t}{\epsilon_0 \Delta x} (H_z^{(n+1/2)}(i - \frac{1}{2}) - H_z^{(n+1/2)}(i + \frac{1}{2})), \quad (1.10)$$

$$H_z^{(n+1/2)}(i + \frac{1}{2}) = H_z^{(n-1/2)}(i + \frac{1}{2}) + \frac{\Delta t}{\mu_0 \Delta x} (E_y^{(n)}(i) - E_y^{(n)}(i + 1)), \quad (1.11)$$

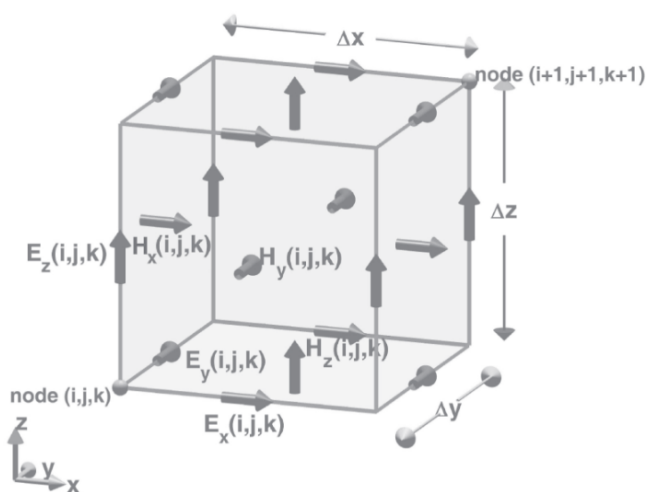
což jsou rovnice pro výpočet nových hodnot E a H , které je možné implementovat v kódu.

1.3 Yeeho buňka

Pokud řešíme problém ve třech dimenzích, výpočetní mřížka je složena ze základních prvků, tzv. Yeeho buněk, viz obr. 1.5. Složky polí E a H jsou



Obrázek 1.5: Prostor vyplněn Yeeho buňkami [2]

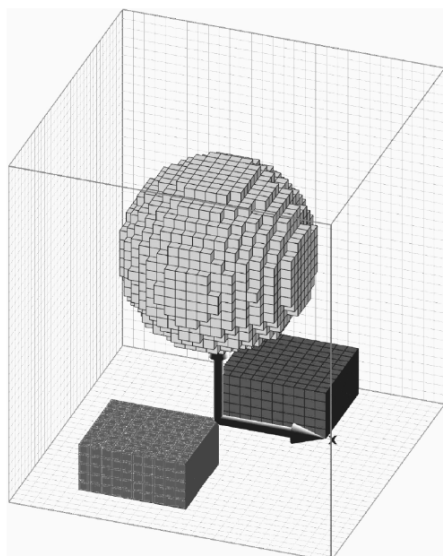


Obrázek 1.6: Yeeho buňka s definovanými složkami E a H [2]

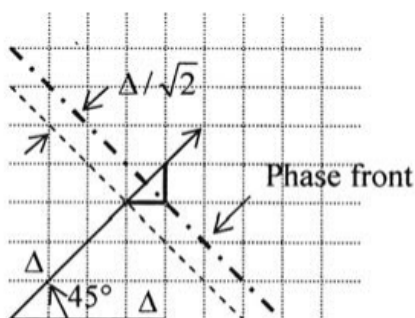
specificky uspořádané tak, že složky E jsou umístěny ve středu příslušných hran, se kterými jsou rovnoběžné a složky H vychází kolmo ze středu stěn buňky, viz obr. 1.6. To znamená, že Yeeho buňky tvoří prostor popsáný řadou Faradayových a Ampérových zákonů, neboť každou složku H obklopují čtyři složky E a naopak, kromě buňek na hranici, pro které musí být definovány speciální aktualizací rovnice popsané v sekci 1.5. Yeeho buňkami je možné modelovat i objekty, které nesplývají s jejich hranou pomocí tzv. schodovité aproximace, viz obr. 1.7, avšak čím více se velikost Yeeho buňky blíží k velikosti modelovaného objektu, tím menší bude přesnost simulace.

1.4 Numerická stabilita

Protože FDTD algoritmus vzorkuje elektrické a magnetické pole v diskretních bodech v prostoru i v čase, volba vzorkovacích period má určitá omezení, která



Obrázek 1.7: Prostor obsahující schodovitou aproximaci koule [2]



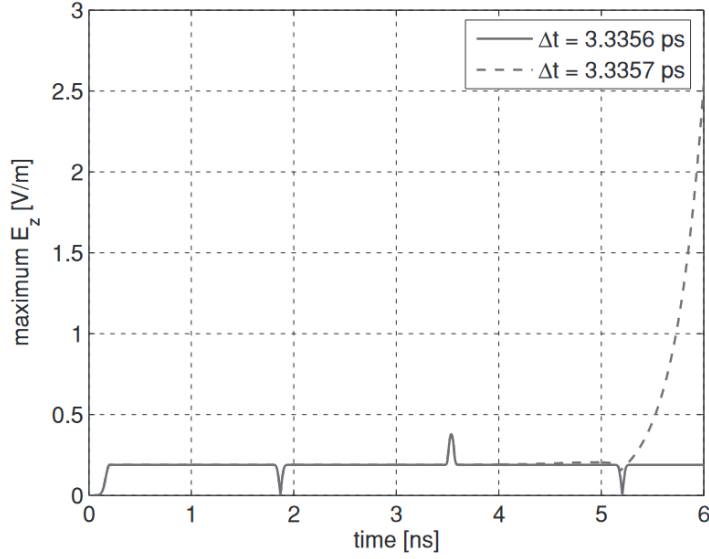
Obrázek 1.8: Průchod vlny mřížkou pod úhlem 45° [1]

ovlivňují jak stabilitu simulace, tak i přesnost výsledků [2]. Musí být splněna tzv. Courant-Friedrich-Levy (CFL) podmínka [5], která je v jednorozměrné FDTD reprezentována vztahem

$$\Delta t \leq \frac{\Delta x}{v_{\max}}, \quad (1.12)$$

kde Δt je perioda vzorkování v čase, Δx je perioda vzorkování v prostoru a v_{\max} je maximální fázová rychlost ve výpočetní oblasti [1]. Z (1.12) vyplývá, že volba Δt musí být taková, aby vlna nemohla urazit vzdálenost větší, než je Δx , a přeskočit tak navzorkování v následujícím uzlu.

Pokud do problému přidáme další dimenzi, vlna nyní může procházet mřížkou různými směry. Podmínku nejvíce ovlivní průchod pod úhlem 45°, viz obr. 1.8. Z obrázku je jasně patrné, že vlna šířící se tímto způsobem urazí



Obrázek 1.9: Příklad nestability. Maximální složka E_{\max} v jednorozměrné úloze pro dva časové kroky, $\Delta t = 3.3356$ ps a $\Delta t = 3.3357$ ps. [2]

nejkratší možnou vzdálenost předtím, než doputuje k následujícím uzlům. Tento fakt se projeví v rovnici pro CFL,

$$v_{\max}\Delta t \leq \frac{1}{\sqrt{\frac{1}{(\Delta x)^2} + \frac{1}{(\Delta y)^2}}}, \quad (1.13)$$

kde Δy je rozměr mřížky podél osy y . Pokud $\Delta x = \Delta y$, je možné zjednodušit na

$$\Delta t \leq \frac{\Delta x}{v_{\max}\sqrt{2}}. \quad (1.14)$$

Stejným postupem lze odvodit CFL i pro třetí prostorový rozměr,

$$v_{\max}\Delta t \leq \frac{1}{\sqrt{\frac{1}{(\Delta x)^2} + \frac{1}{(\Delta y)^2} + \frac{1}{(\Delta z)^2}}}, \quad (1.15)$$

kde je navíc člen Δz , rozměr mřížky ve směru osy z . Za předpokladu, že $\Delta x = \Delta y = \Delta z$, vztah pro Δt vyjde

$$\Delta t \leq \frac{\Delta x}{v_{\max}\sqrt{3}}. \quad (1.16)$$

Ukázka nesplnění CFL podmínky je zobrazena v obrázku 1.9. Jedná se o FDTD v jedné dimenzi, kde podmínka stability vyšla $\Delta t \geq 3.3356$ ps.

1.4.1 Numerická disperze

FD implementace derivací dělá řešení pouze přibližné, z toho důvodu se fázová konstanta mění s frekvencí v prostředí, které je jinak nedisperzní. Této disperzi se říká numerická [5][6]. Velikost disperze je závislá na vlnové délce, volbě kroku Δt a, pro příklad v jedné dimenzi, Δx . Pro ukázkou použijí postup z [1], kde se vychází z vlnové rovnice

$$\frac{1}{c^2} \frac{\partial^2 E}{\partial t^2} = \frac{\partial^2 E}{\partial x^2}, \quad (1.17)$$

a monochromatické, sinusové řešení postupné vlny

$$E_y(x, t) = e^{j(\omega t - kx)}, \quad (1.18)$$

s konstantou šíření

$$k = \pm \frac{\omega}{c}. \quad (1.19)$$

Po vyjádření (1.17) a (1.18) centrální diferencí, dosazením do sebe a úpravách získáme

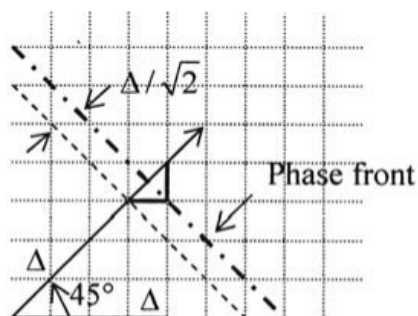
$$\sin\left(\frac{\omega \Delta t}{2}\right) = \pm \left(\frac{c \Delta t}{\Delta x}\right) \sin\left(\frac{\bar{k} \Delta x}{2}\right), \quad (1.20)$$

kde \bar{k} je konstanta šíření numerické vlny, která vznikla diskretizací (1.18). Této rovnici se říká disperzní relace a dává do poměru ω s \bar{k} . \bar{k} a k se obecně liší, což způsobuje numerické chyby dokazující přítomnost numerické disperze. Je snadné ověřit, že pro $\Delta x \rightarrow 0$ a $\Delta t \rightarrow 0$ se rovnice zjednoduší zpět do podoby v (1.19). Po vydělení rovnice Δt a aplikování identity $\lim_{x \rightarrow 0} \frac{\sin(x)}{x} = 1$ lze levou a pravou stranu upravit následovně:

$$\lim_{\Delta t \rightarrow 0} \frac{\sin\left(\frac{\omega \Delta t}{2}\right)}{\Delta t} = \lim_{\Delta t \rightarrow 0} \frac{\sin\left(\frac{\omega \Delta t}{2}\right)}{\frac{\Delta t}{2}} \cdot \frac{1}{2} = \frac{\omega}{2}, \quad (1.21)$$

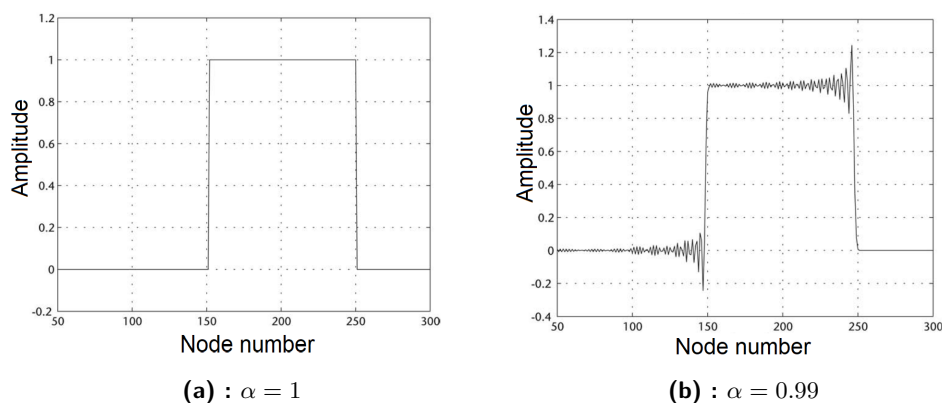
$$\lim_{\Delta x \rightarrow 0} \frac{\sin(\bar{k} \Delta x) \cdot c}{\Delta x} = \lim_{\Delta x \rightarrow 0} \left(\frac{c}{\Delta x}\right) \frac{\sin\left(\frac{\bar{k} \Delta x}{2}\right)}{\frac{\Delta x}{2}} \cdot \frac{1}{2} = \frac{c \bar{k}}{2}. \quad (1.22)$$

To potvrzuje, že pro nekonečně malý krok platí $k = \bar{k}$ k numerické disperzi nedochází. Disperze ale zmizí navíc v případě $\Delta t = \Delta x/c$, čemuž se říká magický časový krok [1]. Aproximace dvojité derivací v (1.17) centrální diferencí zanechá na obou stranách rovnice chybové členy, které se vykrátí právě v tomto případě, a tak získáme přesné řešení. Pro přehlednější vyjádření poměru $\Delta t = \Delta x/c$ zavádí [1] poměr stability $\alpha = c \Delta t / \Delta x$. Fázová rychlost $\bar{v}_p = \omega / \bar{k}$ v poměru ku c je vykreslena na obr. 1.10 jako funkce velikosti buňky a α jako parametrem. Lze vidět, že pro velikost mřížky podstatně menší než je vlnová délka jsou rozdíly ve \bar{v}_p/c minimální, ale s rostoucím Δx začne na parametru α více záležet. Jednoduchý příklad demonstrující



Obrázek 1.10: Vliv parametru α na numerickou disperzi v jedné dimenzi [1]

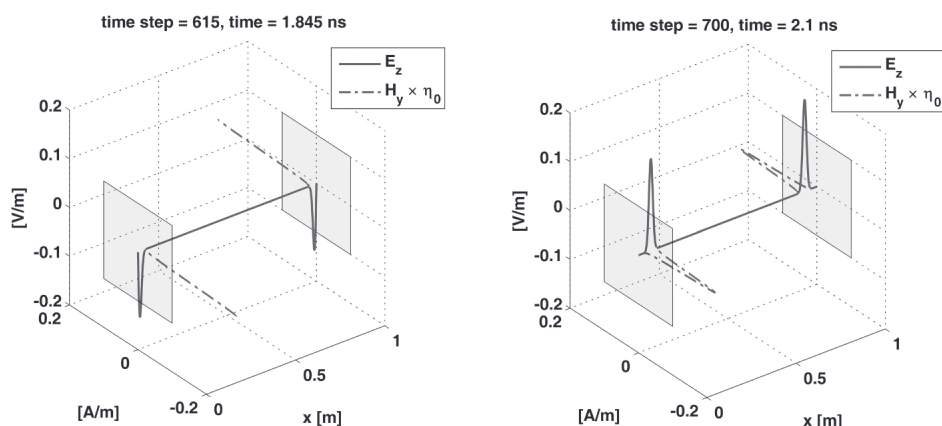
efekt numerické disperze na vlnu v jedné dimenzi je na obr. 1.11. Numerická disperze způsobuje, že vlny s odlišnými frekvencemi se šíří různými fázovými rychlostmi. Vzhledem k tomu, že jakýkoli signál lze vyjádřit jako součet sinusových vln s rozdílnými frekvencemi, tento signál během šíření mění svůj tvar [2].



Obrázek 1.11: Vliv numerické disperze na tvar pulzu. [1]

1.5 Hraniční podmínky

Jelikož mřížka, kterou je nutné vytvořit pro FDTD algoritmus, má konečnou délku, je nutné ošetřit případy na hranicích výpočetní oblasti. Tam rovnice jako (1.10) a (1.11) pro nové hodnoty elektrického a magnetického pole nemohou nadále fungovat, protože neexistují sousední uzly, ze kterých by bylo možné získávat hodnoty. Hraničním uzlům, popřípadě i více vrstvám uzlů od okraje, se vnutí speciálním algoritmem hodnoty, tzv. hraniční podmínky, zajišťující správné chování dopadající vlny. Například Murova absorbní podmínka prvního řádu [1] zajišťuje bezodrazovost a vychází z faktu, že bezodrazová je hranice tehdy, když vlna pokračuje v šíření pouze v dopředném směru. Podobné úvahy mohou vést na mnoho různorodých hraničních podmínek.



Obrázek 1.12: Odraz vlny od PEC stěn. Vlevo před odrazem, vpravo po odrazu. [2]

Cílem této kapitoly je však poskytnout základní přehled těch nejpoužívanějších, dokonalého elektrického vodiče (PEC) a dokonale přizpůsobené vrstvy (PML).

1.5.1 Dokonalý vodič

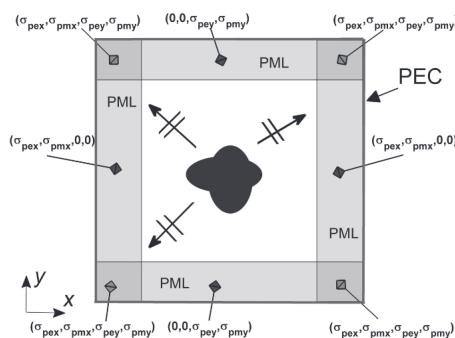
Dokonalý vodič, v angl. Perfect Electric Conductor (PEC)² reprezentuje vodič, který je schopný vést neomezený proud bez jakékoli ztráty energie. To znamená, že povrch PEC má vždy stejný elektrický potenciál a tečná složka elektrického pole na jeho povrchu musí být nulová [7]. Při dopadu vlny na PEC se odrazí vlna se stejnou amplitudou, ale opačnou orientací, aby se splnila podmínka nulové tečné složky na povrchu, což zapříčiní otočení směru elektromagnetické vlny, neboť magnetické pole se odrazí beze změny, viz obr. 1.12. Tato hraniční podmínka je jednoduchá na implementaci a používá se pro modelování nevyzařujících struktur, u kterých se nemůže stát, že by se výpočetní prostor zahltil nespočetnými odrazy.

1.5.2 Dokonale přizpůsobená vrstva

Zatímco PEC zaručuje ideální odrazy vln od hranic vymezeného prostoru, v mnoha aplikacích, zejména problémů zářičů a odražečů, je potřeba naopak simulovat volný prostor. Hraniční podmínky, které modelují šíření vln za hranice výpočetní oblasti se nazývají absorbující a jednou z těch nejspolehlivějších se ukázala být metoda dokonale přizpůsobené vrstvy, v angl. Perfectly Matched Layer (PML) [2].

Velkým příspěvkem na odrazech je nesoulad impedancí mezi absorbujícím a původním prostředím, když dopadá vlna pod různými úhly [1]. Souladu

²Na stejném principu funguje dokonalý magnetický vodič, v angl. Perfect Magnetic Conductor, avšak pro složku magnetického pole.



Obrázek 1.13: Problém ohraničen PML ve dvou dimenzích [2]

se dosáhne rozdělením elektrických a magnetických složek v absorbujičím prostředí a přiřazením různých vodivostí σ pro jednotlivé složky. Efekt tohoto přístupu je vytvoření prostředí, ve kterém je vlnová impedance nezávislá na úhlu dopadu a frekvenci. Příklad použití metody PML ve dvou dimenzích je na obr. 1.13. Problém obsahuje složky E_x , E_y , a H_z , která se dělí na dvě fiktivní komponenty H_{zx} a H_{zy} . Každá z hraničních oblastí má nenulové vodivosti pro složky, které do ní vstupují kolmo.

1.6 Buzení a transformace z časové do frekvenční oblasti

Pro frekvenční analýzu systému je klíčové buzení správným časovým průběhem, který obsahuje kýžené frekvenční složky. Zároveň by buzení mělo mít hladký přechod při zapnutí a vypnutí, aby se minimalizoval vliv vysokofrekvenčních komponent [2]. Brát v potaz se musí také vzdálenost mezi uzly mřížky, což je parametr, který omezuje maximální frekvenci, jejíž analýzu je možné provést s dostatečnou přesností. Důvodem je vlnová délka, která se na vysokých kmitočtech může stát srovnatelnou s rozměry mřížky. Obecně se za mez, kde hodnoty nejsou příliš zatížené numerickou chybou, považuje vlnová délka alespoň dvacetkrát větší než rozměr mřížkové buňky [2][1]. Mezi nejčastěji používané budící funkce patří sinus, který ale má v ideálním případě ve spektru jedinou frekvenční složku a není tedy vhodný pro širokopásmovou analýzu, a gaussovský průběh, který může být při vhodném nastavení délky trvání i velmi širokopásmový.

Protože předpis budících funkcí je známý, frekvenční spektrum lze získat pomocí Furierovy transformace [2]

$$X(\omega) = \int_{-\infty}^{\infty} x(t) \cdot e^{-j\omega t} dt, \quad (1.23)$$

kde ω je úhlová frekvence. Pokud je třeba zjistit spektrum funkce, pro kterou není možné znát předpis, např. spektrum pole odraženého od objektu nebo

výkonu na obvodovém prvku (viz kapitola 2), použije se diskretní Furierova transformace (DFT)³

$$X[k] = \sum_{n=0}^{N-1} x[n] \cdot e^{-j2\pi \frac{kn}{N}}, \quad (1.24)$$

kde k je index spektrální čáry ($f = k \frac{fs}{N}$), fs je vzorkovací frekvence, a N je počet vzorků.

1.6.1 Sinusový průběh

Jak bylo zmíněno výše, ideální resp. nekonečný sinusový průběh má ve spektru pouze složku odpovídající své frekvenci. Simulaci ovšem můžeme budít jen po omezenou dobu a součástí průběhu musí být začátek a konec funkce, což vnese do spektrální oblasti další frekvence. Vliv těchto frekvencí je možné korigovat využitím periodicity sinu. Čím delší dobu necháme sinusem budít systém, tím více period bude k dispozici pro Furierovu transformaci a chybné frekvence budou potlačeny. Na obrázku 1.14a je úkazka buzení sinusem o frekvenci 1 Hz po čtyři sekundy. Spektrum sinu je ukázáno na obrázku 1.14b. Při buzení po dvojnásobnou dobu se zdvojnásobí i počet period, viz obr. 1.15a. Je jasně patrné, že frekvence sinu⁴ je mnohem výraznější, než v předchozím případě.

1.6.2 Gaussovský průběh

Gaussovský průběh nebo také pulz, viz obr. 1.16, může být zapsán jako funkce času,

$$g(t) = e^{-\frac{t^2}{\tau^2}}, \quad (1.25)$$

kde τ je parametr určující šířku Gaussovského pulzu jak v časové doméně, tak ve frekvenční doméně. Fourierova transformace Gaussovského průběhu, viz obr. 1.17, je také Gaussovský průběh, který může být vyjádřen jako funkce frekvence [2]

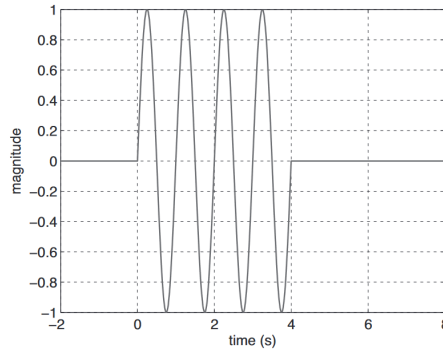
$$G(\omega) = \tau \sqrt{\pi} e^{-\frac{t^2 \omega^2}{4}}. \quad (1.26)$$

Maximální dovolená frekvence je známá hodnota odvozená od rozměru mřížky. Zároveň u spektra Gaussovského pulsu za nejvyšší validní frekvenci považujeme takovou, kde je amplituda 10% maximální hodnoty spektra pulsu. Parametr τ tedy musí být volen tak, aby se tyto dvě hodnoty rovnaly. Vycházíme z rovnice

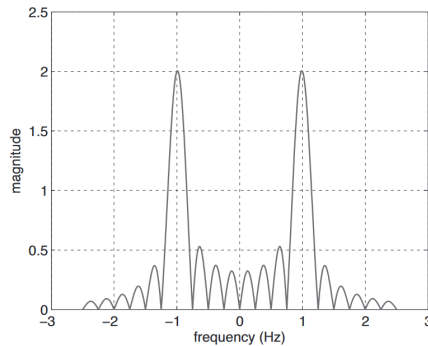
$$0.1 = e^{-\frac{t^2 \omega_{(max)}^2}{4}}, \quad (1.27)$$

³DFT bývá implementována jako rychlá Furierova transformace (FFT)

⁴Symetrické spektrum je vlastnost Furierovy transformace, tím pádem špička na frekvenci -1 Hz v 1.14b a 1.15b je pouze obrazem.



(a) : Sinusový průběh buzen 4 sekundy [2]



(b) : Sinusový průběh buzen 4 sekundy: spektrum [2]

ze které se vyjádří τ :

$$\tau = \frac{\sqrt{2.3}}{\pi f_{\max}}. \quad (1.28)$$

Gaussovský puls tak, jak je definován v (1.25), ještě není možné implementovat pro FDTD, protože začíná v záporných hodnotách t a maximum má v čase nula. Naopak v FDTD simulaci jsou hodnoty zpočátku nulové, a tak musejí být nulové i zdroje. Proto je třeba rovnici (1.25) upravit tak, aby na začátku simulace měl puls hodnotu nulovou či zanedbatelnou. Posun v čase zajistí přidaný člen t_0 ,

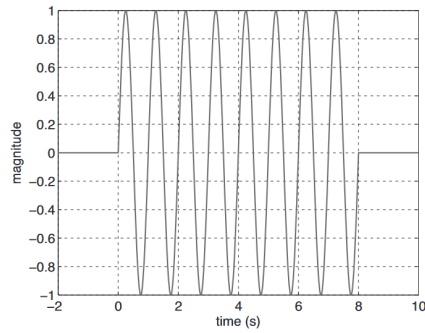
$$g(t) = e^{-\frac{(t-t_0)^2}{\tau^2}}. \quad (1.29)$$

viz obr. 1.18. Určení přesné hodnoty t_0 závisí na žádoucí přesnosti. [2] uvádí, že pro $g(0) = e^{-20}$, což stačí na více než 16 číslic numerické přesnosti, platí

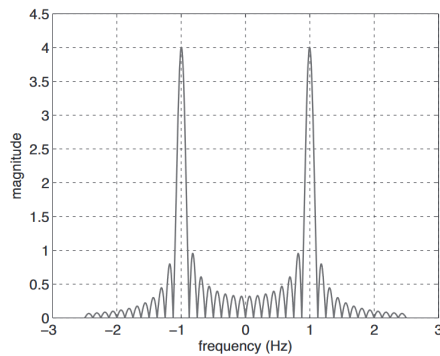
$$t_0 = \sqrt{20}\tau \approx 4.5\tau. \quad (1.30)$$

1.7 Transformace z blízké do vzdálené oblasti

Parametrem, který nejvýrazněji ovlivňuje trvání simulace, je velikost výpočetního prostoru, respektive množství uzlů, které se v prostoru nachází. Pro



(a) : Sinusový průběh buzen 8 sekund [2]



(b) : Sinusový průběh buzen 8 sekund: spektrum [2]

problémy, kde se vypočítávají E a H v blízkém okolí nějaké elektromagnetické struktury je doba simulace krátká. Tato oblast se nazývá Near-Field (NF). Některé simulace ale vyžadují hodnoty E a H ve vzdálené oblasti, Far-Field (FF). Jedná se například o zářiče nebo odražeče, které jsou excitovány dopadající vlnou, jejichž implementace je tématem kapitoly 2. Jednoduchá podmínka pro FF je

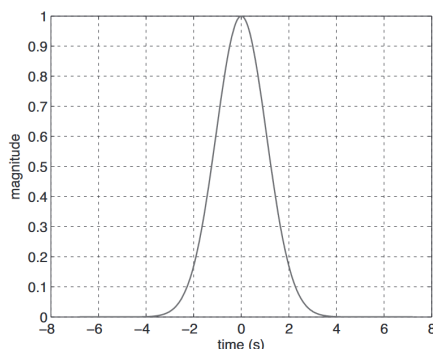
$$kR \gg 1 \implies \frac{2\pi R}{\lambda} \gg 1, \quad (1.31)$$

kde k je vlnové číslo, R je vzdálenost od zářiče a λ je vlnová délka [2]. Pro elektricky velkou anténu jako je parabolický odražeč se také používá vztah obsahující velikost aparatury D :

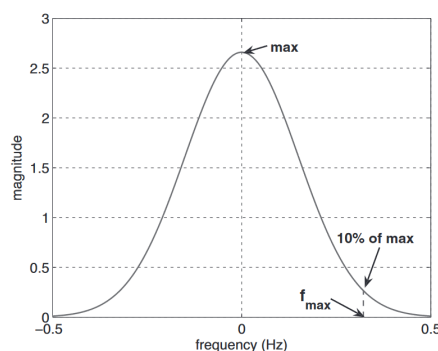
$$r > \frac{2D^2}{\lambda}, \quad (1.32)$$

kde r je vzdálenost od středu aparatury.[8]

Pro přímý získání hodnot ve vzdálené oblasti by FDTD algoritmu bylo nutné nastavit velmi rozsáhlou výpočetní oblast, která by drasticky zvýšila výpočetní dobu. Proto je pro simulaci tohoto typu úloh nutné použít NF-FF transformaci, která umožňuje získat hodnoty z FF nepřímou cestou bez potřeby zvětšit výpočetní prostor.



Obrázek 1.16: Gaussovský průběh podle [2]



Obrázek 1.17: Spektrum Gaussovského průběhu podle [2]

1.7.1 Princip ekvivalentního povrchu

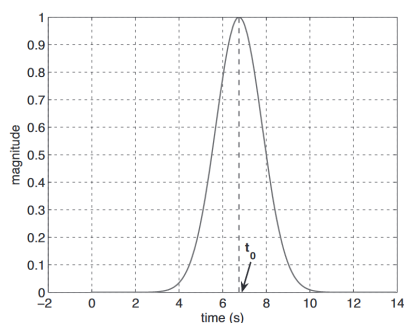
Základem pro NF-FF transformaci je princip ekvivalentního povrchu (ang. Surface Equivalence Theorem). Zdroje ve výpočetní oblasti jako jsou antény a odražeče se nahradí fiktivními proudy na povrchu, který prostor obklopuje.[2] Tyto proudy mají takovou hodnotu, aby si vyzářená pole E a H před a po aplikování teorému byla rovna, viz obr. 1.19.

Pokud jsou známy hodnoty polí vstupující do povrchu, povrchové proudy je možné získat pomocí rovnic [2]

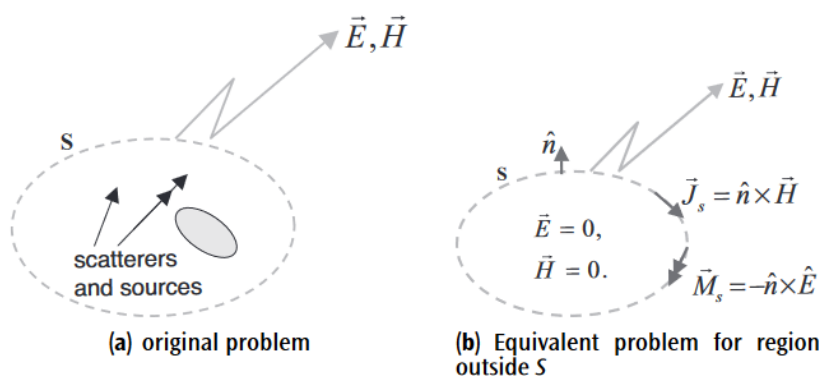
$$\vec{J}_S = \hat{n} \times (\vec{H}^{\text{out}} - \vec{H}^{\text{in}}) = \hat{n} \times \vec{H}, \quad (1.33)$$

$$\vec{M}_S = -\hat{n} \times (\vec{E}^{\text{out}} - \vec{E}^{\text{in}}) = -\hat{n} \times \vec{E}. \quad (1.34)$$

Poté je možné spočítat hodnoty polí v libovolně vzdáleném observačním bodu pomocí metody vektorových potenciálů, ze kterých se vypočítají hodnoty pole za použití podmínek pro FF.



Obrázek 1.18: Gaussovský průběh posunutý o t_0 podle [2]



Obrázek 1.19: Princip ekvivalentního povrchu [2]

Zavedení věty o ekvivalenci povrchu zjednodušuje výpočet vzdáleného pole, protože uvnitř povrchu S mohou být materiály s různými permitivitami a permeabilitami, a tak je potřeba pro výpočet odvozovat komplexní Greenovu funkci vyzařujícího pole. Po aplikování teorému jsou ale pole uvnitř povrchu nulová a permitivita a permeabilita mohou být nastaveny stejně jako ve vnějším volném prostoru. Poté se k výpočtu vyzařujícího pole použije jednoduchá Greenova funkce pro volný prostor.[2]

Kapitola 2

Implementace FDTD v prostředí MATLAB

2.1 Struktura kódu

Pro simulace byly použity MATLAB skripty připravené v [2] přizpůsobené pro konkrétní scénář. Hlavní skript je A.1, ve kterém se volají subskripty, jejichž úkolem je definování geometrie mřížky a objektů v simulaci, výpočet konstant, samotný FDTD algoritmus, zobrazení grafů a mnoho dalších. Takový přístup je výhodný, protože rozděluje kód na dílčí výpočetní procedury a dává přímou a intuitivní kontrolu nad parametry simulace. V A.1 se volají jako první čtyři definiční skripty: *define_geometry*, *define_output_parameters*, *define_problem_space_parameters*, *define_sources_and_lumped_elements* a několik skriptů inicializující výpočetní oblast úlohy a parametry FDTD metody. Pokud uživatel nepotřebuje funkcionalitu, která by nebyla pokrytá v [2], jsou tyto čtyři soubory jediné, které je potřeba upravit pro řešení zvolené úlohy.

Po inicializačních skriptech se samotný FDTD algoritmus spustí v A.2. Doba výpočtu je určena počtem časových kroků, což je hodnota definovaná v *define_problem_space_parameters*. Lze vidět, že je dodrženo pořadí aktualizací zmíněné v 1.2. Nejprve jsou aktualizovány hodnoty magnetického pole v časových krocích $n + 1/2$, a poté hodnoty elektrického pole v krocích $n + 1$.

2.2 Harmonická efektivní odrazná plocha

Jednou z veličin, která bude v následujících sekcích počítána, je harmonická efektivní odrazná plocha, z angl. Harmonic Radar Cross Section (HRCS). Zatímco standardní RCS, která je implementována i ve skriptech z [2] měří schopnost cíle odrážet radarové signály na stejné frekvenci, na které byly vyslány, HRCS měří schopnost cíle odrážet frekvence harmonické. Výpočet HRCS je podle [8]

$$\text{HRCS} = G_{\text{FRX}} G_{\text{HTX}} \frac{\lambda_F^2}{4\pi} \eta_H, \quad (2.1)$$

kde G_{FRX} a G_{HTX} jsou zisky antény na základní a první harmonické frekvenci,

λ_F je vlnová délka základní frekvence a η_H je konverzní účinnost, převrácená hodnota Conversion Loss (CL). Pro výpočet zisku vycházím z rovnice [7]

$$G = D\eta, \quad (2.2)$$

kde D je směrovost a η je vyzařovací účinnost antény. Směrovost se ve skriptech počítá pro zobrazení RCS v polárním grafu a je to vektor hodnot závislých na úhlu. Do rovnice 2.2 dosazují maximální hodnotu z vektoru. Vyzařovací účinnost je definována vztahem [2]

$$\eta = \frac{P_{\text{rad}}}{P_{\text{del}}} \quad (2.3)$$

kde P_{rad} je výkon vyzářený a P_{del} je výkon dodaný zdrojem, který se snadno spočítá stejně, jako výkon na diodě, viz sekce 2.3.2. Výpočet vyzářeného výkonu je součástí funkcionality skriptů z [2] a je počítán s pomocí principu ekvivalentního povrchu rovnicí

$$P_{\text{rad}} = \frac{1}{2} \text{Re} \int_S (\mathbf{E} \times \mathbf{H}^*) \cdot \mathbf{n} dS = \frac{1}{2} \text{Re} \int_S (\mathbf{J}^* \times \mathbf{M}) \cdot \mathbf{n} dS. \quad (2.4)$$

2.3 Dioda v FDTD

Předtím, než přejdu k celým strukturám zatížených nelineárním prvkem, by bylo vhodné vysvětlit implementaci prvku samotného, kterým pro všechny následující simulace bude dioda. V důsledku nelineární závislosti mezi proudem diodou a napětím na ní, je implementace rovnic pro aktualizaci složek polí o něco složitější než v případě základních obvodových prvků (rezistor, induktor a kapacitor) [2].

Dioda je charakterizována vztahem mezi proudem a napětím popsaným následující rovnicí:

$$I = I_d = I_0 \left[e^{\frac{qV_d}{kT}} - 1 \right], \quad (2.5)$$

kde q je absolutní hodnota náboje elektronu v coulombech, k je Boltzmannova konstanta a T je absolutní teplota v kelvinech. V některých zdrojích se člen $\frac{q}{kT}$ definuje jako termální napětí V_t . [9] Rovnici 2.5 je možné přepsat pro FDTD v diskrétní formě jako

$$J_{i,z}^{n+\frac{1}{2}}(i, j, k) = \frac{I_0}{\Delta x \Delta y} \left(e^{\left(\frac{q\Delta z}{2kT}\right)(E^{n+1}(i,j,k) + E^n(i,j,k))} - 1 \right) \quad (2.6)$$

kde je použit vztah pro diodu ve směru z , $V_d = \Delta V = \Delta z E$. Tato proudová hustota J se poté dosadí do rovnice

$$\frac{\partial E_z}{\partial t} = \frac{1}{\varepsilon_z} \left(\frac{\partial H_y}{\partial x} - \frac{\partial H_x}{\partial y} - \sigma_z^e E_z - J_{iz} \right) \quad (2.7)$$

což je jedna z šesti rovnic reprezentujících Maxwellovy rotační rozložené na jednotlivé složky. Pro směr z byla zvolena rovnice obsahující derivaci složky E_z . Po aproximaci centrální diferencí má podobu

$$\begin{aligned} \frac{E_z^{n+\frac{1}{2}}(i, j, k) - E_z^{n-\frac{1}{2}}(i, j, k)}{\Delta t} = & \frac{\Delta t}{\varepsilon_z(i, j, k)} \frac{H_y^n(i, j, k) - H_y^n(i-1, j, k)}{\Delta x} \\ & - \frac{H_x^n(i, j, k) - H_x^n(i, j-1, k)}{\Delta y} \\ & - \frac{\sigma_z(i, j, k)}{2\varepsilon_z(i, j, k)} (E_z^{n+1}(i, j, k) + E_z^n(i, j, k)) \\ & - \frac{1}{\varepsilon_z(i, j, k)} J_{iz}^{n+\frac{1}{2}}(i, j, k). \end{aligned} \quad (2.8)$$

Po dosažení a vhodných úpravách je možné získat rovnici

$$Ae^{Bx} + x + C = 0, \quad (2.9)$$

kde x je hledaná hodnota E_z^{n+1} , $B = (q\Delta z)/(2kT)$, a A a C jsou hodnoty závislé na předchozí hodnotě E . Tuto rovnici je možné jednoduše řešit vhodnou numerickou metodou. Ve skriptu A.3 je implementována Newtonova-Raphsonova metoda, což je jedna z nejpoužívanějších metod pro hledání kořenů nelineární funkce[2].

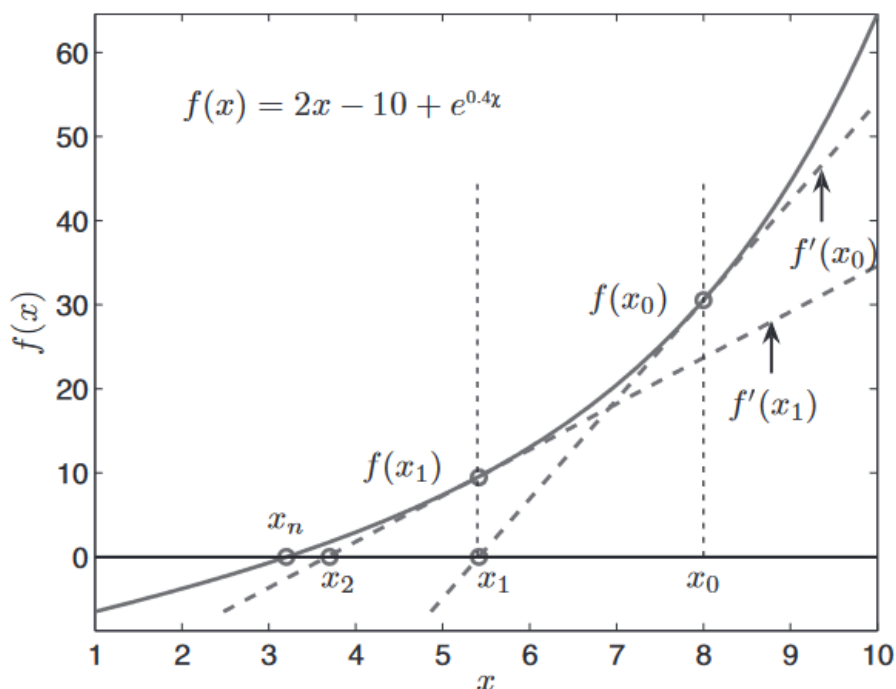
Jedná se o iterační proces, který začíná z počátečního bodu v blízkosti kořene a blíží se k němu pomocí derivace funkce v konečném počtu kroků, viz obr. 2.1. Jeho největším nebezpečím jsou lokální minima, které metoda nemusí být schopna překročit kvůli opačným hodnotám derivací v jeho okolí. V 2.9 je ale parametr A vždy kladný a rovnice je tedy součtem monotónně rostoucí¹ exponenciální funkce a monotónně rostoucí lineární funkce posunutým o konstantu C , tím pádem nemá lokální minima a metoda může být bezpečně použita.

Skript `solve_diode_equation` je volán každý časový krok v `update_diodes`, viz A.4. Tam se nejprve vypočítají proměnné C , A a spolu s hodnotou E , která byla uložena v minulém kroku, se využijí jako argumenty funkce `solve_diode_equation`, jejíž výsledek se poté vloží do $Ey(fi)$, což je místo ve výpočetním prostoru, ve kterém je dioda definována.

2.3.1 Simulace diody

Pro simulaci jednoduchého obvodu s diodou, viz definiční skripty v C, jsem vytvořil zdroj napětí produkující sinusový signál o frekvenci 500 MHz, amplitudou 1 V a vnitřním odporem 50 Ω , který je přes dvě vodivé desky spojen s diodou, viz obr. 2.2, jejíž hodnota saturačního proudu $I_0 = 5e-6$ A. Na diodě je vzorkováno napětí a proud, viz obr. 2.3. Lze vidět, že napětí na diodě odpovídá napětí na zdroji, pokud je dioda zavřená a teče v ní pouze saturační proud. Je-li dioda otevřená, je na ní úbytek napětí přibližně 0.2 V, což je hodnota odpovídající Shottkyho diodám, a proud zde nabývá maxima.

¹Rostoucí na celém definičním oboru.

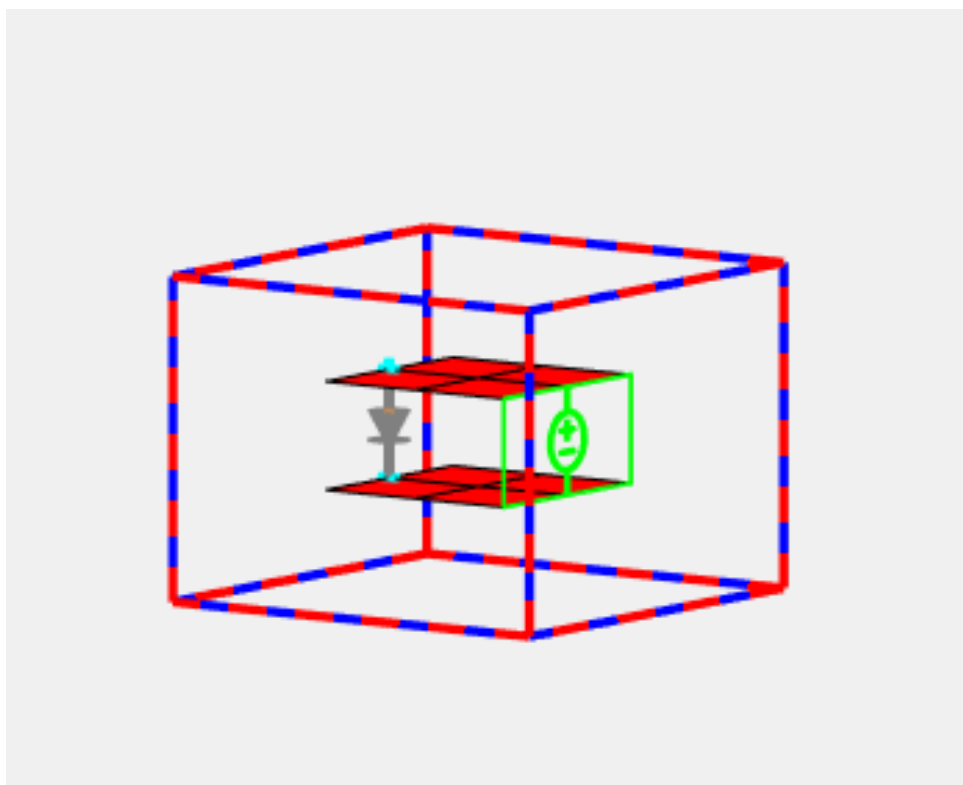


Obrázek 2.1: Ukázka použití Newtonovy - Raphsonovy metody [2]

Zatímco napětí na diodě má očekávaný průběh, na obr. 2.3b lze vidět, že ačkoliv je dioda zavřená, saturační proud je zpočátku vyšší než udává parametr I_0 a následně stoupá do kladných hodnot v době, kdy je na diodě stále závěrné napětí. Tato chyba je způsobena tím, že dioda přímo aktualizuje pouze hodnotu E , jak bylo vysvětleno v sekci 2.3, ale proud je počítán z hodnot H obklopujících diodu, které se aktualizují standardním algoritmem FDTD a nejsou tak přesně řízeny. Rozdíl mezi teoretickou a simulovanou hodnotou navíc roste při snižování paramateru I_0 . Na obr. 2.4 můžeme vidět výstupy ze stejné simulace jako v předchozím případě, pouze s parametrem $I_0 = 5e-8$ A. Napětí na diodě je při jejím otevření větší, což odpovídá teorii, protože zmenšení I_0 zmenší i proud v propustném směru, a dioda se tak o něco méně chová jako zkrat a více jako odporový prvek. Saturační proud při zavření diody ale tentokrát mnohem silněji překročí hodnotu I_0 a znovu stoupá až do kladných hodnot.

Přesnost simulace se naopak zvýší se zmenšením budící frekvence, což je potvrzeno stejnou simulací s budící frekvencí 50 MHz a původním $I_0 = 5e-6$ A, viz obr. 2.5. Menší frekvence znamená méně prudké změny, protože jsou rozloženy v tomto případě do desetkrát více časových kroků a hodnota proudu nepřekročí I_0 a ani nenabude kladných hodnot.

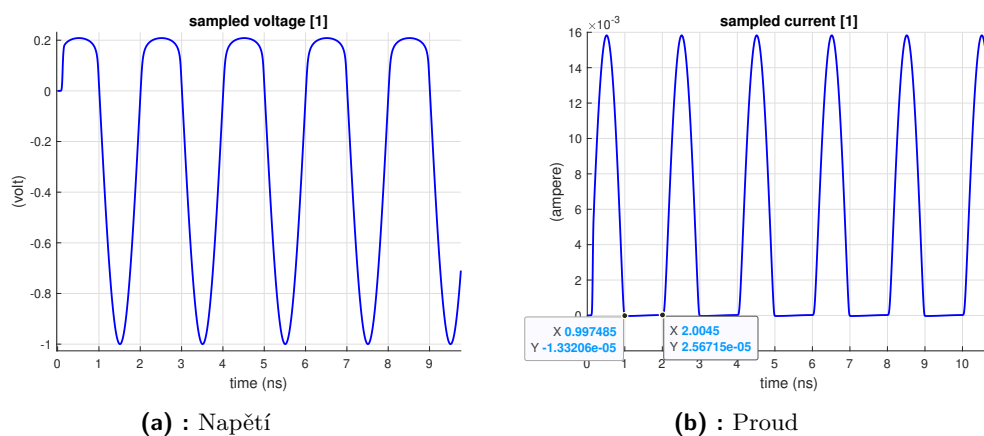
Z těchto zjištění je možné udělat závěr, že přesnost navzorkovaného proudu na diodě se zmenšuje s rostoucí frekvencí a s klesajícím saturačním proudem I_0 . Tato nepřesnost byla pro předchozí simulace v podstatě zanedbatelná, protože



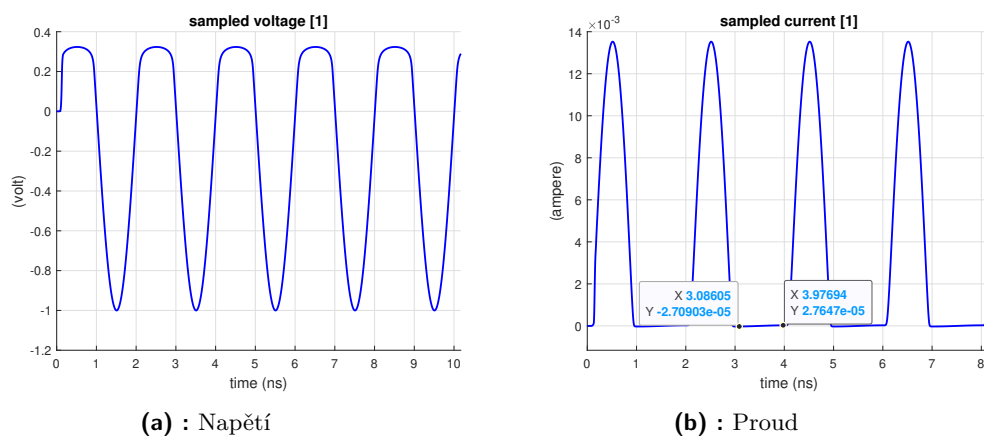
Obrázek 2.2: Simulace diody

dioda byla napájena vysokým napětím a proud v obvodu byl tak pro kladně polarizovanou diodu dostatečně velký, že v poměru na nepřesnosti v závěrné polarizaci záleží jen málo. Simulace v následujících sekcích ale budou zkoumat diody v malosignálovém režimu, konkrétně s výkonem od -40 do -5 dBm, zatímco v předchozích simulacích to bylo 0.6 dBm. To znamená, že proud v kladně polarizované diodě se zmenší natolik, že začne být srovnatelný se saturačním a rozdíl např. tří řádů mezi teoretickou a navzorkovanou hodnotou silně ovlivní časový průběh proudu, ale i frekvenční spektrum, které je klíčové k výpočtu konverzních ztrát, z angl. CL, a výpočet výkonu na diodě, který je přemětem následující sekce.

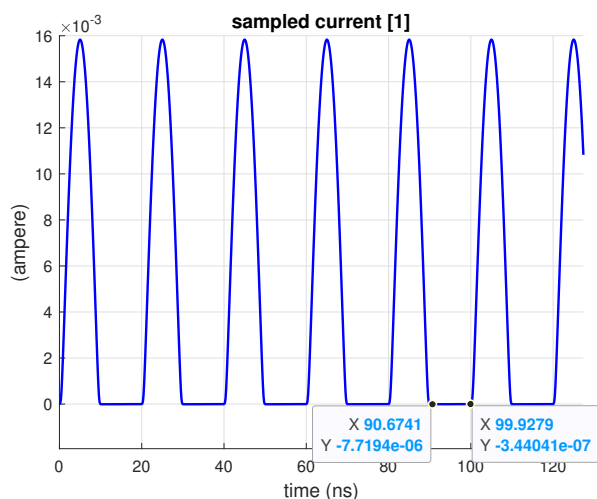
Přesnost vzorkování se naopak zvýší se zmenšením časového kroku. Toho je možné dosáhnout např. zjemněním mřížky. Δt je vázán k rozměru mřížky CFL podmínkou, viz (1.15), kde lze vidět, že mezi Δt a Δx je přímá úměrnost. Pro jednoduché geometrie v této sekci by zjemnění mřížky ovlivnilo simulaci jen málo, ale pro simulace složitějších struktur by to kvůli velikosti výpočetní oblasti znamenalo mnohonásobné prodloužení výpočetní doby. Další možnost je zjemnit pouze časový krok a velikost mřížky zachovat, to ovšem způsobí větší numerickou disperzi, viz obr. 1.10.



Obrázek 2.3: Napětí a proud vzorkované na diodě



Obrázek 2.4: Napětí a proud vzorkované na diodě s menším saturačním proudem



Obrázek 2.5: Proud na diodě buzené menší frekvencí

2.3.2 Výpočet výkonu na diodě

Chování diody je silně závislé na dodávaném výkonu, proto jsem potřeboval u každé simulace zjistit, k jakému výkonu měření přiřadit. V [2] výkon počítán není, ale jsou poskytnuty nástroje, které stačilo přizpůsobit. Pro výpočet okamžitého výkonu platí rovnice

$$p(t) = u(t) * i(t), \quad (2.10)$$

kde $u(t)$ a $i(t)$ jsou napětí a výkon v čase a výpočet obou hodnot je součástí funkcionality skriptů z [2]. Pokud je navíc třeba vypočítat výkon, kód je upraven tak, že stačí napětí a proud přiřadit struktuře `inst_powers` v `define_output_parameters`, viz B.1.

Pokud struktura `inst_powers` existuje, detekuje to skript `display_transient_parameters`, viz B.2, který mezi sebou vynásobí přiřazené hodnoty napětí a proudu, případně elektrického a magnetického pole. Protože veličiny jsou počítány a vzorkovány v různých časových krocích, je potřeba u jedné zprůměrovat vždy dvě sousední hodnoty. V mé implementaci průměruji proud, respektive magnetické pole.

Jakmile je určen okamžitý výkon v čase, můžeme z něj vyjádřit střední hodnotou výkonu rovnicí

$$\bar{f} = \frac{1}{T} \int_0^T f(t) dt, \quad (2.11)$$

úpravou pro diskrétní čas a aplikací na okamžitý výkon získáme

$$P_{\text{avg}} = \frac{1}{N} \sum_{n=1}^N p(n), \quad (2.12)$$

což je rovnice implementována v B.2. Pro pohodlí uživatele je výraz navíc v absolutní hodnotě, protože znaménko u vzorkovaných hodnot je závislé na směru, kterým se vzorkuje, a výkon tak snadno může vyjít záporný a převedení na dBm způsobí chybu. Dále je třeba pamatovat, že tato rovnice nebere v potaz budící funkci a chování diody bude odlišné, pokud stejný průměrný výkon bude generován sinusoidou, sinusoidou s jinou frekvencí, nebo např. Gaussovským pulsem, protože okamžité výkony na diodě, ze kterých se P_{avg} počítá, budou mít vysoce rozdílné průběhy.

2.3.3 Konverzní ztráty na diodě

Konverzní ztráty, v angl. conversion loss (CL), jsou rozdílem výkonových úrovní budící frekvence f_0 a vyšších harmonických $2f_0, 3f_0, \dots$. Obvykle bývá konverzní výkon na druhé harmonické frekvenci nejvyšší. V následujícím textu zkratka CL představuje konverzní ztráty mezi výkonem základní f_0 a první $2f_0$ harmonické frekvence. Pro jejich výpočet vycházím z rovnice pro časově průměrovaný Poyntingův vektor [7]

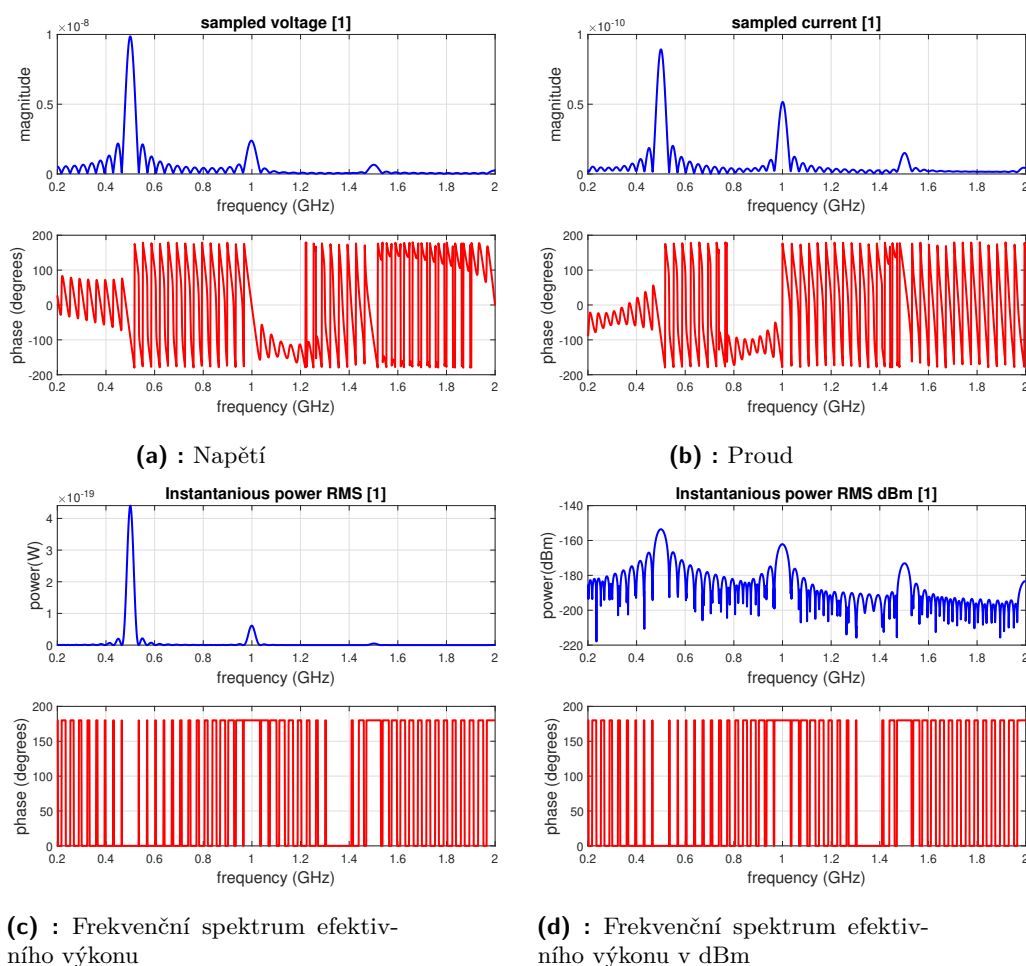
$$P_{\text{rms}} = \frac{1}{2} \text{Re}(E \cdot H^*) \quad (2.13)$$

upravenou pro napětí a proud

$$P_{\text{rms}} = \frac{1}{2} \text{Re}(U \cdot I^*) \quad (2.14)$$

kteřá se počítá ve skriptu *calculate_frequency_domain_outputs*, viz B.3, a následně se zobrazí, viz B.4. Pro snadnější odečítání jsem navíc přidal výpočet bodů na frekvenční ose, které odpovídají budící a vyšší frekvenci. Toto zjednodušení funguje pouze pro sinusové budící funkce, které mají jednu nosnou frekvenci.

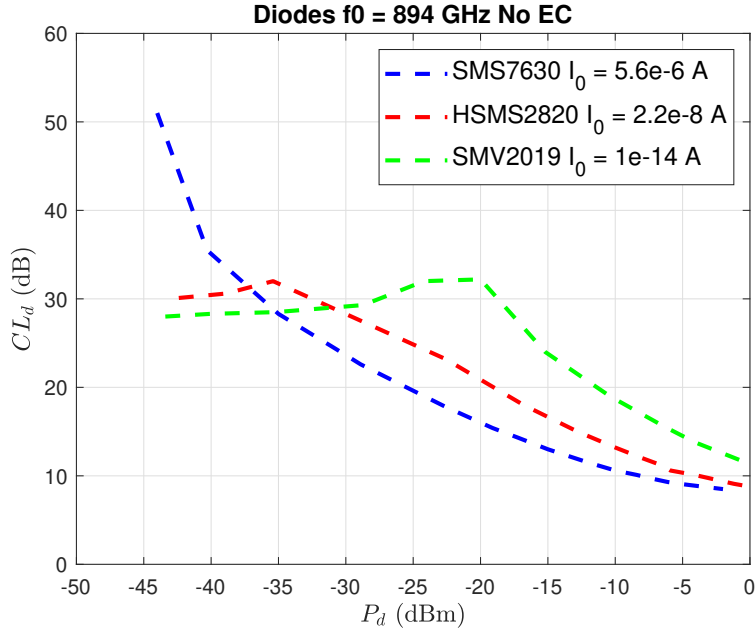
Ukázka frekvenčních spekter napětí, proudu a výkonu jsou na obr. 2.6. Znovu se jedná o obvod se zdrojem a diodou, na které obě veličiny vzorkovány.



Obrázek 2.6: Frekvenční spektra veličin na diodě

Porovnání CL různých diod je na obr. 2.7. Kromě toho obrázek ilustruje vliv saturačního proudu na přesnost zmíněnou v sekci 2.3.1. Pro diody SMV2019

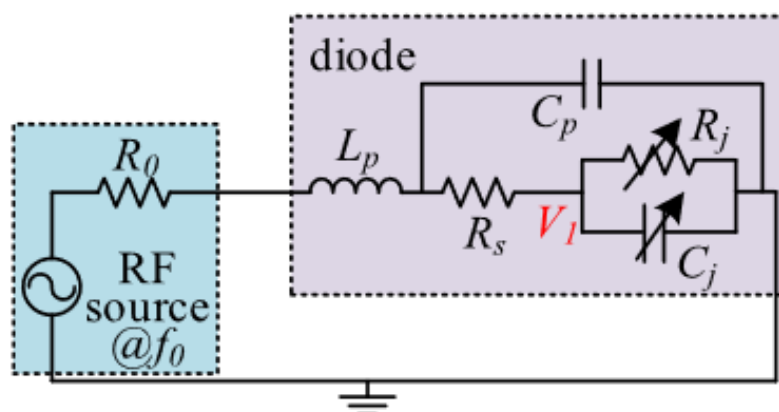
a HSMS2820 při výkonu na diodě -20 dBm, resp -35 dBm nastane zlom, po kterém už je chyba velice hrubá. Ve výkonovém spektru složka na dvojnásobné frekvenci ztrácí ostrost a rozloží se na dva laloky.



Obrázek 2.7: Srovnání konverzních ztrát na různých diodách metodou FDTD

2.3.4 Náhradní model diody

Stojí za povšimnutí, že [2] modeluje diodu jako ideální prvek, mnoho jiných metod ovšem používá SPICE parametry, kterými se simulace více přiblíží chování reálného prvku. Mezi SPICE parametry běžně používané v simulacích patří emisní koeficient n , který měří odchylku diody od rovnice ideální diody, pro P-N přechod jeho kapacita C_j a seriový odpor R_s . Pro ještě přesnější simulaci se uvažují i parazitní prvky pouzdra diody, v nejjednodušším modelu tvořené paralelní kapacitou C_p a sériovou indukčností L_p . Zatímco n je snadné vložit do exponentu v (2.5), modelování parazitních prvků je obtížnější. Jednou možností je přímé vložení do výpočetního prostoru, což zmenší přehlednost a geometrie problému to vůbec nemusí dovolit, aniž by se silně zjemnila mřížka, a tak i zvýšila výpočetní doba. Efektivnější postup by byl upravit výpočet v A.4, aby při výpočtu nové hodnoty na parazitní prvky bral ohled bez jejich přímého vložení. O to se pokouším ve skriptu B.5, kde počítám s náhradním modelem, viz obr. 2.8. Hodnota intenzity elektrického pole na diodě, která předtím byla argumentem pro skript `solve_diode_equation`, je nyní menší o úbytky na parazitních prvcích, které jsou spolu s dalšími parametry diody definované v B.6. Po dokončení funkce se k nové hodnotě úbytky znovu přidají a finální hodnota se vypíše do uzlu, na kterém je dioda definována. Rovnice vycházejí z [3] a byly převedeny z frekvenční do časové oblasti.



Obrázek 2.8: Náhradní model diody pro základní harmonickou složku f_0 dle [3]

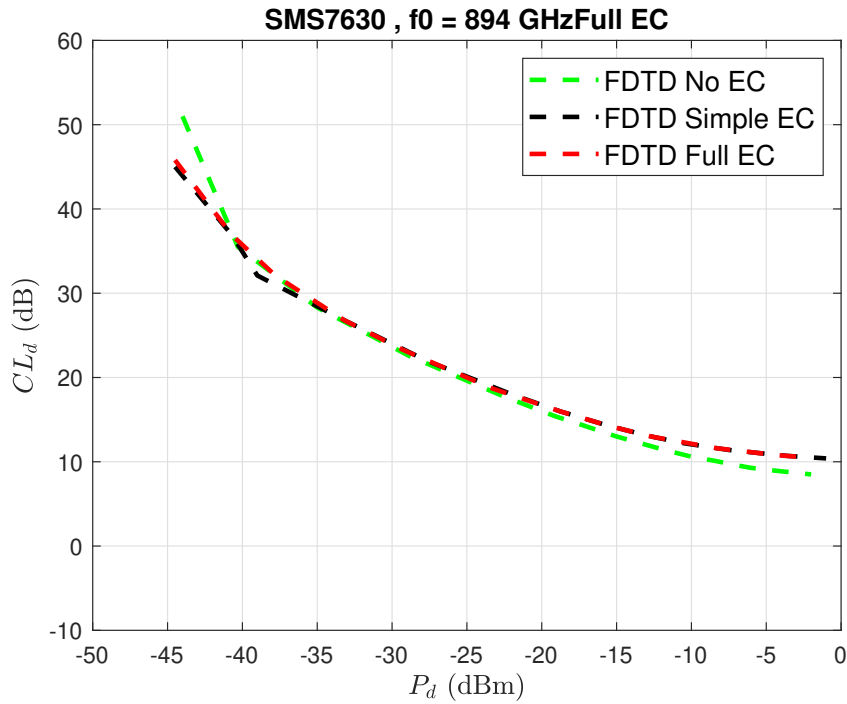
Na obr. 2.9 jsou vykreslené konverzní ztráty počítané na diodě bez parazitních prvků, s C_j a R_s , a poté s C_j , R_s , C_p a L_p . Zvolena byla dioda SMS7630, protože její simlační hodnoty vycházejí nejpřesněji. Je jasně patrné, že v mé implementaci je vliv parazitních prvků velmi malý.

2.3.5 Porovnání FDTD s metodou harmonické rovnováhy

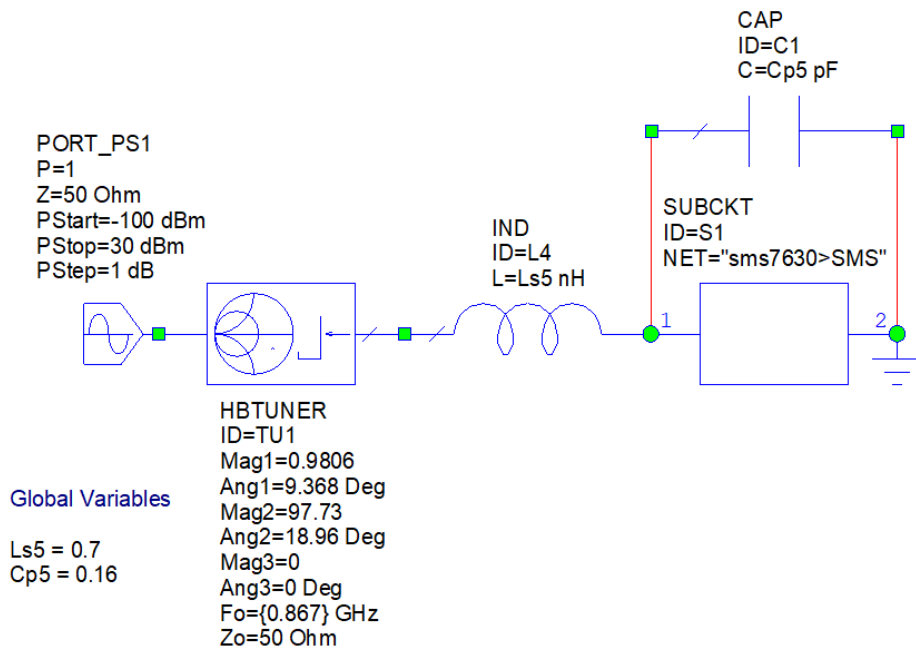
Pro ověření přesnosti výsledků je vhodné porovnat metodu FDTD s jinou, dobře zavedenou metodou. K tomu využiji simulátor AWR Microwave Office, ve kterém použiji metodu harmonické rovnováhy (Harmonic-Balance (HB)). Schéma použité pro výpočet CL je na obr. 2.10. Použil jsem prvek HBTuner, který ideálně přizpůsobuje impedanci diody k portu, aby se co nejméně výkonu odrazilo. Porovnání metod pro diodu SMS7630 je na obr. 2.11. Pro FDTD vychází konverzní ztráty vyšší a chybí minimum v rozsahu -30 až -15 dBm, které je přítomné i u HB s odstraněnými SPICE parametry, takže to nemůže být zapříčiněno špatnou implementací parazitních prvků.

2.4 Liniový dipól zatížený diodou

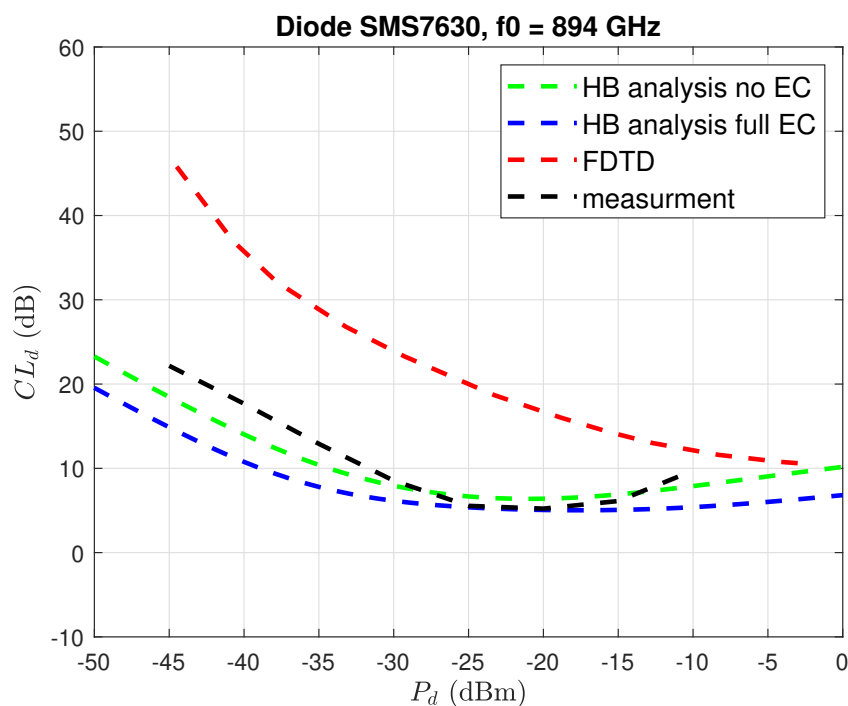
Pro simulaci této struktury jsem vycházel z [10]. Kromě dipólu a diody je navíc použita induktivní smyčka, viz obr. 2.12. Při ozáření vysokofrekvenčním signálem, o kmitočtu např. 2,9 GHz, funguje anténa jako zdroj napětí, který na diodě vytváří usměrněný proud, zatímco ladicí indukčnost L poskytuje cestu stejnosměrnému proudu a vytváří rezonanční obvod s kapacitou přechodu diody. Výsledkem je, že výstupní proud diody obsahuje harmonické složky zdrojového signálu, které jsou vyzařovány zpět přes anténu [8]. Skripty poskytnuté v [2] modelují dipól pomocí struktury *thin_wire*, která ale při použití jako přijímač nemá očekávané charakteristiky. Ponechal jsem je tedy



Obrázek 2.9: Vliv parazitních prvků na průběh konverzních ztrát diody SMS7630

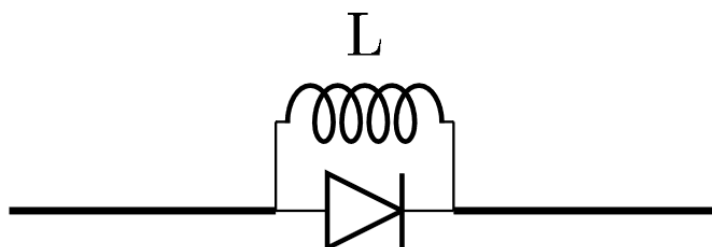


Obrázek 2.10: Schema náhradního modelu diody ideálně přizpůsobené k bráně s výkonovým rozmitáním prvkem HBTuner simulujícím optimální (komplexně sdruženou) impedanci k impedanci diody.



Obrázek 2.11: Schema porovnání konverzních ztrát pomocí metod HB a FDTD.

jen pro zjištění zářivých charakteristik, viz [D](#), a pro výpočet CL ve smyčce jsem nahradil anténu zdrojem napětí, který ji reprezentuje, viz [obr. 2.13](#).

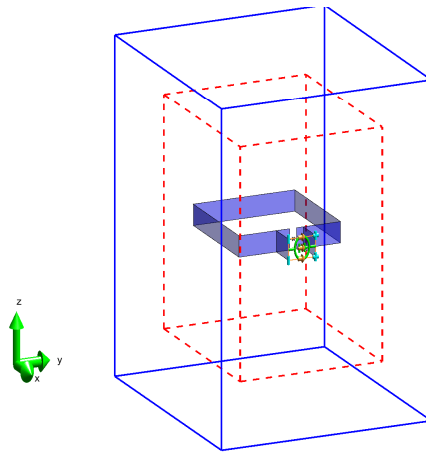


Obrázek 2.12: Obvodové schema transpondéru sestávající z dipólové antény, (Schottkyho) diody a ladicí indukčnosti L . [10]

Pro vstupní výkon $-15,1$ dBm vychází $CL = 23,4$ dB. V grafu [2.14](#) nejsou uvedené konverzní ztráty CL, ale účinnost antény η , ze které se CL snadno spočítají ze vztahu [10]

$$\eta = \eta_{pc} G(f_0) G(2f_0). \quad (2.15)$$

kde η_{pc} je konverzní účinnost a $G(nf_0)$ je zisk na n -té harmonické frekvenci. Pro výpočet zisku byl v [10] použit simulátor MMANA-GAL [11], po jehož



Obrázek 2.13: Geometrie transpondéru podle [10] bez dipólu

použití a dosazení do (2.15) vyjde po úpravě, pro vstupní výkon -15 dBm, průměr indukční smyčky 6 mm a optimální délku dipólu, $CL = 32,28$ dB. Rozdíl ve vypočítaných konverzních ztrátách tedy činí necelých 10 dBm.

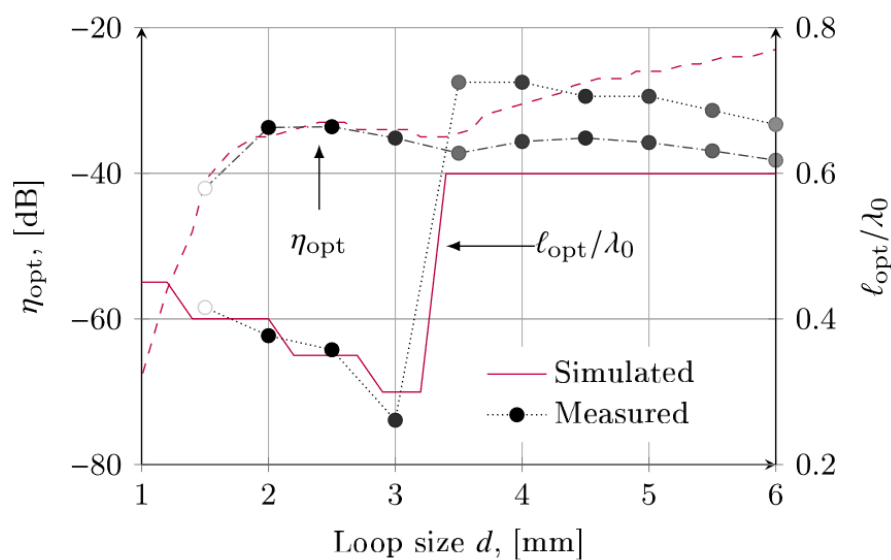
Pro výpočet HRCS potřebuji znát zisk antény, takže jsem tentokrát ponechal dipól, napájel ho zdrojem napětí a postupoval způsobem popsáním v sekci 2.2. Zisk na základní a dvojnásobné frekvenci vyšel $0,087$ dBi a $2,65$ dBi, což je méně než hodnoty $2,39$ a $4,89$ ze simulátoru MMANA-GAL. Po dosazení hodnot vyjde $HRCS = 7.4436e-6 m^2$.

2.5 Flíčková anténa zatížená diodou

Tento harmonický transpondér s dvojitou symetrií, který se skládá z dvoupásmové flíčkové antény zatížené Schottkyho diodou HSMS2820, byl představen v [12]. Geometrie transpondéru je na obr. 2.15, jeho definiční skripty v E. Jedná se o největší a nejsložitější strukturu v této práci, a pro analýzu jejích zářivých vlastností využívám stejného zjednodušení jako v minulém případě. Místo diody HSMS2820, jejíž CL už byly počítány v předchozí sekci, je zapojen zdroj napětí, který budí anténu na základní a dvojnásobné frekvenci.

Veličiny analyzované pomocí HB/full-wave metody a jejich srovnání s metodou FDTD je v tabulce 2.1. Rozdíl v účinnosti a zisku, který je z ní přímo počítán, znamená, že vyzářený výkon je oproti výkonu na zdroji mnohem menší, než by měl být. Je tedy možné, že na přechodu mezi vodičem a volným prostorem dochází k přílišnému utlumení. To naznačuje i rozdíl ve směrovosti, jejíž hodnota je počítána přímo v původních skriptech z [2].

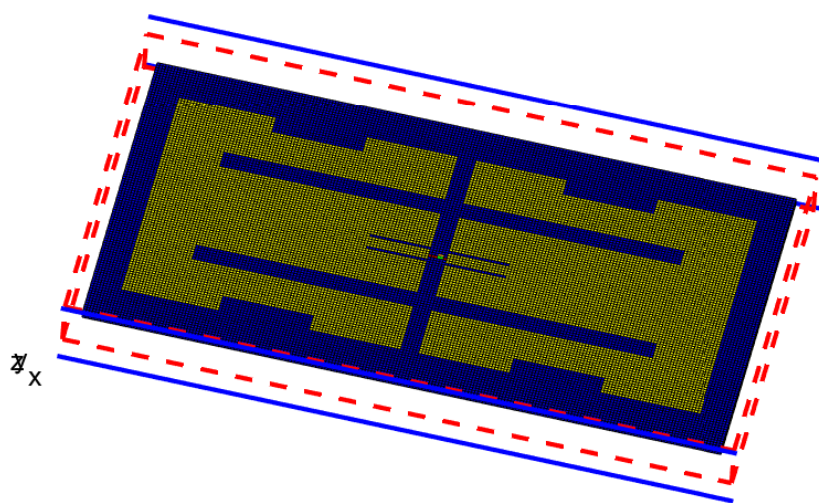
Z hodnot v tabulce a znalosti CL diody HSMS2820 ze sekce 2.3.3 lze pro $P_{in} = -15$ dBm spočítat $HRCS = 3.5212e-4 m^2$



Obrázek 2.14: Naměřená a simulovaná účinnost transpondéru při $f_0/2f_0 = 2,9/5,8$ GHz s $P_{in} = -15$ dBm: nejlepší účinnost transpondéru η_{opt} (levá osa y) a odpovídající délka dipólu l_{opt}/λ_0 (pravá osa y) [10]

Tabulka 2.1: Srovnání simulované směrovosti, vyzařovací účinnosti a zisku ve volném prostoru HB/Full wave metodou [12] a FDTD

| | HB/full-wave | | FDTD | |
|-------------------------|--------------|--------|-------|--------|
| | f_0 | $2f_0$ | f_0 | $2f_0$ |
| Směrovost (dBi) | 3,2 | 6,5 | 1.86 | 5.2 |
| Vyzařovací účinnost (%) | 69 | 83 | 43 | 55 |
| Zisk (dBi) | 1,6 | 8,7 | 0.8 | 2.9 |



Obrázek 2.15: Geometrie transpondéru s flíčkovou anténou podle [12]

Závěr

V této práci byla představena metoda FDTD a její aplikace na problematiku harmonických transpondérů zatížených nelineárním prvkem, diodou. Cílem bylo zjistit, jak metoda obstojí v porovnání s jinými dostupnými metodami a jak budou numerické parametry odpovídat hodnotám získaných z měření.

Nejprve jsem podrobně zkoumal chování několika různě složitých modelů samotné diody a analyzoval jejich konverzní ztráty. Zjistil jsem, že přesnost simulace je závislá kromě časového a prostorového kroku i na budící frekvenci a saturačním proudu diody. V porovnání s metodou harmonické rovnováhy a měřením tvar křivek nebyl příliš odlišný, ale metodou FDTD vyšly konverzní ztráty pro výkon na diodě $P_d = -10$ dBm o cca 5 až 7 dB vyšší, avšak sklesajícím výkonem rozdíl roste až na cca 20 dB pro $P_d = -40$ dB. Měřené hodnoty konverzních ztrát jsou hodnotám získaných FDTD simulací trochu blíže, ale metoda HB vychází mnohem lépe.

V další části jsem simuloval zářivé vlastnosti transpondérů. U liniového dipólu s indukční smyčkou jsem získal konverzní ztráty tentokrát přibližně o 10 dB menší, než vyšlo metodou harmonické rovnováhy, která pro můj výběr průměru smyčky blízce odpovídala naměřené hodnotě. U flíčkové antény jsem vyhodnocoval směrovost, vyzařovací účinnost a zisk. Hodnoty se velmi lišili od vzorových. Simulace nejspíš byla zatížena chybou, kterou jsem zatím přesně neidentifikoval, ale nejspíš je spojena s množstvím buňek mezi strukturou a hranicí výpočetního prostoru, protože při opakované simulaci s větší mezerou se výsledky simulace těm vzorovým přiblížily.

Použitá implementace metody FDTD pro analýzu vlastností diody má jistě potenciál pro zlepšení, neboť v mém provedení stále existuje několik nedostatků, které negativně ovlivňují přesnost výsledků. Jedním z nich je nepřesné modelování SPICE parametrů diody. Způsob, který jsem implementoval, ovlivňuje výsledky zanedbatelně a určitě není dostačující. Dále je třeba identifikovat chybu ve struktuře *thin_wire*, která nefunguje správně při použití jako přijmač a obecně ověřit správnost přechodu polí mezi vodiči a volným prostorem.



Literatura

- [1] GARG, Ramesh. *Analytical and Computational Methods in Electromagnetics*. Artech House, 2008. ISBN 978-1-59693-685-9.
- [2] ELSHERBENI, Atef Z. a VEYSEL Demir. *The Finite-Difference Time-Domain Method for Electromagnetics with MATLAB R© Simulations*. Scitech Publishing, 2nd ed., 2015. ISBN 978-1-891121-71-5.
- [3] GU, Xiaoqiang, SRINAGA, Nikhil N., HEMOUR, Simon a WU, Ke. *Diplexer-Based Fully Passive Harmonic Transponder for Sub-6-GHz 5G-Compatible IoT Applications*. in IEEE Transactions on Microwave Theory and Techniques, vol. 67, no. 5, pp. 1675-1687, May 2019
- [4] MAAS, Stephen A. *Nonlinear microwave and RF circuits*. Artech House, 2003 ISBN 978-1-58053-484-0
- [5] TAFLOVE, Allen a HAGNESS, Susan C. *Computational Electrodynamics: The Finite Difference Time Domain Method*. 3rd ed. Norwood, MA: Artech House Publishers, 2005, ISBN 978-1580530767
- [6] PETERSON, Andrew F., SCOTT, Ray L. and MITTRA, Raj. *Computational Methods for Electromagnetics*. Hyderabad: University Press, 2001, Chapter 12
- [7] BALANIS, Constantine A. *Antenna Theory Analysis and Design*. 2nd Edition, John Wiley & Sons, Inc., New York, 1997. ISBN 078471592686
- [8] STORTZ, Greg a LAVRENKO, Anastasia. *Compact Low-cost FMCW Harmonic Radar for Short Range Insect Tracking*. 2020 IEEE International Radar Conference (RADAR), Washington, DC, USA, 2020, pp. 642-647
- [9] LUDWIG, Reinhold a BOGDANOV, Gene. *RF Circuit Design: Theory and Applications*. Pearson international edition. Ilustrované vydání. Prentice Hall, 2009. ISBN 978-0131471375
- [10] LAVRENKO, Anastasia, LITCHFIELD, Benjamin, WOODWARD, Greame a PAWSON, Stephen. *Design and Evaluation of a Compact Harmonic Transponder for Insect Tracking*. in IEEE Microwave and Wireless Components Letters, vol. 30, no. 4, pp. 445-448, April 2020

- [11] SCHEWELEW, Alexander. MMANA-GAL: Antenna-Analyzing Tool Based on the Moment Method. Dostupné : <http://gal-ana.de/basicmm/en/>
- [12] POLÍVKA, Milan, HUBATA-VACEK, Václav a ŠVANDA, Milan. *Harmonic Balance/Full-Wave Analysis of Wearable Harmonic Transponder for IoT Applications*. in IEEE Transactions on Antennas and Propagation, vol. 70, no. 2, pp. 977-987, Feb. 2022.



Slovník

- CEM** Computational Electromagnetism. 1
- CFL** Courant-Friedrich-Levy. 7, 8, 23
- CL** Conversion Loss. 20, 23, 25, 26, 28, 30, 31
- DFT** diskrétní Furierova transformace. 13
- FD** Finite-Difference. 3, 9
- FDTD** Finite-Difference Time-Domain. vi, viii, ix, 1, 3, 5–8, 10, 14, 15, 19, 20, 31, 32
- FF** Far-Field. 15, 16
- FFT** rychlá Furierova transformace. 13
- HB** Harmonic-Balance. 28
- HRCS** Harmonic Radar Cross Section. 19, 31
- NF** Near-Field. 15
- PEC** Perfect Electric Conductor. 11
- PML** Perfectly Matched Layer. 11

Příloha A

Vybrané skripty

```
% define the problem
define_problem_space_parameters;
define_geometry;
define_sources_and_lumped_elements;
define_output_parameters;

% initialize the problem space and parameters
initialize_fDTD_material_grid;
display_problem_space;
display_material_mesh;
if run_simulation
    initialize_fDTD_parameters_and_arrays;
    initialize_sources_and_lumped_elements;
    initialize_updating_coefficients;
    initialize_boundary_conditions;
    initialize_output_parameters;
    initialize_farfield_arrays;
    initialize_display_parameters;

    % FDTD time marching loop
    run_fDTD_time_marching_loop;

    % display simulation results
    post_process_and_display_results;
end
```

Kód A.1: Hlavní implementační skript

```
disp(['Starting the time marching loop']);
disp(['Total number of time steps : ' ...
      num2str(number_of_time_steps)]);

start_time = cputime;
current_time = 0;
```

```

for time_step = 1:number_of_time_steps
    update_incident_fields;
    update_magnetic_fields;
    update_magnetic_field_CPML_ABC;
    capture_sampled_magnetic_fields;
    capture_sampled_currents;
    update_electric_fields;
    update_electric_field_CPML_ABC;
    update_voltage_sources;
    update_current_sources;
    update_inductors;
    update_diodes;
    capture_sampled_electric_fields;
    capture_sampled_voltages;
    calculate_JandM;
    display_sampled_parameters;
end

end_time = cputime;
total_time_in_minutes = (end_time - start_time)/60;
disp(['Total simulation time is ' ...
      num2str(total_time_in_minutes) ' minutes.']);

```

Kód A.2: Implementace FDTD algoritmu

```

function [x] = solve_diode_equation(A, B, C, x)
% Function used to solve the diode equation
% which is in the form  $Ae^{Bx} + x + C = 0$ 
% using the Newton-Raphson method

tolerance = 1e-25;
max_iter = 50;
iter = 0;
f = A * exp(B*x) + x + C;
while ((iter < max_iter) && (abs(f) > tolerance))
    fp = A * B * exp(B*x) + 1;
    x = x - f/fp;
    f = A * exp(B*x) + x + C;
    iter = iter + 1;
end

```

Kód A.3: *solve_diode_equation*

```

% updating electric field components
% associated with the diodes

for ind = 1:number_of_diodes
    fi = diodes(ind).field_indices;

```

```
B = diodes(ind).B;
switch (diodes(ind).direction(1))
case 'x'
    E = diodes(ind).Exn;
    C = -Ex(fi) + diodes(ind).Cexd;
    A = -diodes(ind).Cexd * exp(B * E);
    E = solve_diode_equation(A, B, C, E);
    Ex(fi) = E;
    diodes(ind).Exn = E;
case 'y'
    E = diodes(ind).Eyn;
    C = -Ey(fi) + diodes(ind).Ceyd;
    A = -diodes(ind).Ceyd * exp(B * E);
    E = solve_diode_equation(A, B, C, E);
    Ey(fi) = E;
    diodes(ind).Eyn = E;
case 'z'
    E = diodes(ind).Ezn;
    C = -Ez(fi) + diodes(ind).Cezd;
    A = -diodes(ind).Cezd * exp(B * E);
    E = solve_diode_equation(A, B, C, E);
    Ez(fi) = E;
    diodes(ind).Ezn = E;
end
end
```

Kód A.4: *update_diodes*

Příloha B

Obecné úpravy skriptů

```
% define sampled voltages
sampled_voltages(1).min_x = 2.0e-3;
sampled_voltages(1).min_y = 1.0e-3;
sampled_voltages(1).min_z = 0;
sampled_voltages(1).max_x = 2.0e-3;
sampled_voltages(1).max_y = 1.0e-3;
sampled_voltages(1).max_z = 1.0e-3;
sampled_voltages(1).direction = 'zp';
sampled_voltages(1).display_plot = false;

% define sampled currents
sampled_currents(1).min_x = 2e-3;
sampled_currents(1).min_y = 1e-3;
sampled_currents(1).min_z = 0e-3;
sampled_currents(1).max_x = 2e-3;
sampled_currents(1).max_y = 1e-3;
sampled_currents(1).max_z = 1e-3;
sampled_currents(1).direction = 'zn';
sampled_currents(1).display_plot = false;

%power
inst_powers(1).sampled_voltage_index = 1;
inst_powers(1).sampled_current_index = 1;
```

Kód B.1: Přidání výpočtu okamžitého výkonu v *define_output_parameters*

```
for ind=1:number_of_inst_powers
    figure;
    xlabel('time (ns)', 'fontsize', 12);
    ylabel('(watts)', 'fontsize', 12);
    title(['Instantaneous power [' num2str(ind) ']', '
    fontsize', 12);
    grid on; hold on;
    sampled_time = time(1:time_step)*1e9;
    if isfield(inst_powers(ind), 'sampled_voltage_index')
```

```

        voltage = sampled_voltages(inst_powers(ind).
sampled_voltage_index).sampled_value(1:time_step);
        current = zeros(1, number_of_time_steps);
        for i=1:number_of_time_steps -1
            current(i) = (sampled_currents(inst_powers(ind).
sampled_current_index).sampled_value(i) ...
                + sampled_currents(inst_powers(ind).
sampled_current_index).sampled_value(i+1)) / 2;
        end
        sampled_value = voltage .* current;
    end

    if isfield(inst_powers(ind), 'sampled_electric_fields')
        electric_f = sampled_voltages(inst_powers(ind).
sampled_voltage_index).sampled_value(1:time_step);
        magnetic_f = zeros(1, number_of_time_steps);
        for i=1:number_of_time_steps -1
            magnetic_f(i) = (sampled_magnetic_fields(
inst_powers(ind).sampled_magnetic_fields).sampled_value(i)
            ...
                + sampled_magnetic_fields(inst_powers(ind).
sampled_magnetic_fields).sampled_value(i+1)) / 2;
        end
        sampled_value = voltage .* current;
    end

    plot(sampled_time, sampled_value, 'b-', 'linewidth', 1.5);
    drawnow;
    inst_powers(ind).sampled_values = sampled_value;

    % average power
    average_power = abs(sum(inst_powers(ind).sampled_values)
/ number_of_time_step);
    average_power_dBm = 10 * log10(average_power / 0.001);
    disp(['Average power(' num2str(ind) ') = ' num2str(
average_power) ' W']);
    disp(['Average power(' num2str(ind) ') = ' num2str(
average_power_dBm) ' dBm']);
end

```

Kód B.2: Výpočet a zobrazení okamžitého výkonu v časové oblasti v *display_transient_parameters*

```

% Instantaneous power
for ind=1:number_of_inst_powers
    if isfield(inst_powers(ind), 'sampled_voltage_index')
        inst_powers(ind).frequency_domain_value_RMS = 1/2 *

```



```

real(sampled_voltages(inst_powers.sampled_voltage_index).
frequency_domain_value...
.*
conj(sampled_currents(inst_powers.sampled_current_index).
frequency_domain_value));
end

if isfield(inst_powers(ind), 'sampled_electric_fields')
    inst_powers(ind).frequency_domain_value_RMS = 1/2 *
real(sampled_electric_fields(inst_powers.
sampled_electric_fields).frequency_domain_value...
.*
conj(sampled_magnetic_fields(inst_powers.
sampled_magnetic_fields).frequency_domain_value));
end
inst_powers(ind).frequencies = frequency_array;
end

```

Kód B.3: Výpočet efektivní hodnoty výkonu ve frekvenční oblasti v *calculate_frequency_domain_outputs*

```

% figures for RMS power
for ind=1:number_of_inst_powers
    frequencies = inst_powers(ind).frequencies*1e-9;
    fd_value = inst_powers(ind).frequency_domain_value_RMS;
    figure;
    subplot(2,1,1);
    plot(frequencies, abs(fd_value), 'b-', 'linewidth', 1.5);
    title(['Instantaneous power RMS [' num2str(ind) ']' ], '
    fontsize', 12);
    xlabel('frequency (GHz)', 'fontsize', 12);
    ylabel('power (W)', 'fontsize', 12);
    grid on;
    subplot(2,1,2);
    plot(frequencies, angle(fd_value)*180/pi, 'r-', 'linewidth'
    , 1.5);
    xlabel('frequency (GHz)', 'fontsize', 12);
    ylabel('phase (degrees)', 'fontsize', 12);
    grid on;
    drawnow;

    fd_value_dBm = 10 * log10(abs(fd_value) * 1000);

    figure;
    subplot(2,1,1);
    plot(frequencies, fd_value_dBm, 'b-', 'linewidth', 1.5);
    title(['Instantaneous power RMS dBm [' num2str(ind) ']' ], '

```

```

    fontsize',12);
    xlabel('frequency (GHz)', 'fontsize',12);
    ylabel('power(dBm)', 'fontsize',12);
    grid on;
    subplot(2,1,2);
    plot(frequencies, angle(fd_value)*180/pi, 'r-', 'linewidth',
    ,1.5);
    xlabel('frequency (GHz)', 'fontsize',12);
    ylabel('phase (degrees)', 'fontsize',12);
    grid on;
    drawnow;

    sampling_point_f0 = (waveforms.sinusoidal(1).frequency -
    frequency_domain.start) / (frequency_domain.step);
    sampling_point_2f0 = (2* waveforms.sinusoidal(1).
    frequency - frequency_domain.start) / (frequency_domain.
    step);

    disp(['na f0: ' num2str(fd_value_dBm(1,sampling_point_f0
    +1))])
    disp(['na 2f0: ' num2str(fd_value_dBm(1,
    sampling_point_2f0+1))])
    disp(['CL: ' num2str(fd_value_dBm(1,sampling_point_f0-1) -
    fd_value_dBm(1,sampling_point_2f0-1))])
end

```

Kód B.4: Zobrazení výkonového spektra v *display_frequency_domain_outputs*

```

% updating electric field components
% associated with the diodes
for ind = 1:number_of_diodes
    fi = diodes(ind).field_indices;
    B = diodes(ind).B;

    ni = diodes(ind).node_indices;
    is = ni.is; js = ni.js; ks = ni.ks;
    ie = ni.ie; je = ni.je; ke = ni.ke;

    switch (diodes(ind).direction(1))
    case 'x'

        I_tot = ...
        + dy * sum(sum(sum(Hy(is:ie-1,js:je,ks-1)))) ...
        + dz * sum(sum(sum(Hz(is:ie-1,je,ks:ke))))...
        - dy * sum(sum(sum(Hy(is:ie-1,js:je,ke))))...
        - dz * sum(sum(sum(Hz(is:ie-1,js-1,ks:ke)))));
    end
end

```

```

    I_Rj = (I_tot - diodes(ind).Exn*dx*Cj - (diodes(ind).
Exn*dx*Cj*Rs + diodes(ind).Exn*dx)*Cp) / (1+Rs*Cp);

    E = diodes(ind).Exn;

    E_diode = (E - I_tot*Lp/dx - I_Rj*Rs/dx) / (1+1*Cj*Rs
); %part of E that actually makes it to the junction and
is used in solve_diode_equation

    C = -Ex(fi) + diodes(ind).Cexd;
    A = -diodes(ind).Cexd * exp(B * E_diode);
    E_diode = solve_diode_equation(A, B, C, E_diode);

    E_out = E_diode + (I_Rj + E_diode*Cj)*Rs/dx + I_tot*
Lp/dx;

    Ex(fi) = E_out;
    diodes(ind).Exn = E_out;           %ulozeni pro dalsi
cyklus

```

Kód B.5: Výpočet nové hodnoty v *update_diodes* s ohledem na parazitní prvky

```

disp('initializing diode updating coefficients');

q = 1.602*1e-19; % charge of an electron
k = 1.38066e-23; % Boltzman constant, joule/kelvin
T = 273+27;      % Kelvin; room temperature
I_0 = 5e-6;      % saturation current

%SMS-7630
n = 1.05;        %ideality factor
Rs = 20;
Cj = 0.14e-12;
Cp = 0.16e-12;
Lp = 0.7e-9;

```

Kód B.6: Inicializace parazitních prvků v *initialize_diode Updating_coefficients*

Příloha C

Definiční skripty pro simulace diod

```
disp('defining the problem geometry');

bricks = [];
spheres = [];

bricks(1).min_x = 0;
bricks(1).min_y = 0;
bricks(1).min_z = 0;
bricks(1).max_x = 2e-3;
bricks(1).max_y = 2e-3;
bricks(1).max_z = 0;
bricks(1).material_type = 2;

bricks(2).min_x = 0;
bricks(2).min_y = 0;
bricks(2).min_z = 1e-3;
bricks(2).max_x = 2e-3;
bricks(2).max_y = 2e-3;
bricks(2).max_z = 1e-3;
bricks(2).material_type = 2;
```

Kód C.1: *define_geometry*

```
disp('defining the problem space parameters');

% maximum number of time steps to run FDTD simulation
number_of_time_steps = 665*20+7;%*100;%646 *10 ;%577 * 20;

% A factor that determines duration of a time step
% wrt CFL limit
courant_factor = 0.9;

% A factor determining the accuracy limit of FDTD results
number_of_cells_per_wavelength = 20;
```

```
% Dimensions of a unit cell in x, y, and z directions (meters
)
dx=1.0e-3;
dy=1.0e-3;
dz=1.0e-3;

% ==<boundary conditions>=====
% Here we define the boundary conditions parameters
% 'pec' : perfect electric conductor
boundary.type_xp = 'pec';
boundary.air_buffer_number_of_cells_xp = 2;

boundary.type_xn = 'pec';
boundary.air_buffer_number_of_cells_xn = 2;

boundary.type_yp = 'pec';
boundary.air_buffer_number_of_cells_yp = 2;

boundary.type_yn = 'pec';
boundary.air_buffer_number_of_cells_yn = 2;

boundary.type_zp = 'pec';
boundary.air_buffer_number_of_cells_zp = 2;

boundary.type_zn = 'pec';
boundary.air_buffer_number_of_cells_zn = 2;

% ===<material types>=====
% Here we define and initialize the arrays of material types
% eps_r   : relative permittivity
% mu_r    : relative permeability
% sigma_e : electric conductivity
% sigma_m : magnetic conductivity

% air
material_types(1).eps_r = 1;
material_types(1).mu_r = 1;
material_types(1).sigma_e = 0;
material_types(1).sigma_m = 0;
material_types(1).color = [1 1 1];

% PEC : perfect electric conductor
material_types(2).eps_r = 1;
material_types(2).mu_r = 1;
material_types(2).sigma_e = 1e10;
material_types(2).sigma_m = 0;
```

```

material_types(2).color = [1 0 0];

% PMC : perfect magnetic conductor
material_types(3).eps_r = 1;
material_types(3).mu_r = 1;
material_types(3).sigma_e = 0;
material_types(3).sigma_m = 1e10;
material_types(3).color = [0 1 0];

% index of material types defining air, PEC, and PMC
material_type_index_air = 1;
material_type_index_pec = 2;
material_type_index_pmc = 3;

```

Kód C.2: *define_problem_space_parameters*

```

disp('defining sources and lumped element components');

voltage_sources = [];
current_sources = [];
diodes = [];
resistors = [];
inductors = [];
capacitors = [];

% define source waveform types and parameters
waveforms.sinusoidal(1).frequency = 0.867e9;

% voltage sources
% direction: 'xp', 'xn', 'yp', 'yn', 'zp', or 'zn'
% resistance : ohms, magititude : volts
voltage_sources(1).min_x = 0;
voltage_sources(1).min_y = 0;
voltage_sources(1).min_z = 0;
voltage_sources(1).max_x = 0.0e-3;
voltage_sources(1).max_y = 2.0e-3;
voltage_sources(1).max_z = 1.0e-3;
voltage_sources(1).direction = 'zp';
voltage_sources(1).resistance = 50;
voltage_sources(1).magnitude = 0.1;
voltage_sources(1).waveform_type = 'sinusoidal';
voltage_sources(1).waveform_index = 1;

% diodes
% direction: 'xp', 'xn', 'yp', 'yn', 'zp', or 'zn'
diodes(1).min_x = 2.0e-3;
diodes(1).min_y = 1.0e-3;

```

```
diodes(1).min_z = 0.0e-3;  
diodes(1).max_x = 2.0e-3;  
diodes(1).max_y = 1.0e-3;  
diodes(1).max_z = 1.0e-3;  
diodes(1).direction = 'zn';
```

Kód C.3: *define_sources_and_lumped_elements*

```
disp('defining output parameters');  
  
sampled_electric_fields = [];  
sampled_magnetic_fields = [];  
sampled_voltages = [];  
sampled_currents = [];  
  
% figure refresh rate  
plotting_step = 10;  
  
% mode of operation  
run_simulation = true;  
show_material_mesh = false;  
show_problem_space = true;  
  
% frequency domain parameters  
frequency_domain.start = 0.2e9;  
frequency_domain.end = 2e9;  
frequency_domain.step = 10e5;  
  
% define sampled voltages  
sampled_voltages(1).min_x = 2.0e-3;  
sampled_voltages(1).min_y = 1.0e-3;  
sampled_voltages(1).min_z = 0;  
sampled_voltages(1).max_x = 2.0e-3;  
sampled_voltages(1).max_y = 1.0e-3;  
sampled_voltages(1).max_z = 1.0e-3;  
sampled_voltages(1).direction = 'zp';  
sampled_voltages(1).display_plot = false;  
  
% define sampled currents  
sampled_currents(1).min_x = 2e-3;  
sampled_currents(1).min_y = 1e-3;  
sampled_currents(1).min_z = 0e-3;  
sampled_currents(1).max_x = 2e-3;  
sampled_currents(1).max_y = 1e-3;  
sampled_currents(1).max_z = 1e-3;  
sampled_currents(1).direction = 'zn';  
sampled_currents(1).display_plot = false;
```



```
sampled_currents(1).diode_ind = 1;  
  
%power  
inst_powers(1).sampled_voltage_index = 1;  
inst_powers(1).sampled_current_index = 1;
```

Kód C.4: *define_output_parameters*

Příloha D

Definiční skripty pro simulaci liniového dipólu zatíženého diodou

```
disp('defining the problem geometry');

bricks = [];
spheres = [];
thin_wires = [];

%dipole up
thin_wires(1).min_x = 1.6e-3;
thin_wires(1).min_y = 0.8e-3;
thin_wires(1).min_z = 0;
thin_wires(1).max_x = 1.6e-3;
thin_wires(1).max_y = 0.8e-3+0.031;
thin_wires(1).max_z = 0;
thin_wires(1).radius = 0.08e-3;
thin_wires(1).direction = 'y';

%dipole down
thin_wires(2).min_x = 1.6e-3;
thin_wires(2).min_y = -0.8e-3-0.0310;
thin_wires(2).min_z = 0;
thin_wires(2).max_x = 1.6e-3;
thin_wires(2).max_y = -0.8e-3;
thin_wires(2).max_z = 0;
thin_wires(2).radius = 0.08e-3;
thin_wires(2).direction = 'y';

% define a brick - PCB
% bricks(1).min_x = -7.5e-3;
% bricks(1).min_y = -7.5e-3;
% bricks(1).min_z = 2*0.8e-3;
% bricks(1).max_x = 7.5e-3;
% bricks(1).max_y = 7.5e-3;
% bricks(1).max_z = 0e-3;
```

```
% bricks(1).material_type = 4;

% loop
% bricks(1).min_x = -6e-3;
% bricks(1).min_y = -3e-3;
% bricks(1).min_z = 0;
% bricks(1).max_x = -6e-3;
% bricks(1).max_y = 3e-3;
% bricks(1).max_z = 0.8e-3;
% bricks(1).material_type = 4;

% loop 1a
% bricks(2).min_x = 0;
% bricks(2).min_y = 0.4e-3;
% bricks(2).min_z = 0;
% bricks(2).max_x = 0;
% bricks(2).max_y = 3e-3; %0.8e-3+2.2e-3;
% bricks(2).max_z = 0.8e-3;
% bricks(2).material_type = 4;

% % loop 1b
% bricks(3).min_x = 0;
% bricks(3).min_y = -3e-3; %-0.8e-3-2.2e-3;
% bricks(3).min_z = 0;
% bricks(3).max_x = 0;
% bricks(3).max_y = -0e-3;
% bricks(3).max_z = 0.8e-3;
% bricks(3).material_type = 4;

% %loop 2a
% bricks(4).min_x = -6e-3;
% bricks(4).min_y = 3e-3;
% bricks(4).min_z = 0;
% bricks(4).max_x = 0;
% bricks(4).max_y = 3e-3;
% bricks(4).max_z = 0.8e-3;
% bricks(4).material_type = 4;

% %loop 2b
% bricks(5).min_x = -6e-3;
% bricks(5).min_y = -3e-3;
% bricks(5).min_z = 0;
% bricks(5).max_x = 0;
% bricks(5).max_y = -3e-3;
% bricks(5).max_z = 0.8e-3;
% bricks(5).material_type = 4;
```

```

% diode - dipole
% bricks(6).min_x = 0;
% bricks(6).min_y = 0.8e-3;
% bricks(6).min_z = 0;
% bricks(6).max_x = 1.6e-3;
% bricks(6).max_y = 0.8e-3;
% bricks(6).max_z = 0.8e-3;
% bricks(6).material_type = 4;

% diode - dipole
% bricks(7).min_x = 0;
% bricks(7).min_y = -0.8e-3;
% bricks(7).min_z = 0;
% bricks(7).max_x = 1.6e-3;
% bricks(7).max_y = -0.8e-3;
% bricks(7).max_z = 0.8e-3;
% bricks(7).material_type = 4;

```

Kód D.1: *define_geomtery*

```

disp('defining the problem space parameters');

% maximum number of time steps to run FDTD simulation
number_of_time_steps = floor(497*20+20);

% A factor that determines duration of a time step
% wrt CFL limit
courant_factor = 0.9;

% A factor determining the accuracy limit of FDTD results
number_of_cells_per_wavelength = 20;

% Dimensions of a unit cell in x, y, and z directions (meters
)
dx = 0.4e-3;
dy = 0.4e-3;
dz = 0.4e-3;

% ==<boundary conditions>=====
% Here we define the boundary conditions parameters
% 'pec' : perfect electric conductor
% 'cpml' : convolutional PML
% if cpml_number_of_cells is less than zero
% CPML extends inside of the domain rather than outwards

boundary.type_xn = 'cpml';

```

```
boundary.air_buffer_number_of_cells_xn = 10;
boundary.cpml_number_of_cells_xn = 8;

boundary.type_xp = 'cpml';
boundary.air_buffer_number_of_cells_xp = 10;
boundary.cpml_number_of_cells_xp = 8;

boundary.type_yn = 'cpml';
boundary.air_buffer_number_of_cells_yn = 10;
boundary.cpml_number_of_cells_yn = 8;

boundary.type_yp = 'cpml';
boundary.air_buffer_number_of_cells_yp = 10;
boundary.cpml_number_of_cells_yp = 8;

boundary.type_zn = 'cpml';
boundary.air_buffer_number_of_cells_zn = 10;
boundary.cpml_number_of_cells_zn = 8;

boundary.type_zp = 'cpml';
boundary.air_buffer_number_of_cells_zp = 10;
boundary.cpml_number_of_cells_zp = 8;

boundary.cpml_order = 3;
boundary.cpml_sigma_factor = 1.3;
boundary.cpml_kappa_max = 7;
boundary.cpml_alpha_min = 0;
boundary.cpml_alpha_max = 0.05;

% ===<material types>=====
% Here we define and initialize the arrays of material types
% eps_r    : relative permittivity
% mu_r     : relative permeability
% sigma_e  : electric conductivity
% sigma_m  : magnetic conductivity

% air
material_types(1).eps_r = 1;
material_types(1).mu_r = 1;
material_types(1).sigma_e = 0;
material_types(1).sigma_m = 0;
material_types(1).color = [1 1 1];

% PEC : perfect electric conductor
material_types(2).eps_r = 1;
material_types(2).mu_r = 1;
```

```

material_types(2).sigma_e = 1e10;
material_types(2).sigma_m = 0;
material_types(2).color   = [1 0 0];

% PMC : perfect magnetic conductor
material_types(3).eps_r   = 1;
material_types(3).mu_r    = 1;
material_types(3).sigma_e = 0;
material_types(3).sigma_m = 1e10;
material_types(3).color   = [0 1 0];

% PCB substrate
material_types(4).eps_r   = 3.5;    %optimisticky, dost mozna
                                     trosku vyssi
material_types(4).mu_r    = 1;
material_types(4).sigma_e = 0;
material_types(4).sigma_m = 0;
material_types(4).color   = [0 0 1];

% index of material types defining air, PEC, and PMC
material_type_index_air = 1;
material_type_index_pec = 2;
material_type_index_pmc = 3;

```

Kód D.2: *define_problem_space_parameters*

```

disp('defining sources and lumped element components');

voltage_sources = [];
current_sources = [];
diodes = [];
inductors = [];
capacitors = [];

% % define source waveform types and parameters
waveforms.sinusoidal(1).frequency = 2.9e9;

% voltage sources
% direction: 'xp', 'xn', 'yp', 'yn', 'zp', or 'zn'
% resistance : ohms, magitude : volts
voltage_sources(1).min_x = 1.6e-3;
voltage_sources(1).min_y = -0.8e-3;
voltage_sources(1).min_z = 0;
voltage_sources(1).max_x = 1.6e-3;
voltage_sources(1).max_y = 0.8e-3;
voltage_sources(1).max_z = 0.8e-3;
voltage_sources(1).direction = 'yp';

```

```

voltage_sources(1).resistance = 50;
voltage_sources(1).magnitude = 1.3;
voltage_sources(1).waveform_type = 'sinusoidal';
voltage_sources(1).waveform_index = 1;

% diodes
% direction: 'xp', 'xn', 'yp', 'yn', 'zp', 'zn',
% diodes(1).min_x = 0;
% diodes(1).min_y = 0;
% diodes(1).min_z = 0.4e-3;
% diodes(1).max_x = 0;
% diodes(1).max_y = 0.4e-3;
% diodes(1).max_z = 0.4e-3;
% diodes(1).direction = 'yp';

```

Kód D.3: *define_sources_and_lumped_elements*

```

disp('defining output parameters');

sampled_electric_fields = [];
sampled_magnetic_fields = [];
sampled_voltages = [];
sampled_currents = [];
ports = [];
%farfield.frequencies = [];

% figure refresh rate
plotting_step = 10;

% mode of operation
run_simulation = true;
show_material_mesh = false;
show_problem_space = true;

% far field calculation parameters
farfield.frequencies(1) = 2.9e9;
farfield.number_of_cells_from_outer_boundary = 13;

% frequency domain parameters
frequency_domain.start = 2e9*2;
frequency_domain.end = 6.5e9*2;
frequency_domain.step = 20e6;

%diode
%define sampled voltages
% sampled_voltages(1).min_x = 0;
% sampled_voltages(1).min_y = 0;

```


Příloha E

Definiční skripty pro simulaci flíčkové antény zatížené diodou

```
disp('defining the problem geometry');

bricks = [];
spheres = [];
thin_wires = [];

% dielectric
bricks(1).min_x = -52.7e-3;
bricks(1).min_y = -26.15e-3;
bricks(1).min_z = -1.5e-3;
bricks(1).max_x = 52.7e-3;
bricks(1).max_y = 26.15e-3;
bricks(1).max_z = 0;
bricks(1).material_type = material_type_index_dielectric;

%%
% transponder
bricks(2).min_x = -47.7e-3;
bricks(2).min_y = -20.15e-3;
bricks(2).min_z = 0;
bricks(2).max_x = 47.7e-3;
bricks(2).max_y = 20.15e-3;
bricks(2).max_z = 0;
bricks(2).material_type = material_type_index_transponder;

%%
% cutting into the transponder

% 4 cuts in the center - right, down
bricks(3).min_x = 1.5e-3;
bricks(3).min_y = -11.3e-3; %-8.3e-3 - 3e-3
bricks(3).min_z = 0;
bricks(3).max_x = 38e-3;    %1.5+36.5
```



```
bricks(8).max_y = 20.15e-3;
bricks(8).max_z = 0e-3;
bricks(8).material_type = material_type_index_dielectric;

%4 cuts on the edges - left, down
bricks(9).min_x = -31.5e-3;
bricks(9).min_y = -20.15e-3;
bricks(9).min_z = 0;
bricks(9).max_x = -16.5e-3;
bricks(9).max_y = -17.15e-3;
bricks(9).max_z = 0e-3;
bricks(9).material_type = material_type_index_dielectric;

%4 cuts on the edges - left, up
bricks(10).min_x = -31.5e-3;
bricks(10).min_y = 17.15e-3; %20.15-3
bricks(10).min_z = 0;
bricks(10).max_x = -16.5e-3;
bricks(10).max_y = 20.15e-3;
bricks(10).max_z = 0e-3;
bricks(10).material_type = material_type_index_dielectric;

% space between the two halves
bricks(11).min_x = -1.5e-3;
bricks(11).min_y = -20.15e-3;
bricks(11).min_z = 0;
bricks(11).max_x = 1.5e-3;
bricks(11).max_y = 20.15e-3;
bricks(11).max_z = 0;
bricks(11).material_type = material_type_index_dielectric;

%%
%notches in the middle - right, down
bricks(12).min_x = 1.5e-3;
bricks(12).min_y = -1.5e-3;
bricks(12).min_z = 0;
bricks(12).max_x = 11.5e-3;
bricks(12).max_y = -1e-3;
bricks(12).max_z = 0e-3;
bricks(12).material_type = material_type_index_dielectric;

%notches in the middle - right, up
bricks(13).min_x = 1.5e-3;
bricks(13).min_y = 1e-3;
bricks(13).min_z = 0;
bricks(13).max_x = 11.5e-3;
```



```
bricks(18).min_z = -1.5e-3;
bricks(18).max_x = -21e-3+0.5e-3;
bricks(18).max_y = -15e-3-0.5e-3;
bricks(18).max_z = 0;
bricks(18).material_type = material_type_index_transponder;

bricks(19).min_x = -21e-3+0.5e-3;
bricks(19).min_y = -15e-3-0.5e-3;
bricks(19).min_z = -1.5e-3;
bricks(19).max_x = -21e-3+0.5e-3;
bricks(19).max_y = -15e-3+0.5e-3;
bricks(19).max_z = 0;
bricks(19).material_type = material_type_index_transponder;

bricks(20).min_x = -21e-3-0.5e-3;
bricks(20).min_y = -15e-3+0.5e-3;
bricks(20).min_z = -1.5e-3;
bricks(20).max_x = -21e-3+0.5e-3;
bricks(20).max_y = -15e-3+0.5e-3;
bricks(20).max_z = 0;
bricks(20).material_type = material_type_index_transponder;

% shortening pins - left up
bricks(21).min_x = -21e-3-0.5e-3;
bricks(21).min_y = 15e-3-0.5e-3;
bricks(21).min_z = -1.5e-3;
bricks(21).max_x = -21e-3-0.5e-3;
bricks(21).max_y = 15e-3+0.5e-3;
bricks(21).max_z = 0;
bricks(21).material_type = material_type_index_transponder;

bricks(22).min_x = -21e-3-0.5e-3;
bricks(22).min_y = 15e-3-0.5e-3;
bricks(22).min_z = -1.5e-3;
bricks(22).max_x = -21e-3+0.5e-3;
bricks(22).max_y = 15e-3-0.5e-3;
bricks(22).max_z = 0;
bricks(22).material_type = material_type_index_transponder;

bricks(23).min_x = -21e-3+0.5e-3;
bricks(23).min_y = 15e-3-0.5e-3;
bricks(23).min_z = -1.5e-3;
bricks(23).max_x = -21e-3+0.5e-3;
bricks(23).max_y = 15e-3+0.5e-3;
bricks(23).max_z = 0;
bricks(23).material_type = material_type_index_transponder;
```



```
bricks(29).max_x = 21e-3-0.5e-3;
bricks(29).max_y = 15e-3+0.5e-3;
bricks(29).max_z = 0;
bricks(29).material_type = material_type_index_transponder;

bricks(30).min_x = 21e-3-0.5e-3;
bricks(30).min_y = 15e-3-0.5e-3;
bricks(30).min_z = -1.5e-3;
bricks(30).max_x = 21e-3+0.5e-3;
bricks(30).max_y = 15e-3-0.5e-3;
bricks(30).max_z = 0;
bricks(30).material_type = material_type_index_transponder;

bricks(31).min_x = 21e-3+0.5e-3;
bricks(31).min_y = 15e-3-0.5e-3;
bricks(31).min_z = -1.5e-3;
bricks(31).max_x = 21e-3+0.5e-3;
bricks(31).max_y = 15e-3+0.5e-3;
bricks(31).max_z = 0;
bricks(31).material_type = material_type_index_transponder;

bricks(32).min_x = 21e-3-0.5e-3;
bricks(32).min_y = 15e-3+0.5e-3;
bricks(32).min_z = -1.5e-3;
bricks(32).max_x = 21e-3+0.5e-3;
bricks(32).max_y = 15e-3+0.5e-3;
bricks(32).max_z = 0;
bricks(32).material_type = material_type_index_transponder;

% VIA 2
% shortening pins - left, down
bricks(33).min_x = -38e-3-0.5e-3;
bricks(33).min_y = -0.5e-3-0.5e-3;
bricks(33).min_z = -1.5e-3;
bricks(33).max_x = -38e-3-0.5e-3;
bricks(33).max_y = -0.5e-3+0.5e-3;
bricks(33).max_z = 0e-3;
bricks(33).material_type = material_type_index_transponder;

bricks(34).min_x = -38e-3-0.5e-3;
bricks(34).min_y = -0.5e-3-0.5e-3;
bricks(34).min_z = -1.5e-3;
bricks(34).max_x = -38e-3+0.5e-3;
bricks(34).max_y = -0.5e-3-0.5e-3;
bricks(34).max_z = 0e-3;
```

```
bricks(34).material_type = material_type_index_transponder;

bricks(35).min_x = -38e-3+0.5e-3;
bricks(35).min_y = -0.5e-3-0.5e-3;
bricks(35).min_z = -1.5e-3;
bricks(35).max_x = -38e-3+0.5e-3;
bricks(35).max_y = -0.5e-3+0.5e-3;
bricks(35).max_z = 0e-3;
bricks(35).material_type = material_type_index_transponder;

bricks(36).min_x = -38e-3-0.5e-3;
bricks(36).min_y = -0.5e-3+0.5e-3;
bricks(36).min_z = -1.5e-3;
bricks(36).max_x = -38e-3+0.5e-3;
bricks(36).max_y = -0.5e-3+0.5e-3;
bricks(36).max_z = 0e-3;
bricks(36).material_type = material_type_index_transponder;

% shortening pins - left up
bricks(37).min_x = -38e-3-0.5e-3;
bricks(37).min_y = 0.6e-3-0.5e-3;
bricks(37).min_z = -1.5e-3;
bricks(37).max_x = -38e-3-0.5e-3;
bricks(37).max_y = 0.6e-3+0.5e-3;
bricks(37).max_z = 0e-3;
bricks(37).material_type = material_type_index_transponder;

bricks(38).min_x = -38e-3-0.5e-3;
bricks(38).min_y = 0.6e-3-0.5e-3;
bricks(38).min_z = -1.5e-3;
bricks(38).max_x = -38e-3+0.5e-3;
bricks(38).max_y = 0.6e-3-0.5e-3;
bricks(38).max_z = 0e-3;
bricks(38).material_type = material_type_index_transponder;

bricks(39).min_x = -38e-3+0.5e-3;
bricks(39).min_y = 0.6e-3-0.5e-3;
bricks(39).min_z = -1.5e-3;
bricks(39).max_x = -38e-3+0.5e-3;
bricks(39).max_y = 0.6e-3+0.5e-3;
bricks(39).max_z = 0e-3;
bricks(39).material_type = material_type_index_transponder;

bricks(40).min_x = -38e-3-0.5e-3;
bricks(40).min_y = 0.6e-3+0.5e-3;
bricks(40).min_z = -1.5e-3;
```

```
bricks(40).max_x = -38e-3+0.5e-3;
bricks(40).max_y = 0.6e-3+0.5e-3;
bricks(40).max_z = 0e-3;
bricks(40).material_type = material_type_index_transponder;

% shortening pins - right, down
bricks(41).min_x = 38e-3-0.5e-3;
bricks(41).min_y = -0.5e-3-0.5e-3;
bricks(41).min_z = -1.5e-3;
bricks(41).max_x = 38e-3-0.5e-3;
bricks(41).max_y = -0.5e-3+0.5e-3;
bricks(41).max_z = 0e-3;
bricks(41).material_type = material_type_index_transponder;

bricks(42).min_x = 38e-3-0.5e-3;
bricks(42).min_y = -0.5e-3-0.5e-3;
bricks(42).min_z = -1.5e-3;
bricks(42).max_x = 38e-3+0.5e-3;
bricks(42).max_y = -0.5e-3-0.5e-3;
bricks(42).max_z = 0e-3;
bricks(42).material_type = material_type_index_transponder;

bricks(43).min_x = 38e-3+0.5e-3;
bricks(43).min_y = -0.5e-3-0.5e-3;
bricks(43).min_z = -1.5e-3;
bricks(43).max_x = 38e-3+0.5e-3;
bricks(43).max_y = -0.5e-3+0.5e-3;
bricks(43).max_z = 0e-3;
bricks(43).material_type = material_type_index_transponder;

bricks(44).min_x = 38e-3-0.5e-3;
bricks(44).min_y = -0.5e-3+0.5e-3;
bricks(44).min_z = -1.5e-3;
bricks(44).max_x = 38e-3+0.5e-3;
bricks(44).max_y = -0.5e-3+0.5e-3;
bricks(44).max_z = 0e-3;
bricks(44).material_type = material_type_index_transponder;

% shortening pins - right up
bricks(45).min_x = 38e-3-0.5e-3;
bricks(45).min_y = 0.6e-3-0.5e-3;
bricks(45).min_z = -1.5e-3;
bricks(45).max_x = 38e-3-0.5e-3;
bricks(45).max_y = 0.6e-3+0.5e-3;
bricks(45).max_z = 0e-3;
bricks(45).material_type = material_type_index_transponder;
```

```
bricks(46).min_x = 38e-3-0.5e-3;
bricks(46).min_y = 0.6e-3-0.5e-3;
bricks(46).min_z = -1.5e-3;
bricks(46).max_x = 38e-3+0.5e-3;
bricks(46).max_y = 0.6e-3-0.5e-3;
bricks(46).max_z = 0e-3;
bricks(46).material_type = material_type_index_transponder;

bricks(47).min_x = 38e-3+0.5e-3;
bricks(47).min_y = 0.6e-3-0.5e-3;
bricks(47).min_z = -1.5e-3;
bricks(47).max_x = 38e-3+0.5e-3;
bricks(47).max_y = 0.6e-3+0.5e-3;
bricks(47).max_z = 0e-3;
bricks(47).material_type = material_type_index_transponder;

bricks(48).min_x = 38e-3-0.5e-3;
bricks(48).min_y = 0.6e-3+0.5e-3;
bricks(48).min_z = -1.5e-3;
bricks(48).max_x = 38e-3+0.5e-3;
bricks(48).max_y = 0.6e-3+0.5e-3;
bricks(48).max_z = 0e-3;
bricks(48).material_type = material_type_index_transponder;

%%
% to diode
bricks(49).min_x = -1.5e-3;
bricks(49).min_y = -0.5e-3;
bricks(49).min_z = 0;
bricks(49).max_x = -0e-3;
bricks(49).max_y = 0;
bricks(49).max_z = 0e-3;
bricks(49).material_type = material_type_index_pec;

bricks(50).min_x = 0.5e-3;
bricks(50).min_y = -0.5e-3;
bricks(50).min_z = 0;
bricks(50).max_x = 1.5e-3;
bricks(50).max_y = 0;
bricks(50).max_z = 0e-3;
bricks(50).material_type = material_type_index_pec;
```

Kód E.1: *define_geomtery*

```
disp('defining the problem space parameters');
```

```

% maximum number of time steps to run FDTD simulation
number_of_time_steps = 1310 * 10 ;%* 50;

% A factor that determines duration of a time step
% wrt CFL limit
courant_factor = 0.9;

% A factor determining the accuracy limit of FDTD results
number_of_cells_per_wavelength = 20;

% Dimensions of a unit cell in x, y, and z directions (meters
)
dx = 0.5e-3;
dy = 0.5e-3;
dz = 0.5e-3;

% ==<boundary conditions>=====
% Here we define the boundary conditions parameters
% 'pec' : perfect electric conductor
% 'cpml' : convolutional PML
% if cpml_number_of_cells is less than zero
% CPML extends inside of the domain rather than outwards

boundary.type_xn = 'cpml';
boundary.air_buffer_number_of_cells_xn = 5;
boundary.cpml_number_of_cells_xn = 4;

boundary.type_xp = 'cpml';
boundary.air_buffer_number_of_cells_xp = 5;
boundary.cpml_number_of_cells_xp = 4;

boundary.type_yn = 'cpml';
boundary.air_buffer_number_of_cells_yn = 5;
boundary.cpml_number_of_cells_yn = 4;

boundary.type_yp = 'cpml';
boundary.air_buffer_number_of_cells_yp = 5;
boundary.cpml_number_of_cells_yp = 4;

boundary.type_zn = 'cpml';
boundary.air_buffer_number_of_cells_zn = 7;
boundary.cpml_number_of_cells_zn = 4;

boundary.type_zp = 'cpml';
boundary.air_buffer_number_of_cells_zp = 7; %far field

```

```
boundary.cpml_number_of_cells_zp = 4;

boundary.cpml_order = 3;
boundary.cpml_sigma_factor = 1.3;
boundary.cpml_kappa_max = 7;
boundary.cpml_alpha_min = 0;
boundary.cpml_alpha_max = 0.05;

% ===<material types>=====
% Here we define and initialize the arrays of material types
% eps_r    : relative permittivity
% mu_r     : relative permeability
% sigma_e  : electric conductivity
% sigma_m  : magnetic conductivity

% air
material_types(1).eps_r = 1;
material_types(1).mu_r = 1;
material_types(1).sigma_e = 0;
material_types(1).sigma_m = 0;
material_types(1).color = [1 1 1];

% PEC : perfect electric conductor
material_types(2).eps_r = 1;
material_types(2).mu_r = 1;
material_types(2).sigma_e = 1e10;
material_types(2).sigma_m = 0;
material_types(2).color = [1 0 0];

% PMC : perfect magnetic conductor
material_types(3).eps_r = 1;
material_types(3).mu_r = 1;
material_types(3).sigma_e = 0;
material_types(3).sigma_m = 1e10;
material_types(3).color = [0 1 0];

% patch + ground plane
material_types(4).eps_r = 1;
material_types(4).mu_r = 1;
material_types(4).sigma_e = 5.6e7;      %Cu
material_types(4).sigma_m = 0;
material_types(4).color = [1 1 0];

% dielectric
material_types(5).eps_r = 3.66;
material_types(5).mu_r = 1;
```

```

material_types(5).sigma_e = 0;
material_types(5).sigma_m = 0;
material_types(5).color   = [0 0 1];

% index of material types defining air, PEC, and PMC
material_type_index_air = 1;
material_type_index_pec = 2;
material_type_index_pmc = 3;
material_type_index_transponder = 4;
material_type_index_dielectric = 5;

```

Kód E.2: *define_problem_space_parameters*

```

disp('defining sources and lumped element components');

voltage_sources = [];
current_sources = [];
diodes = [];
inductors = [];
capacitors = [];

% define source waveform types and parameters
waveforms.sinusoidal(1).frequency = 0.867e9;

% voltage sources
% direction: 'xp', 'xn', 'yp', 'yn', 'zp', or 'zn'
% resistance : ohms, magitude : volts
voltage_sources(1).min_x = -0e-3;
voltage_sources(1).min_y = -0.5e-3;
voltage_sources(1).min_z = 0;
voltage_sources(1).max_x = 0.5e-3;
voltage_sources(1).max_y = 0e-3;
voltage_sources(1).max_z = 0;
voltage_sources(1).direction = 'xp';
voltage_sources(1).resistance = 50;
voltage_sources(1).magnitude = 5;
voltage_sources(1).waveform_type = 'sinusoidal';
voltage_sources(1).waveform_index = 1;

% diodes
% direction: 'xp', 'xn', 'yp', 'yn', 'zp', 'zn',
% diodes(1).min_x = -0e-3;
% diodes(1).min_y = 0;
% diodes(1).min_z = 0;
% diodes(1).max_x = 0.5e-3;
% diodes(1).max_y = 0e-3;
% diodes(1).max_z = 0;

```

```
% diodes(1).direction = 'xp';
```

Kód E.3: *define_sources_and_lumped_elements*

```
disp('defining output parameters');

sampled_electric_fields = [];
sampled_magnetic_fields = [];
sampled_voltages = [];
sampled_currents = [];
ports = [];
%farfield.frequencies = [];

% figure refresh rate
plotting_step = 10;

% mode of operation
run_simulation = true;
show_material_mesh = false;
show_problem_space = true;

% far field calculation parameters
farfield.frequencies(1) = 0.867e9;
farfield.number_of_cells_from_outer_boundary = 5;

% frequency domain parameters
frequency_domain.start = 0.5e9;
frequency_domain.end = 2e9;
frequency_domain.step = 10e5;

%define sampled voltages
sampled_voltages(1).min_x = 0e-3;
sampled_voltages(1).min_y = 0e-3;
sampled_voltages(1).min_z = 0;
sampled_voltages(1).max_x = 0.5e-3;
sampled_voltages(1).max_y = 0e-3;
sampled_voltages(1).max_z = 0;
sampled_voltages(1).direction = 'xp';
sampled_voltages(1).display_plot = false;

% define sampled currents
% diode current
sampled_currents(1).min_x = 0e-3;
sampled_currents(1).min_y = 0e-3;
sampled_currents(1).min_z = 0;
sampled_currents(1).max_x = 0.5e-3;
sampled_currents(1).max_y = 0e-3;
```



```
sampled_currents(1).max_z = 0;  
sampled_currents(1).direction = 'xp';  
sampled_currents(1).display_plot = false;  
  
inst_powers(1).sampled_voltage_index = 1;  
inst_powers(1).sampled_current_index = 1;
```

Kód E.4: *define_output_parameters*