

Diplomová práce



České  
vysoké  
učení technické  
v Praze

**F3**

Fakulta elektrotechnická  
Katedra mikroelektroniky

## Analýzy v grafickém editoru elektrotechnických obvodů

Bc. Ondřej Bukovský

Vedoucí: doc. Dr. Ing. Jiří Hospodka  
Studijní program: Elektronika a komunikace  
Květen 2021



## I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Bukovský** Jméno: **Ondřej** Osobní číslo: **465957**  
Fakulta/ústav: **Fakulta elektrotechnická**  
Zadávající katedra/ústav: **Katedra mikroelektroniky**  
Studijní program: **Elektronika a komunikace**  
Specializace: **Elektronika**

## II. ÚDAJE K DIPLOMOVÉ PRÁCI

Název diplomové práce:

**Analýzy v grafickém editoru elektrotechnických obvodů**

Název diplomové práce anglicky:

**Analysis in Schematic Editor of Electronic Circuits**

Pokyny pro vypracování:

Provedte návrh nového uspořádání analýz v simulátoru elektrických a elektronických obvodů v 'Grafickém editoru elektrotechnických obvodů' (GEEC) psaného v jazyce JavaScript. Zaměřte se jak na symbolické, tak numerické analýzy spojitě pracujících obvodů i vybrané analýzy periodicky spínaných obvodů. Navrhněte jejich novou strukturu a implementujte ji do editoru v jazyce JavaScript tak, aby bylo možné přidávat i další simulátory. Vylepšete možnosti zadávání analýz, zejména parametrických pro více proměnných. Dále se zaměřte na zobrazování grafů, jejich možnosti a uživatelskou podporou. V sekci analýz přidejte spektrální analýzu a analýzu přenosových členů definovaných semisymbolicky pro numerický simulátor. Zhodnoťte možnosti zpětného zobrazování výsledků analýz ve schématu a implementujte variantu dle dohody s vedoucím práce. Při implementaci dbejte na co největší zpětnou kompatibilitu s dosavadní aplikací.

Seznam doporučené literatury:

- [1] Paulů F.: Grafický editor elektrotechnických obvodů, diplomová práce, ČVUT 2015.
- [2] Tuma T., Burmen A.: Circuit Simulation with SPICE OPUS Theory and Practice, Birkhauser Boston, a part of Springer Science+Business Media, LLC 2009, ISBN: 978-0-8176-4866-4.
- [3] <http://www.w3schools.com/js/default.asp>
- [4] Bukovský O.: Analýzy v grafickém editoru elektrotechnických obvodů, bakalářská práce, ČVUT 2019.

Jméno a pracoviště vedoucí(ho) diplomové práce:


**doc. Dr. Ing. Jiří Hospodka, katedra teorie obvodů FEL**

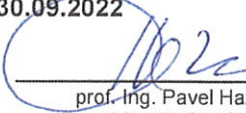
Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) diplomové práce:

Datum zadání diplomové práce: **03.02.2021**

Termín odevzdání diplomové práce: **21.5.2021**

Platnost zadání diplomové práce: **30.09.2022**

  
doc. Dr. Ing. Jiří Hospodka  
podpis vedoucí(ho) práce

  
prof. Ing. Pavel Hazdra, CSc.  
podpis vedoucí(ho) ústavu/katedry


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

## III. PŘEVZETÍ ZADÁNÍ

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

**17.5.2021**

Datum převzetí zadání

  
Podpis studenta



## Poděkování

Poděkování patří především doc. Dr. Ing. Jiřímu Hospodkovi, jenž byl můj vedoucí práce, konzultant, tester a recenzent, který je velký fanda a uživatel této aplikace. Dále pak Ing. Filipovi Paulů, původnímu vývojáři GEEC, za programátorské rady a tipy.

## 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, 20. května 2021

## Abstrakt

Tato práce popisuje metodiku analýzy elektrických obvodů pomocí počítače. Dále seznamuje s vybranými webovými aplikacemi pro kreslení a simulaci obvodů. Praktická část se zabývá Grafickým editorem elektrických obvodů (GEEC), psaného v jazyce JavaScript. Řeší problematiku analýz, jejich vylepšení a nové možnosti.

**Klíčová slova:** GEEC, NGSPICE, PraCA<sub>n</sub>, Maple, JavaScript, D3, VEGA, Webové aplikace, Analýzy obvodů

**Vedoucí:** doc. Dr. Ing. Jiří Hospodka  
Katedra teorie obvodů,  
Technická 2,  
Praha 6

## Abstract

This thesis describes methods of analyzing electrical circuits using computers. It also introduces some web applications for making electronic circuit diagrams and simulations. The practical part deals with the Graphic Editor of Electrical Circuits (GEEC), written in JavaScript. It solves issues of analysis, their improvement, and new possibilities.

**Keywords:** GEEC, NGSPICE, PraCA<sub>n</sub>, Maple, JavaScript, D3, VEGA, Web applications, Circuit analysis

**Title translation:** Analysis in schematic editor of electronic circuits

# Obsah

<b>1 Úvod</b>	<b>1</b>	4.3 D3	38
		4.3.1 VEGA	38
		4.4 Recharts	38
		4.5 Chart.js	39
		4.6 Victory	39
		4.7 Google Charts	39
		4.7.1 Shrnutí	40
		<b>Část II</b>	
		<b>Vlastní práce</b>	
		<b>5 Grafy</b>	<b>43</b>
		<b>6 NGSPICE</b>	<b>47</b>
		6.1 Nastavení NGSPICE	48
		6.1.1 Obecné nastavení simulátoru	48
		6.1.2 IC a Nodeset	49
		6.2 Vlastní skripty	50
		<b>7 Analýzy</b>	<b>51</b>
		7.1 Nové možnosti v analýzách	51
		7.1.1 Menu	51
		7.1.2 Více grafů v jednom okně	52
		7.1.3 Design variables	53
		7.1.4 Substituce a přiřazení	53
		7.2 AC analýza jedné frekvence	54
		7.3 AC analýza - rozmítání parametru	55
		7.4 AC analýza - fázory	56
		7.5 AC analýza - hodograf	57
		7.6 AC analýza - skupinové zpoždění	58
		7.7 Citlivostní analýza	59
		7.8 Fourierova analýza	60
		7.9 FFT analýza	61
		7.10 PZ analýza	62
		7.11 Transientní analýza - XY mód	64
		<b>8 Výběr výsledků</b>	<b>65</b>
		8.1 Kalkulátor	65
		8.2 Nastavení parametrické analýzy	67
		<b>9 Interaktivní analýza</b>	<b>69</b>
		<b>10 Závěr</b>	<b>71</b>
		<b>Literatura</b>	<b>73</b>
		<b>Seznam použitých zkratk</b>	<b>77</b>
<b>2 Analýzy elektrických obvodů</b>	<b>5</b>		
2.1 Metodika	6		
2.1.1 Modifikovaná metoda uzlových napětí	7		
2.1.2 Řešení lineárních obvodů	8		
2.1.3 Řešení nelineárních obvodů	10		
2.1.4 Řešení obvodů v časové oblasti	14		
2.2 Typy analýz	20		
2.2.1 OP	21		
2.2.2 DC sweep	22		
2.2.3 AC	23		
2.2.4 Transientní	24		
2.2.5 Fourierova analýza	25		
2.2.6 Spektrální analýza	25		
2.2.7 Citlivostní analýza	25		
2.2.8 PZ analýza	26		
<b>3 Simulátory</b>	<b>27</b>		
3.1 Typy simulátorů	27		
3.1.1 Simulátory analogových obvodů	27		
3.1.2 Simulátory digitálních obvodů	27		
3.1.3 Smíšené simulátory	28		
3.1.4 Symbolické simulátory	28		
3.2 Použitelné simulátory	29		
3.2.1 WINSPIICE	29		
3.2.2 HSPICE	29		
3.2.3 SPICE OPUS	30		
3.2.4 NGSPICE	30		
3.2.5 Shrnutí	31		
3.3 Online simulátory	32		
3.3.1 Circuit Simulator	32		
3.3.2 Every Circuit	32		
3.3.3 EasyEDA	32		
3.3.4 PartSim	33		
3.3.5 CircuitLab	33		
3.3.6 TinaCloud	33		
3.3.7 Multisim Live	34		
3.3.8 GEEC	34		
<b>4 Knihovny pro vizualizaci dat</b>	<b>37</b>		
4.1 amCharts	37		
4.2 CanvasJS	37		

## Obrázky

2.1 Obvod pro ilustraci MMUN. ....	7	7.11 Numerické výsledky parametrické PZ analýzy. ....	62
2.2 Grafické znázornění Newtonovy metody [40]. ....	12	7.12 Symbolické výsledky PZ analýzy.	63
2.3 Obvod pro ilustraci Newtonovy metody. ....	12	7.13 Hysterézní křivka zobrazená pomocí transientní analýzy. ....	64
2.4 Napětí $U_D$ v závislosti na iteraci při různých $x_0$ . ....	13	8.1 Minimalizovaný kalkulátor. ....	65
2.5 Algoritmus řešení transientní analýzy ve SPICE [34]. ....	14	8.2 Kalkulátor. ....	66
2.6 Obvod pro ilustraci numerického řešení diferenciálních rovnic. ....	15	8.3 Okno pro nastavení parametrické analýzy. ....	67
2.7 Numerické řešení diferenciální rovnice 2.31 Eulerovou metodou. . .	17	8.4 Předání nastavení do konfiguračního panelu analýzy. . . .	67
2.8 Numerické řešení dif. rovnice 2.31 lichoběžníkovou metodou. ....	18	9.1 Výsledky analýzy ve schématu a interaktivní změna parametrů. ....	69
2.9 Struktura OP analýzy [12]. ....	21		
2.10 Struktura DC analýzy [12]. . . .	22		
2.11 Struktura AC analýzy [12]. ....	23		
2.12 Struktura transientní analýzy [12]. ....	24		
3.1 Grafický editor elektrických obvodů (GEEC). ....	35		
5.1 Nastavení grafů - záložka General.	44		
5.2 Nastavení grafů - záložka Axis. .	45		
5.3 Nastavení grafů - záložka Grid. .	45		
5.4 Nastavení grafů - záložka Data. .	46		
6.1 Možnosti nastavení NGSPICE. .	48		
6.2 Nastavení IC a Nodeset. ....	49		
7.1 Více grafů v jednom okně. ....	52		
7.2 Nastavení hodnot Design variables. ....	53		
7.3 Okno AC analýzy jedné frekvence.	54		
7.4 Okno AC analýzy při rozmítání parametru. ....	55		
7.5 Okno AC analýzy se zobrazenými fázory. ....	56		
7.6 Okno AC analýzy se zobrazeným hodografem. ....	57		
7.7 Okno AC analýzy s výsledkem skupinového zpoždění. ....	58		
7.8 Okno citlivostní analýzy. ....	59		
7.9 Okno Fourierovy analýzy. ....	60		
7.10 Okno FFT analýzy. ....	61		



## Tabulky

3.1 Stručné shrnutí vlastností použitelných simulátorů. . . . .	31
4.1 Stručné shrnutí vlastností grafických knihoven. . . . .	40





# Kapitola 1

## Úvod

Tato práce se zabývá analýzami elektrických obvodů v Grafickém editoru elektrických obvodů (GEEC). Jedná se o online webovou aplikaci, která je dostupná na adrese <http://asinus.feld.cvut.cz/geec/>. Aplikace umožňuje nakreslit a následně zanalyzovat daný obvod.

V současné době existuje celá řada online simulátorů (viz kapitola 3.3), každý má svá specifika. Tento například nabízí nejen numerické ale i symbolické výsledky, dále export do L<sup>A</sup>T<sub>E</sub>Xu.

GEEC neslouží jen pro návrh a analýzu obvodů. Hojně je využíván i ve výuce. Cílem práce je přidání nových funkcionalit a tím i rozšíření možností jeho využití. Snaha je o zachování co nejlepší uživatelské přívětivosti a jednoduchosti.

Předmětem práce je nejen vylepšení stávajících analýz, ale i přidání nových. Konkrétními novými analýzami jsou PZ, citlivostní, Fourierova a FFT. Vylepšení se dočkaly například analýzy AC a Transientní. Úpravou také prošly možnosti zadávání analýz a manipulace nad výsledky (část II).

Pro pochopení jak funguje, a jaké jsou možnosti, je zde stručně vyložena teorie simulací obvodů. Popsány tu jsou používané numerické metody řešení lineárních, nelineárních a diferenciálních rovnic (kapitola 2).

S výsledky analýz souvisí i jejich zobrazení, proto se tato práce zabývá také vizualizací dat, možnostmi a vylepšením.





**Část I**

**Rešerše**



## Kapitola 2

### Analýzy elektrických obvodů

Elektrické obvody se analyzují za účelem zjištění jejich vlastností a chování. To je dáno různým zapojením různých obvodových prvků. Obvodové prvky neboli elektronické součástky fungují na určitých fyzikálních principech, a to především v oblasti elektřiny a magnetismu.

Dříve se místo simulací obvody realizovaly na nepájivých polích a fyzicky se zkoumaly jejich vlastnosti. S rozvojem integrovaných obvodů to již nebylo možné z hlediska rozdílného chování integrovaných a diskrétních součástek. Problém je takto zkoumat i obvody pracující na vyšších frekvencích.

Kupředu šel i vývoj výpočetní techniky a tedy i možnost obvody simulovat. Aby bylo možné provádět simulace, je potřeba daný obvod respektive jeho součástky matematicky popsat. Matematický model dané součástky vychází z fyzikálních zákonitostí, které se v součástce odehrávají. Reálnou součástku je takřka nemožné dokonale matematicky popsat. Existují proto různé modely a jejich úrovně, které se používají v závislosti na přesnosti a rychlosti simulace. Modelování součástek je disciplína sama o sobě a není to předmětem této práce. Nicméně tato problematika je podrobněji řešena například v [21].

V roce 1971 vznikl na kalifornské univerzitě v Berkeley program SPICE (Simulation Program with Integrated Circuit Emphasis). Jak název napovídá, tento program byl určen především pro simulaci integrovaných obvodů. Měl velmi dobře propracované modely bipolárního (Gummel-Poonův) a unipolárního (Shichman-Hodgesův) tranzistoru. Tím že to byl volně dostupný program, tak se hojně začal využívat a stal se z něj de facto standart [18]. Dnes existuje celá řada verzí SPICE, a to jak komerčních, tak i volně dostupných. Příkladem mohou být NGSPICE, SPICE OPUS, PSPICE, HSPICE a další.

Díky velkému zastoupení SPICE v oblasti simulací elektrických obvodů, bude zde dále v kapitole 2.1 vysvětlena metodika popisu a výpočtu obvodových rovnic. SPICE je univerzální program umožňující simulace obvodů pro nelineární DC, nelineární transientní a lineární AC analýzu [34].

Ačkoliv je SPICE navržen pro simulace elektrických obvodů, nachází také uplatnění v celé řadě jiných odvětví fyziky. Dobrá analogie elektrických obvodů je s vedením tepla, tedy termodynamika. Dále je využíván v optice a optoelektronice, mechanice, mikrofluidice nebo například v biologických systémech. To vše se popisuje pomocí diferenciálních rovnic, které nemá problém SPICE vyřešit [17].

## 2.1 Metodika

Zde bude vysvětlena metodika popisu obvodu, sestavování rovnic a jejich řešení pro SPICE simulátory. Obvod vstupuje do SPICE jako tzv. netlist, což je textová forma popisu součástek a jejich propojení. Pro ilustraci, jak takový netlist může vypadat, je tu ukázka 2.1.

**Ukázka kódu 2.1:** Netlist tranzistorového zesilovače.

```
Amplifier
Vin 8 0 dc 0 ac 1 0 sin 0 5m 10k 0 0
Vdd 3 0 dc 10
Q1 6 4 0 mod1
R1 3 4 150k
R2 3 6 1k
C1 1 4 1u
C2 6 out 1u
R3 out 0 1meg
.MODEL mod1 npn bf=100
```

První řádek je vždy název obvodu. Další řádky už jsou součástky popřípadě jejich modely. První písmeno je identifikátor součástky (R rezistor, L induktor, C kapacitor, Q bipolární tranzistor a další - to lze nalézt ve většině SPICE manuálech) následuje bez mezery název, který může být sekvencí písmen a čísel. Poté jsou názvy uzlů, mezi které je součástka připojena, taktéž se jedná o sekvencí písmen a čísel. Následují v závislosti na součástce její parametry nebo název modelu. U simulátorů s integrovaným editorem obvodu není třeba se zabývat netlistem, protože je vygenerován na základě nakresleného obvodu.

Je tedy dán obvod pomocí netlistu. Dalším krokem je sestavení obvodových rovnic. To lze provést několika metodami. Například metodou smyčkových proudů. Tato metoda vychází z druhého Kirchhoffova zákona, který říká, že suma napětí ( $U_k$ ) v uzavřené smyčce je rovna nule, tedy

$$\sum_{k=1}^n U_k = 0. \quad (2.1)$$

Další možností je metoda uzlových napětí (MUN), ta vychází z prvního Kirchhoffova zákona, jenž říká že, suma proudů  $I_k$  vtékajících/vytékajících z uzlu je rovna nule, tedy

$$\sum_{k=1}^n I_k = 0. \quad (2.2)$$

Znaménko proudů se volí tak, že proudy vstupující do uzlu jsou brány s kladným znaménkem a proudy vystupující z uzlu se záporným znaménkem.

Metoda uzlových napětí je vhodná pro počítačové zpracování, nicméně má řadu nevýhod, proto byla vytvořena modifikovaná metoda uzlových napětí (MMUN) viz kapitola 2.1.1. Tato metoda je využívána ve SPICE simulátorech pro sestavení soustavy rovnic daného obvodu [34].



### 2.1.1 Modifikovaná metoda uzlových napětí

U klasické MUN nelze jednoduše pracovat s nezávislými zdroji napětí, nejprve se musí převést na zdroj proudu s paralelní vodivostí podle Nortonova teorému. Dále nelze pracovat s řízenými zdroji (kromě zdroje proudu řízeného napětím). Obecně jde o napěťově definované prvky, jako jsou napěťové zdroje a induktory. Maticový zápis MUN vypadá následovně

$$\mathbf{Y} \cdot \vec{U} = \vec{I} \Rightarrow \vec{U} = \mathbf{Y}^{-1} \cdot \vec{I}, \quad (2.3)$$

kde  $\mathbf{Y}$  je admitanční matice,  $\mathbf{U}$  je vektor neznámých uzlových napětí a  $\mathbf{I}$  je vektor všech budících proudů do jednotlivých uzlů soustavy.

Výše zmíněné problémy řeší právě MMUN. Obecně zapsaná rovnice vypadá následovně

$$\mathbf{A} \cdot \vec{x} = \vec{z}. \quad (2.4)$$

Princip metody vyplývá z toho, že proměnnou nemusí být jenom napětí, ale i proud. Jde tedy o obecný vektor  $\vec{x}$ , který už nemá rozměr [V]. Dále už se nejedná o admitanční matici  $\mathbf{Y}$  a o obecnou matici  $\mathbf{A}$ . Totéž platí pro pravou stranu  $\vec{z}$ , která už nemá rozměr [A]. Tento obecný tvar lze dále rozepsat na

$$\begin{bmatrix} \mathbf{G} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix} \begin{bmatrix} \vec{V} \\ \vec{I} \end{bmatrix} = \begin{bmatrix} \vec{J} \\ \vec{F} \end{bmatrix}, \quad (2.5)$$

kde  $\mathbf{G}$  je submatice dána propojením pasivních prvků mezi sebou,

$\mathbf{B}$  je submatice dána zapojením napěťových zdrojů,

$\mathbf{C}$  je submatice dána zapojením napěťových zdrojů ( $\mathbf{B}$  a  $\mathbf{C}$  spolu souvisejí),

$\mathbf{D}$  je submatice dána řízenými zdroji (pokud nejsou, je nulová),

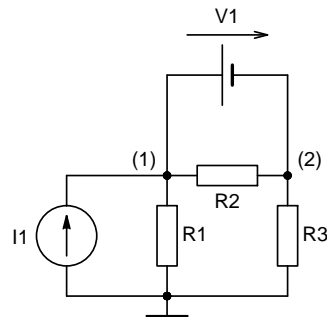
$\vec{V}$  je vektor neznámých uzlových napětí,

$\vec{I}$  je vektor neznámých proudů (zdroji napětí a induktorů),

$\vec{J}$  je vektor součtů proudů vstupujících do uzlů (0 nebo součet zdrojů proudu),

$\vec{F}$  je vektor hodnot nezávislých zdrojů napětí.

Jak tato metoda v praxi vypadá, bude ukázáno na obvodu, který je na Obrázku 2.1. Je to jednoduchý lineární obvod se zdrojem napětí (V1) a proudem (I1).



Obrázek 2.1: Obvod pro ilustraci MMUN.

Pomocí klasické MUN se sestaví obvodové rovnice. Obvod má dva uzly respektive tři. Třetí je referenční uzel (0), ale ten není uvažován v rovnicích (slouží pouze pro referenci nulového napětí). Rovnice 2.6 platí pro uzel (1), rovnice 2.7 pro uzel (2) a rovnice 2.8 je doplňující pro zdroj napětí  $V_1$ .

$$\frac{v(1)}{R_1} + \frac{v(1) - v(2)}{R_2} + i(V_1) = I_1 \quad (2.6)$$

$$\frac{v(2)}{R_3} + \frac{v(2) - v(1)}{R_2} - i(V_1) = 0 \quad (2.7)$$

$$v(1) - v(2) = V_1 \quad (2.8)$$

Přepsáním těchto rovnic do maticového tvar vznikne

$$\begin{bmatrix} \frac{1}{R_1} + \frac{1}{R_2} & -\frac{1}{R_2} & 1 \\ -\frac{1}{R_2} & \frac{1}{R_2} + \frac{1}{R_3} & -1 \\ 1 & -1 & 0 \end{bmatrix} \begin{bmatrix} v(1) \\ v(2) \\ i(V_1) \end{bmatrix} = \begin{bmatrix} I_1 \\ 0 \\ V_1 \end{bmatrix}, \quad (2.9)$$

jenž odpovídá tvaru rovnice 2.5, již s konkrétními hodnotami.

Závěrem lze poukázat na fakt, že SPICE defaultně vypisuje výsledky analýzy pracovního bodu obvodu (OP) ve formě uzlových napětí a proudů napěťovými zdroji a induktory, právě kvůli principu MMUN [34] [33].

### 2.1.2 Řešení lineárních obvodů

Řešení lineárních obvodů patří k nejjednodušší operacím, právě protože se jedná o lineární obvody. Tento algoritmus se používá pro výpočet pracovního bodu (OP) lineárních obvodů a pro AC analýzu, která pracuje s linearizovaným obvodem. Jediný rozdíl mezi nimi je v reprezentaci numerických hodnot. Pro OP jsou to reálná čísla a pro AC komplexní čísla.

Vychází se opět z rovnice 2.4 dané MMUN, kde vektor neznámých  $\vec{x}$  se vypočítá vynásobením vektoru  $\vec{z}$  zleva inverzní maticí maticí  $\mathbf{A}^{-1}$ . Výpočet inverzní matice je časově dost náročný, proto je upřednostňována Gaussova eliminační metoda [34].

$$\mathbf{A} \cdot \vec{x} = \vec{z} \Rightarrow \vec{x} = \mathbf{A}^{-1} \cdot \vec{z}. \quad (2.10)$$

### Gaussova eliminační metoda

Gaussova eliminační metoda (GEM) pomocí souboru pravidel pro manipulaci s řádky matice transformuje danou matici na horní trojúhelníkový tvar, tedy všechny prvky pod hlavní diagonálou jsou nulové. Souborem pravidel se myslí tři pravidla a těmi jsou: prohazování pořadí řádků, násobení řádku nenulovou konstantou a přičítání násobku jednoho řádku k jinému.

Rovnici 2.4 lze rozepsat do složek následujícím způsobem viz 2.11, kde se pro ilustraci bere v úvahu soustava tří rovnic o třech neznámých.

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} \quad (2.11)$$

Po GEM je matice v horním trojúhelníkovém tvaru (rovnice 2.12), s tím že GEM mění hodnoty koeficientů, takže obecné koeficienty  $a_{nn}$  se změny na jiné koeficienty  $b_{nn}$ . Totéž platí pro pravou stranu, která je GEM také ovlivněna, tedy ze  $z_n$  se stane  $y_n$ .

$$\begin{bmatrix} b_{11} & b_{12} & b_{13} \\ 0 & b_{22} & b_{23} \\ 0 & 0 & b_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} \quad (2.12)$$

Z tohoto tvaru je již možné vyjádřit jednotlivé složky vektoru neznámých  $\vec{x}$ , které pro tento příklad vypadají následovně

$$\begin{aligned} x_3 &= \frac{y_3}{b_{33}}, \\ x_2 &= \frac{y_2 - b_{23}x_3}{b_{22}}, \\ x_1 &= \frac{y_1 - b_{12}x_2 - b_{13}x_3}{b_{11}}. \end{aligned}$$

Toto byl způsob řešení soustavy rovnic pomocí GEM. V praxi se pak ještě používá modifikovaná varianta, která se nazývá LU rozklad. O tom bude více řečeno níže spolu se zmíněním jistých výhod oproti GEM [34] [9] [2].

## ■ LU rozklad

Jak již bylo zmíněno LU rozklad je modifikací GEM. Jde o to, že se matice  $\mathbf{A}$  zapíše pomocí dvou jiných matic. Jedná se o matice  $\mathbf{L}$  (lower triangular matrix), tedy matici ve spodním trojúhelníkovém tvaru a matici  $\mathbf{U}$  (upper triangular matrix) v horním trojúhelníkovém tvaru. Pro  $\mathbf{A}$  pak platí vztah

$$\mathbf{A} = \mathbf{LU}. \quad (2.13)$$

$\mathbf{U}$  je dána GEM matice  $\mathbf{A}$ .  $\mathbf{L}$  jsou škálovací faktory v každém kroku eliminace.

$$\mathbf{L} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ 0 & a_{22} & a_{23} \\ 0 & 0 & a_{33} \end{bmatrix}, \mathbf{U} = \begin{bmatrix} 1 & 0 & 0 \\ a_{21}/a_{11} & 1 & 0 \\ a_{31}/a_{11} & a_{32}/a_{22} & 1 \end{bmatrix} \quad (2.14)$$

Výpočet řešené soustavy rovnic se pak dělá ve dvou krocích. První krok dopředná substituce

$$\mathbf{LU} \cdot \vec{x} = \vec{z} \Rightarrow \mathbf{U}\vec{x} = \mathbf{L}^{-1} \cdot \vec{z} = \vec{y}. \quad (2.15)$$

Druhý krok zpětná substituce a tedy výsledkem je vektor neznámých

$$\vec{x} = \mathbf{U}^{-1} \cdot \vec{y} = \mathbf{U}^{-1}\mathbf{L}^{-1} \cdot \vec{z}. \quad (2.16)$$

Tím že  $\mathbf{L}$  a  $\mathbf{U}$  jsou v trojúhelníkovém tvaru, tak výpočet jejich inverzních matic je triviální. Další výhodou je to, že je obě lze uchovat v jednom paměťovém prostoru a to díky jejich trojúhelníkovým tvarům a tím, že matice  $\mathbf{U}$  má na hlavní diagonále jedničky.

Výhodou LU rozkladu oproti GEM je, že obvod lze opakovaně řešit pro různé pravé strany ( $\vec{z}$ ). Tato vlastnost je užitečná v určitých SPICE analýzách, jako jsou Sensitivity, Noise a Distortion analýzy [34] [9] [2].

## ■ LU rozklad - pivotování

Pivotování je proces systematického výběru pivotů pro Gaussovu eliminaci během LU rozkladu matice. Gaussova eliminace, která ve své podstatě nemusí být nestabilní, tak dobré zajistit, aby v každém kroku procesu eliminace byl nenulový pivot. To je důvod, proč je důležité pivotování při výpočtu LU rozkladu. Díky pivotování lze navíc zajistit víc, než jen aby byla Gaussova eliminace úspěšně dokončena. Lze snížit chyby zaokrouhlování během výpočtu. Relativně malé pivoty způsobují nestabilitu, protože během Gaussovy eliminace fungují podobně jako nuly. Prostřednictvím procesu pivotování lze tuto nestabilitu výrazně omezit tím, že zajistíme, aby jako pivot se použil prvek s největší hodnotou v daném sloupci matice. Tato technika má tendenci snižovat velikost odečítaného výrazu v každém kroku a tím zlepšovat přesnost [34] [9] [2].

### ■ 2.1.3 Řešení nelineárních obvodů

Obvod se stává nelineárním pokud, obsahuje alespoň jeden nelineární prvek, tzn. prvek s nelineární závislostí mezi napětím a proudem. Například s kvadratickou (MOS tranzistor) nebo exponenciální (BJT tranzistor) závislostí. Takovými prvky mohou být například polovodiče nebo nelineární řízené zdroje.

Pomocí metody MMUN se opět získá soustava rovnic v tomto případě už nelineárních. Řešení nelineárních rovnic je obecně složitý proces, v mnoha případech je nemožné získat analytický výsledek. Proto se hlavně pro strojové výpočty nelineárních rovnic používají numerické metody, které lze použít univerzálně i pro rovnice s analytickým řešením. Příkladem jednoduše řešitelné nelineární rovnice analytickou metodou může být kvadratická rovnice, pro kterou existují vzorce na výpočet diskriminantu a následně i kořenů.

V opačném případě nastupují numerické metody, jedná se o iterační metody, kdy se v jednotlivých krocích výpočtu získávají stále přesnější aproximace hledaného kořene. Proces iterace může být nekonečný, proto se v praxi používá zastavovací podmínka, která může mít mnoho podob. Běžně používaná podmínka je, že rozdíl dvou po sobě jdoucích výsledků iterace ( $x_i$  a  $x_{i+1}$ ) jsou menší než nějaká předem určená hodnota  $\epsilon$ , tedy

$$|x_i - x_{i+1}| < \epsilon. \quad (2.17)$$

Pro SPICE jsou definované jisté proměnné, které ovlivňují  $\epsilon$  pro ukončovací

podmínku a tím rychlost a přesnost výpočtu, popřípadě problém s konvergencí. Mezi ně patří RELTOL (relativní tolerance), VNTOL (absolutní tolerance napětí) a ABSTOL (absolutní tolerance proudu). Pomocí nich se pak určuje hodnota  $\epsilon$  pro napětí

$$\epsilon = RELTOL \cdot \max(x_i, x_{i+1}) + VNTOL \quad (2.18)$$

a pro proud

$$\epsilon = RELTOL \cdot \max(x_i, x_{i+1}) + ABSTOL. \quad (2.19)$$

Ve SPICE je pro řešení operačního bodu (OP) nelineárních obvodů implementována Newtonova metoda, která bude popsána níže [24] [34].

### ■ Newtonova metoda

Newtonova (Newton-Raphsonova) metoda nebo také metoda tečen patří mezi nejpobulárnější iterační metody pro řešení nelineárních rovnic. Vychází z Taylorova polynomu 1. řadu, což je aproximace funkce tečnou (lineari-zace nelineárního problému). Obecný Taylorův polynom 1. řadu funkce  $f(x)$  v bodě  $[x_i, f(x_i)]$  vypadá

$$T_1(x) = f(x_i) + f'(x_i)(x - x_i) \quad (2.20)$$

Pro  $i$ -tou iteraci  $x_i$  se sestrojí tečna k funkci  $f(x)$  v bodě  $[x_i, f(x_i)]$ . Průsečík této tečny s osou  $x$  se označí jako  $x_{i+1}$ , který je vypočítán položením rovnice 2.20 rovno nule, tedy

$$0 = f(x_i) + f'(x_i)(x - x_i). \quad (2.21)$$

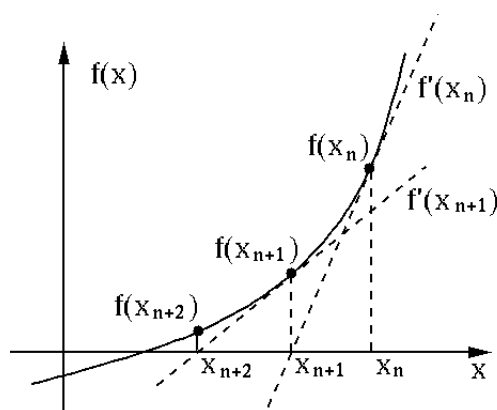
Vyjádřením  $x$ , které v tomto případě označuje  $x_{i+1}$  je výsledkem vztahu pro výpočet  $i + 1$  aproximace kořene  $x_{i+1}$  funkce  $f(x)$ . Čímž vznikla výsledná formulace Newtonovy metody

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}. \quad (2.22)$$

U této metody není implicitně zaručena konvergence, pro zaručení konvergence musí být splněny následující podmínky

$$\begin{aligned} f(a) \cdot f(b) &< 0, \\ f'(x) &\neq 0, \\ f''(x) &\neq 0, \end{aligned}$$

kde  $(a, b)$  je interval, na kterém se hledá kořen funkce  $f(x)$ . Dále pro tuto metodu je klíčová existence první derivace a její hodnota nesmí být nulová. Konvergenci může také ovlivnit volba počáteční aproximace kořene (výchozí  $x_0$ ). Volí se bod  $c \in \{a, b\}$ , pro který platí  $f(c) \cdot f''(c) > 0$ . Na obrázku 2.2 je graficky znázorněn postup výpočtu kořene nějaké konkrétní funkce  $f(x)$  [24] [2] [25].



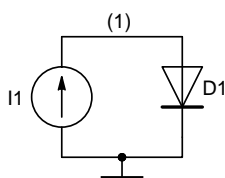
**Obrázek 2.2:** Grafické znázornění Newtonovy metody [40].

Praktické použití Newtonovy metody bude ukázáno na obvodu, který je na obrázku 2.3. Jedná se o jednoduchý obvod s diodou buzenou proudovým zdrojem. Dioda jako nelineární prvek je popsána Shockleyho rovnicí

$$I_D = I_0(e^{\frac{U_D}{U_T}} - 1). \quad (2.23)$$

Pomocí metody uzlových napětí lze pro tento obvod získat rovnici uzlu (1), která vypadá následovně

$$I_0(e^{\frac{U_D}{U_T}} - 1) - I_1 = 0. \quad (2.24)$$



**Obrázek 2.3:** Obvod pro ilustraci Newtonovy metody.

Tímto je dána rovnice, pro kterou je třeba nalézt kořeny. Obecně se to dá zapsat jako  $f(U_D) = 0$ . Jako konkrétní hodnoty známých proměnných byly zvoleny  $I_0 = 10^{-14}$  A,  $U_T = 0.026$  V a  $I_1 = 1$  mA.

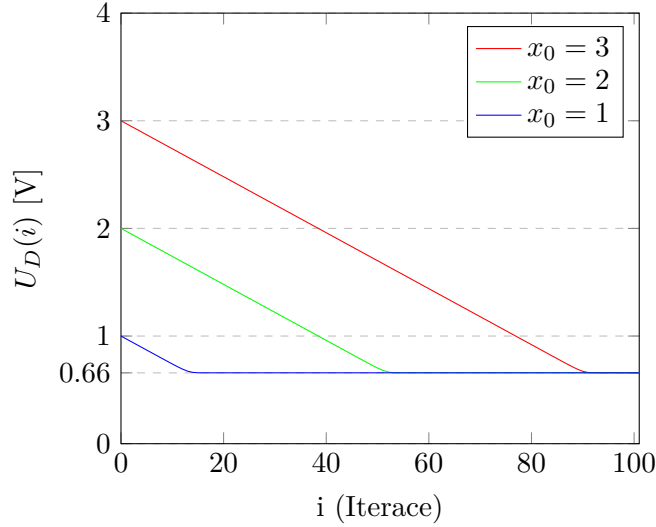
Pro Newtonovu metodu je potřeba znát první derivaci

$$f'(U_D) = \frac{I_0 e^{\frac{U_D}{U_T}}}{U_T}. \quad (2.25)$$

Dosazením první derivace do rovnice 2.22 a po provedení pár algebraických úprav, vznikne výsledná formule pro  $U_{D(i+1)}$ , která bude vypadat

$$U_{D(i+1)} = U_{D(i)} + \frac{U_T \left( (I_0 + I_1) e^{-\frac{U_{D(i)}}{U_T}} - I_0 \right)}{I_0}. \quad (2.26)$$

Po vyčíslení vztahu 2.26 pro danou iteraci lze pozorovat jak napětí  $U_D$  konverguje přibližně k hodnotě 0.66 V, což je typická hodnota propustně polarizované křemíkové diody. To je znázorněno na obrázku 2.4, kde pro tři různé startovací hodnoty  $x_0$  je vidět, po kolika iteracích se hodnota  $U_D$  přiblíží k 0.66 V.



**Obrázek 2.4:** Napětí  $U_D$  v závislosti na iteraci při různých  $x_0$ .

### ■ Newtonova metoda pro řešení soustav nelineárních rovnic

Ve většině případů není obvod tak jednoduchý, jako v předchozím příkladě a vede na soustavu nelineárních rovnic. V tom případě je nutné použít zobecněný vzorec Newtonovy metody s vektory a vektorovými funkcemi

$$\vec{x}_{i+1} = \vec{x}_i - \mathbf{F}^{-1}(\vec{x}_i) \cdot \vec{F}(\vec{x}_i), \quad (2.27)$$

kde  $\vec{x}$  je vektor neznámých o  $n$  prvcích, figurující ve vektorové funkci  $\vec{F}(\vec{x})$ .  $\mathbf{F}(\vec{x})$  je matice parciálních derivací též zvaná jako Jacobiho matice (má význam derivace vektorové funkce  $\vec{F}(\vec{x})$ )

$$\mathbf{F}(\vec{x}) = \begin{bmatrix} \frac{\partial F_1}{\partial x_1} & \cdots & \frac{\partial F_1}{\partial x_n} \\ \vdots & \vdots & \vdots \\ \frac{\partial F_n}{\partial x_1} & \cdots & \frac{\partial F_n}{\partial x_n} \end{bmatrix} \quad (2.28)$$

Ze zápisu 2.27 vyplývá, že se pro  $\mathbf{F}(\vec{x})$  musí vypočítat inverzní matice, což je výpočetně náročné. Proto se využívá spíše alternativy 2.29 [9].

$$\mathbf{F}(\vec{x}_i) \cdot \vec{x}_{i+1} = \mathbf{F}(\vec{x}_i) \cdot \vec{x}_i - \vec{F}(\vec{x}_i) \quad (2.29)$$

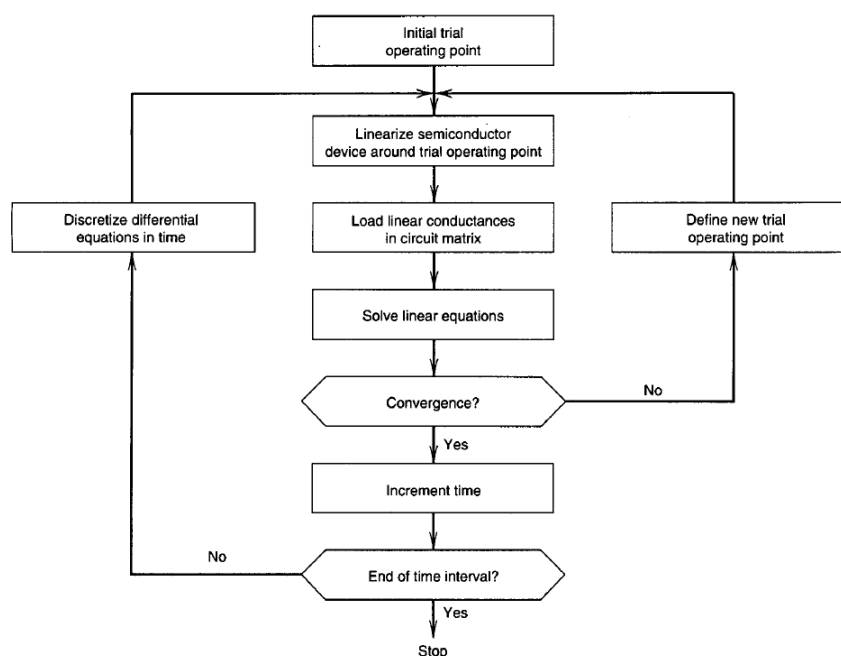
### 2.1.4 Řešení obvodů v časové oblasti

Řešení obvodů v časové oblasti je postup, při kterém se vyhodnocuje stav obvodu v diskrétně daných časových bodech. Jedná se o posloupnost kvazistatických řešení. Výpočet obvodu v časové oblasti je nejkomplexnější proces analýzy elektrických obvodů, který v sobě zahrnuje výpočet soustavy obyčejných diferenciálních rovnic (ODE). Ty jsou numerickou integrací převedeny na soustavu nelineárních rovnic, dále linearizované Newtonovou metodou a nakonec vyřešené pomocí GEM (resp. LU rozkladem). Postup výpočtu implementovaného ve SPICE je ukázán na obrázku 2.5.

V první řadě je potřeba určit počáteční bod analýzy, tj. pro čas  $t=0$ . To lze zařídit buďto výpočtem operačního bodu obvodu nebo nastavením počátečních podmínek, o této problematice více v kapitole 2.2.4 (Transientní analýza).

Pro každý časový bod analýzy (vnější smyčka) se provádí linearizace nelineárních součástí v aktuálním pracovním bodě a řeší se soustava lineárních rovnic. Pokud nelze nalézt řešení, definuje se nový výchozí bod (vnitřní smyčka).

Při volání transientní analýzy je definován časový krok (kapitola 2.2.4). Nicméně se jedná pouze o doporučení pro simulátor. Ve výsledku se časový krok odvíjí od hodnoty LTE (local truncation error), což je chyba numerické metody integrace, která je vypočítávána v každé iteraci (časovém bodě). Pokud je LTE v definovaných mezích, časový krok se nemění, v opačném případě je časový krok upraven, pro lepší přesnost [34].



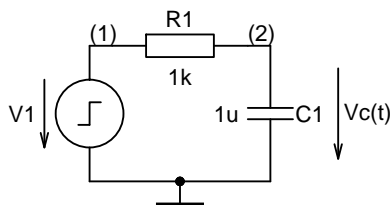
Obrázek 2.5: Algoritmus řešení transientní analýzy ve SPICE [34].



Induktory a kapacitory (setrvačné prvky) z hlediska obvodových rovnic s časově závislými veličinami, přemění rovnice na diferenciální (ODE). To je dáno vztahem proudu a napětí těchto prvků. Pro induktor je napětí  $u_L$  derivací proudu  $i_L$  a pro kapacitor je proud  $i_C$  derivací napětí  $u_C$  viz následující vztahy

$$\begin{aligned} u_L &= L \frac{di_L(t)}{dt} \\ i_C &= C \frac{du_C(t)}{dt}. \end{aligned} \quad (2.30)$$

Řešení obvodů v časové oblasti bude prezentováno na obvodu 2.6, který bude nejprve spočítán analyticky se symbolickými hodnotami. Později až budou představeny numerické metody pro výpočet diferenciálních rovnic. Bude pomocí tohoto obvodu ilustrován princip a výsledek numerického výpočtu.



**Obrázek 2.6:** Obvod pro ilustraci numerického řešení diferenciálních rovnic.

Obvodem je jednoduchý RC článek, který je buzený jednotkovým skokem napětí  $V_1$ . Metodou uzlových napětí se získá rovnice pro uzel (2) respektive napětí  $u_c(t)$

$$\frac{u_c(t) - V_1}{R_1} + C_1 \frac{du_c(t)}{dt} = 0. \quad (2.31)$$

Pro další požití v numerických metodách bude výhodné si přepsat rovnici do tvaru

$$\dot{u}_c(t) = \frac{1}{R_1 C_1} (V_1 - u_c(t)). \quad (2.32)$$

Nemá cenu zde uvádět postup řešení takto jednoduché diferenciální rovnice, není to ani předmětem této práce. Podstatný je až výsledek. Zkušený obvodář už jen při pohledu na takovýto obvod dokáže určit, jak bude výsledek vypadat

$$u_c(t) = V_1 (1 - e^{-t/R_1 C_1}). \quad (2.33)$$

Jde tedy o průběh napětí na kapacitoru, který je nabíjen přes rezistor  $R_1$ , na hodnotu  $V_1$  (konkrétně pak na hodnotu 1V, to ale bude ukázáno dále).

SPICE používá tři numerické metody pro výpočet ODE. O nich bude stručně řečeno na následujících stránkách.

## Eulerova metoda

Eulerova metoda je nejjednodušší metodou řešení diferenciálních rovnic. Je to jednokroková metoda prvního řádu. Existují dvě základní varianty, které odpovídají situaci, kdy se integrál odhaduje pomocí jednostranného Riemannova součtu zleva nebo zprava (obdélníková aproximace integrálu).

Dopředná Eulerova metoda (FE - forward Euler). Jde o explicitní metodu ( $y_{n+1}$  je vyjádřeno pomocí  $y_n$ ). Vyplývá z levého Riemannova součtu. Nejjednodušší odvození je z definice derivace (dopředné)

$$\frac{dy(t)}{dt} = \dot{y} = \lim_{h \rightarrow 0} \frac{y(t+h) - y(t)}{h} \approx \frac{y(t+h) - y(t)}{h}, \quad (2.34)$$

kde  $h = t_{n+1} - t_n$  a po vyjádření  $y(t+h)$  vzniká výsledný vztah pro dopřednou Eulerovu metodu, která mimo jiné má význam Taylorova polynomu prvního řádu

$$y(t+h) = y(t) + h\dot{y}(t). \quad (2.35)$$

Jako vhodnější zápis se bude jevit tento

$$y_{n+1} = y_n + h\dot{y}_n. \quad (2.36)$$

Zpětná Eulerova metoda (BE - backward Euler). Jde o implicitní metodu ( $y_{n+1}$  je vyjádřeno pomocí  $y_n$  i  $y_{n+1}$ ). Vyplývá z pravého Riemannova součtu a vztah pro  $y_{n+1}$  je

$$y_{n+1} = y_n + h\dot{y}_{n+1}. \quad (2.37)$$

Zpětná Eulerova metoda je součástí metody zpětných diferencí (gear 1) popsaných níže [2] [24].

Dopředná Eulerova metoda bude nyní použita pro numerický výpočet rovnice 2.31, kde neznámá je  $u_c(t)$ . Dosazením do 2.36 vzniká

$$u_{c(n+1)} = u_{c(n)} + h\dot{u}_{c(n)}, \quad (2.38)$$

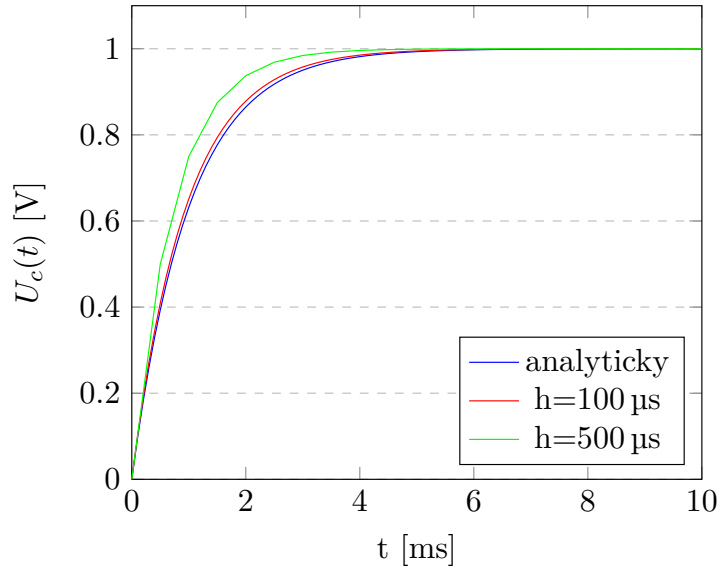
do níž se dosadí přeepsaný tvar 2.32 a výsledkem je finální výraz

$$u_{c(n+1)} = u_{c(n)} + \frac{h}{R_1 C_1} (V_1 - u_{c(n)}). \quad (2.39)$$

Dále zbývá už jen dosadit konkrétní hodnoty, kde časový krok  $h$ , lze zvolit libovolně. Byl tedy proveden výpočet pro dvě různá  $h$ , a to  $h = 100 \mu\text{s}$  a  $h = 500 \mu\text{s}$ . Zbylé hodnoty jsou

$$\begin{aligned} R_1 &= 1 \text{ k}\Omega, \\ C_1 &= 1 \mu\text{F}, \\ V_1 &= 1 \text{ V}, \\ u_{c(0)} &= 0 \text{ V}. \end{aligned} \quad (2.40)$$

Výsledek dosazení a proběhnutí iterací do času  $t = 10$  ms je v grafu 2.7. Pro porovnání byl zobrazen vyčíslený, analyticky vypočítaný průběh ze vztahu 2.33. Je zřejmé, že čím menší časový krok, tím se bude numerické řešení více přibližovat analytickému.



**Obrázek 2.7:** Numerické řešení diferenciální rovnice 2.31 Eulerovou metodou.

### ■ Lichoběžníková metoda

Lichoběžníková metoda pro řešení diferenciálních rovnic se též nazývá Heunova nebo Trapezoidal rule (od toho možnost **trap** ve SPICE). Jde o jednokrokovou metodu druhého řádu, která vychází z lichoběžníkové metody výpočtu integrálů. Jedná se také o implicitní metodu, kde  $y_{n+1}$  figuruje na obou stranách rovnice. Pro výpočet  $y_{n+1}$  z  $y_n$  je třeba řešit obvykle nelineární rovnici, což přináší jisté komplikace, ale za cenu vyšší přesnosti oproti Eulerově metodě.

Jak bylo zmíněno tato metoda vychází z lichoběžníkové metody, tedy pro určitý integrál z  $\dot{y}(t)$  platí

$$\int_{t_n}^{t_{n+1}} \dot{y}(t) dt = \frac{h}{2} (\dot{y}(t_{n+1}) + \dot{y}(t_n)). \quad (2.41)$$

Tento integrál je možné také zapsat Newton-Leibnizovou formulí

$$\int_{t_n}^{t_{n+1}} \dot{y}(t) dt = y(t_{n+1}) - y(t_n), \quad (2.42)$$

z čehož vyplývá, že se obě pravé strany rovnají (přibližně). Zápísem do jedné rovnosti vzniká

$$y(t_{n+1}) - y(t_n) = \frac{h}{2} (\dot{y}(t_{n+1}) + \dot{y}(t_n)). \quad (2.43)$$

Přeskupením a zápisem v jiné formě vzniká výsledný vztah pro lichoběžníkovou metodu

$$y_{n+1} = y_n + \frac{h}{2} (\dot{y}_{n+1} + \dot{y}_n) \quad (2.44)$$

Porovnáním tohoto vztahu se zpětnou a dopřednou Eulerovou metodou, lze zjistit, že jde o jakousi kombinaci obou dvou. Přesněji řečeno jde o zprůměrování směrnic  $\dot{y}_{n+1}$  a  $\dot{y}_n$  [2] [24].

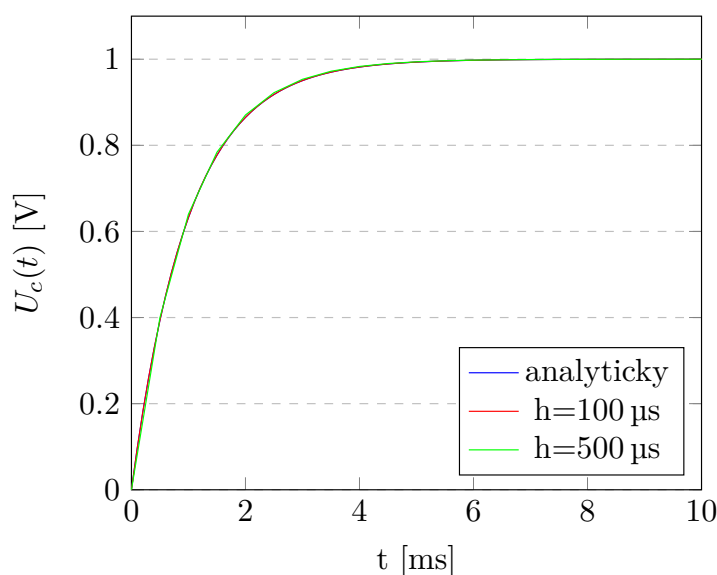
Lichoběžníková metoda bude opět demonstrována na rovnici 2.31. Vyjde se z definičního vztahu 2.44, a za  $y_n$  se dosadí  $u_{c(n)}$ . Za  $\dot{y}$  se dosadí  $\dot{u}_c$  z rovnice 2.32. Z toho vzejde následující rovnost

$$u_{c(n+1)} = u_{c(n)} + \frac{h}{2} \left( \frac{1}{R_1 C_1} (V_1 - u_{c(n+1)}) + \frac{1}{R_1 C_1} (V_1 - u_{c(n)}) \right), \quad (2.45)$$

ze které je potřeba vyjádřit  $u_{c(n+1)}$ , jelikož se jedná o implicitní metodu. Finálním tvarem tedy bude

$$u_{c(n+1)} = \frac{u_{c(n)} + \frac{h(2V_1 - u_{c(n)})}{2R_1 C_1}}{1 + \frac{h}{2R_1 C_1}}. \quad (2.46)$$

Následným dosazením těch samých hodnot jako v 2.40 a proběhnutí iterací do času  $t = 10$  ms, pro ty samé časové kroky  $h = 100 \mu\text{s}$  a  $h = 500 \mu\text{s}$ . Vzniká výsledek zobrazený v grafu 2.8. Pro porovnání byl též zobrazen vyčíslený analytický vztah. Porovnáním s Eulerovou metodou, lze na první pohled říci, že výsledek lichoběžníkové metody a analytického výsledku jsou téměř identické pro tento příklad.



**Obrázek 2.8:** Numerické řešení dif. rovnice 2.31 lichoběžníkovou metodou.

## Metoda zpětných diferencí

Metoda zpětných diferencí (BDF - backward differentiation formula) je více-  
kroková metoda prvního a vyšších řádů. Patří k jedné z neefektivnějších  
metod tohoto druhu. Derivaci funkce aproximuje pomocí použití několika  
předchozích hodnot dané funkce, čímž se zvyšuje přesnost aproximace. Výraz  
pro  $y_{n+1}$  obecné vícekrokové metody lze zapsat jako

$$y_{n+1} = \sum_{i=0}^r a_i y_{n-i} + h \sum_{i=-1}^s b_i \dot{y}_{n-i}. \quad (2.47)$$

Pokud bude koeficient  $b_{-1}$  roven nule, jde o explicitní metodu, v opačném  
případě jde o implicitní metodu. Jestliže  $i > 1$ , pak se jedná o vícekrokovou  
metodu (více předchozích bodů bude použito).

Elektronické obvody mají časové konstanty, které se mohou lišit o několik  
řádů. Rovnice představující tyto obvody tvoří tzv. tuhé systémy (stiff sys-  
tems). Neexistuje všeobecně přijímaná definice tuhosti. Pro přiblížení se jedná  
o systémy, pro jejichž vyřešení je potřeba podstatně větší počet bodů s menším  
časovým krokem, než je žádoucí. To mohou být rovnice, které vedou na řešení  
s rychlými změnami. Integrační metody použitelné pro takovéto systémy musí  
být „stiffly stable“. Z dříve zmíněných metod sem patří zpětný Euler (BE),  
lichoběžníková metoda (Trap) a metoda zpětných diferencí (BDF), naopak  
sem nepatří dopředný Euler (FE).

Ve SPICE lze najít metodu zpětných diferencí pod názvem Gear. Jde  
o skupinu šesti formulí pro výpočet  $y_{n+1}$ . Gear 2 až Gear 6 prokázaly dobré  
vlastnosti stability, proto jsou implementovány ve SPICE. Vyšší než šestý řád,  
už nemusí být stabilní. Pro představu, jak tyto formule vypadají jsou zde níže  
uvedeny, ale praktická ukázka na obvodu 2.6 už zde prezentována nebude.  
Důvodem je přesnost, díky níž by na výsledném průběhu nebylo možné vidět  
rozdíl oproti analytickému výsledku [34] [2].

(Gear 1)  $y_{n+1} = y_n + h\dot{y}_{n+1}$ , (zpětný Euler)

(Gear 2)  $y_{n+1} = \frac{4}{3}y_n - \frac{1}{3}y_{n-1} + \frac{2h}{3}\dot{y}_{n+1}$ ,

(Gear 3)  $y_{n+1} = \frac{18}{11}y_n - \frac{9}{11}y_{n-1} + \frac{2}{11}y_{n-2} + \frac{6h}{11}\dot{y}_{n+1}$ ,

(Gear 4)  $y_{n+1} = \frac{48}{25}y_n - \frac{36}{25}y_{n-1} + \frac{16}{25}y_{n-2} - \frac{3}{25}y_{n-3} + \frac{12h}{25}\dot{y}_{n+1}$ ,

(Gear 5)  $y_{n+1} = \frac{300}{137}y_n - \frac{300}{137}y_{n-1} + \frac{200}{137}y_{n-2} - \frac{75}{137}y_{n-3} + \frac{12}{137}y_{n-4} + \frac{60h}{137}\dot{y}_{n+1}$ ,

(Gear 6)  $y_{n+1} = \frac{360}{147}y_n - \frac{450}{147}y_{n-1} + \frac{400}{147}y_{n-2} - \frac{225}{147}y_{n-3} + \frac{72}{147}y_{n-4} - \frac{10}{147}y_{n-5} + \frac{60h}{147}\dot{y}_{n+1}$ ,

## 2.2 Typy analýz

Zde budou uvedeny základní i pokročilé analýzy, se kterými se lze setkat v různých simulačních softwarech. Důraz bude kladen především na SPICE analýzy, jelikož jde o hojně používaný a známý simulátor.

Elektrické obvody lze analyzovat v mnoha různých pohledech. Existuje celá řada analýz, kterými je možné zkoumat chování a vlastnosti obvodu. Nicméně jsou tři základní, ze kterých ostatní analýzy vycházejí nebo je přímo používají.

Prvním základním typem je OP analýza, tedy analýza operačního bodu obvodu. Jedná se o situaci, kdy je obvod ve stacionárním ustáleném stavu (SUS). Je charakterizovaný tím, že všechny obvodové veličiny jsou konstantní (časově nezávislé). Do tohoto stavu obvod dospěje po doznění přechodných jevů. V tomto stavu jsou všechny obvodové veličiny stejnosměrné (mají trvale konstantní hodnotu). Potom napětí na induktorech a proudy kapacitorů, jenž jsou dány časovými derivacemi (viz vztahy 2.30), jsou nulové.

Druhým základním typem je AC analýza. Jde o analýzu ve frekvenční oblasti, kde se předpokládá, že všechny obvodové veličiny jsou harmonickými funkcemi času o téže frekvenci. Dalším předpokladem je linearita obvodu, nebude-li tento předpoklad splněn. Nebude platit tvrzení, že všechny obvodové veličiny jsou harmonickými funkcemi času o téže frekvenci. Nelinearity totiž způsobují harmonické zkreslení, tedy vznik vyšších harmonických složek. Jedná se tedy o harmonický ustálený stav (HUS), do kterého obvod dospěje po doznění přechodných jevů, při buzení harmonickými zdroji.

Analýza obvodu v HUS se neprovádí v časové oblasti. Výhodnější přístup je transformace do komplexní roviny na fázory. Tato transformace se nazývá Steinmetzova a vychází z Eulerova vztahu mezi goniometrickými funkcemi a exponenciální funkcí

$$U_m e^{j(\omega t + \phi)} = U_m (\cos(\omega t + \phi) + j \sin(\omega t + \phi)). \quad (2.48)$$

Rozdělením exponenciální funkce na  $e^{j\phi}$  a  $e^{j\omega t}$  vznikne

$$U_m e^{j(\omega t + \phi)} = U_m e^{j\phi} e^{j\omega t} = \mathbf{U}_m e^{j\omega t}, \quad (2.49)$$

kde  $\mathbf{U}_m$  je námi chtěný fázor, který je reprezentován dvěma parametry. Těmi jsou amplituda  $U_m$  a fáze  $\phi$ . Zbýlý člen  $e^{j\omega t}$  v sobě obsahuje informaci o času ( $t$ ) a úhlové frekvenci ( $\omega$ ). Za předpokladu HUS platí, že všechny obvodové veličiny mají stejnou frekvenci. Není tento člen pro výpočet jednotlivých obvodových veličin podstatný. Jeho podstata se uplatní až v případě zpětné transformace do časové oblasti. Ta se provádí získáním reálné části komplexního čísla

$$u(t) = U_m \cos(\omega t + \phi) = \Re(\mathbf{U}_m e^{j\omega t}), \quad (2.50)$$

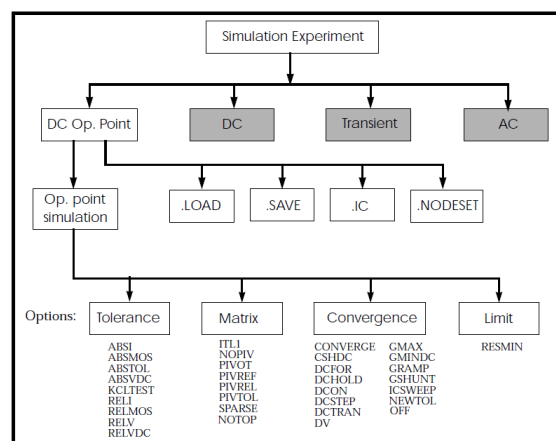
To ale mnohdy není potřeba, protože chtěnými údaji jsou právě amplituda ( $U_m$ ) a fáze ( $\phi$ ).

A třetím základním typem je transientní neboli časová analýza. Obecně lze říci, že obvodové veličiny jsou funkcí času, speciálním případem je pak stacionární ustálený stav, kdy jsou obvodové veličiny konstantní (časově nezávislé). Z tohoto pohledu lze obvody rozdělit na obvody se soustředěnými parametry, ty jsou funkcí pouze času a na obvody s rozprostřenými parametry, ty jsou navíc i funkcí prostorových souřadnic. Obecně jde tedy o problém řešení diferenciálních rovnic, což bylo zmíněno v kapitole 2.1.4. Výsledkem jsou časové průběhy obvodových veličin, které lze dále zkoumat a charakterizovat chování obvodu [8] [20].

### 2.2.1 OP

OP (Operating Point) analýza počítá operační bod obvodu (v SUS). Jde o stěžejní analýzu, na které ve většině případů závisí všechny ostatní. Výsledkem jsou uzlová napětí a proudy napěťovými zdroji a induktory. V druhé řadě jsou pak vypočítány hodnoty specifické pro danou součástku. Například proud a výkon na rezistoru nebo linearizované vlastnosti nelineárních součástek, jako transkonduktance BJT tranzistoru. Všechny kapacitory jsou nahrazeny rozpojeným obvodem a induktory zkratem.

Všechny nelineární obvody jsou počítány iterativními metodami, které potřebují počáteční bod iterace. Ten je defaultně brán nulový. Ve většině případů není třeba tento problém řešit. Najdou se ale obvody, kde po dosažení maximálního počtu iterací, nebylo nalezeno numerické řešení. Pro takovéto případy a nebo pro rychlejší nalezení řešení souží příkaz *nodeset*. Tím lze nastavit počáteční odhad uzlových napětí. Prakticky to SPICE provádí připojením proudového zdroje a paralelní vodivosti (Nortonův teorém). V této konfiguraci se vypočítá pracovní bod a poté se obvod vrátí do původní konfigurace a přepočítá se výsledek, s tím že se vychází z původně vypočítaných hodnot. U zdrojů je podstatný parametr *dc*, není-li definován a je definován časový průběh, bere se hodnota v čase nula [34] [12] [4].



Obrázek 2.9: Struktura OP analýzy [12].

### 2.2.2 DC sweep

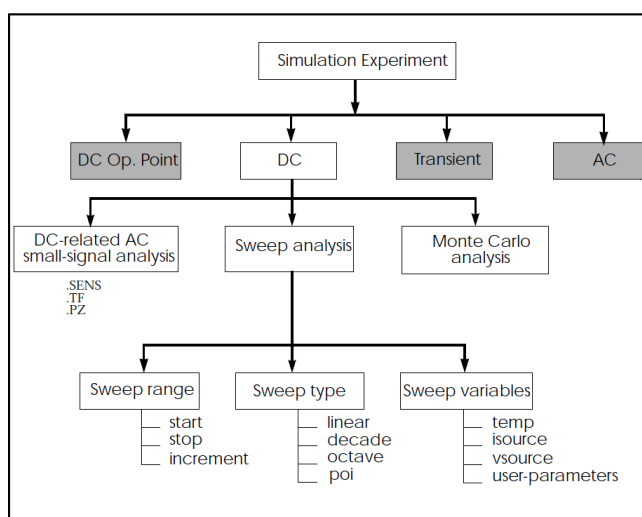
DC analýza také počítá operační bod a to s rozdílem, že nepočítá jeden operační bod, ale celý interval, definovaný parametrem, který se mění (parameter sweep). Dalo by se tomu říkat parametrická OP analýza. Takovým parametrem může být zdroj napětí nebo proudu. To je používáno například pro výpočet volt-ampérových charakteristik. Dalším typem parametru může být teplota, což lze využít pro zjištění teplotní závislosti operačního bodu. Nebo lze měnit parametry součástek případně i jejich modelů.

V praxi je DC analýza definována několika parametry. Konkrétně je to měněný parametr (zdroje, teplota a parametry součástek a modelů). Dále specifikace intervalu hodnot, tedy počátek, konec a buďto krok nebo počet bodů a s tím související rozložení bodů (lineární, dekadické, oktávové nebo logaritmické).

V principu jde o mnohočetné provedení OP analýzy. Není to však stejné, jako kdyby se vícekrát spouštěla OP analýza s různými hodnotami parametru. Důvodem je využití výsledku z předchozího kroku pro použití jako počáteční bod následujícího kroku. Předpokládá se, že krok nebude tak velký a tudíž ani výsledek se nebude o tolik lišit. Takže použitím předchozího výsledku jako počáteční bod Newtonovy metody dojde ke snížení počtu iterací algoritmu a tedy i urychlení celkového výpočtu [34] [12] [4].

Jak bylo zmíněno, uplatnění této analýzy je v zobrazení DC charakteristik jako je například volt-ampérová charakteristika. Také lze sledovat různé nelinearity, saturace, hysteréze, teplotní závislosti a tak dále. Postprocessingem pak je možné z těchto charakteristik dále určovat různé parametry jako zesílení, intervaly linearity a jiné.

Z této analýzy pak vychází další malo-signalové analýzy jako Sens, TF, PZ viz obrázek 2.10.



Obrázek 2.10: Struktura DC analýzy [12].



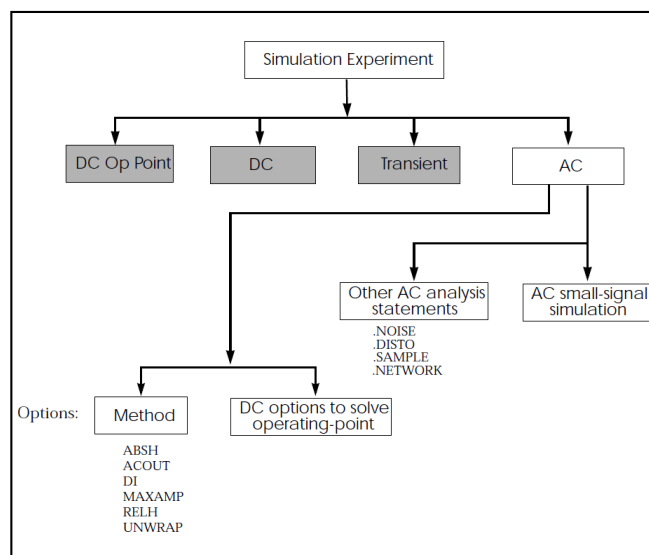
### 2.2.3 AC

AC analýza počítá frekvenční odezvu obvodu v harmonickém ustáleném stavu. Analýza pracuje vždy s lineárním obvodem, není-li lineární, je provedena linearizace v okolí pracovního bodu. Této analýze se také říká malosignálová, protože díky linearizaci jsou nutné signály s amplitudou menší než je tepelné napětí ( $U_{th} = kT/e$ ), aby byl výsledek věrohodný. Důsledkem toho mohou být výsledky v nesmyslně velkých řádech (například v MV). Z hlediska lineárního obvodu je to v pořádku, ale pro reálný obvod je to nemožné.

Výsledkem analýzy jsou napětí a proudy v komplexních číslech (fázorech) na dané frekvenci. AC analýzu lze rozdělit na dva případy. První případ je rozmítání frekvence ve specifikovaném rozsahu. Výsledkem jsou frekvenčně závislé napětí a proudy. Ty lze zobrazit jako přenosovou charakteristiku (amplituda a fáze v závislosti na frekvenci), jako hodograf (v komplexní rovině) nebo pro vysokofrekvenční obvody se používá Smithův diagram. Postprocessingem lze s výsledky dále počítat. Zápornou derivací fáze podle frekvence lze získat skupinové zpoždění. Z přenosové charakteristiky je možné určit zesílení, šířku pásma a jiné.

Druhým případem je analýza v jedné frekvenci, což znamená jedno číslo (komplexní) pro danou veličinu. To lze interpretovat jako číselný výsledek (případně dopočítat amplitudu a fázi) nebo jako fázor v komplexní rovině. Dále lze provádět analýzu v jedné frekvenci při rozmítání jiných parametrů jako teplota, parametry součástek a modelů. To se dá použít pro optimalizaci přenosové charakteristiky respektive šířky pásma, rezonanční frekvence a tak dále.

U této analýzy je pro nezávislé zdroje důležitý parametr  $ac$ , který představuje amplitudu signálu. U zdrojů, které tento parametr nemají, platí, že napěťové jsou zkratovány a proudové rozpojeny [34] [12] [4].



Obrázek 2.11: Struktura AC analýzy [12].

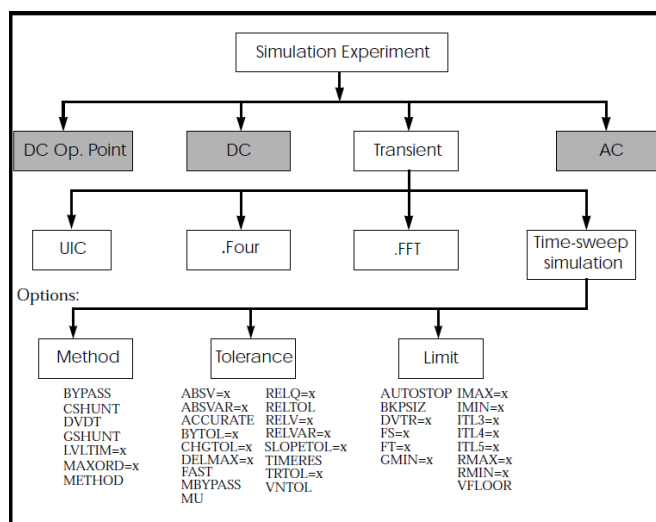
### 2.2.4 Transientní

Transientní analýza vyhodnocuje obvod v časové oblasti. Výsledkem jsou průběhy veličin v závislosti na čase. Ty lze zobrazit klasickým způsobem nebo v módu XY, tedy třeba průběh jednoho napětí na druhém. Tímto způsobem je možné zobrazit například hysterezní křivku. Dále pomocí transientní analýzy se dají pozorovat přechodné děje, zkresení signálu, rychlost přeběhu nebo střední a peak-to-peak hodnota signálu.

Pro spuštění analýzy jsou potřeba minimálně dva parametry. Jedním je koncový čas analýzy a druhým časový krok. Jelikož se výpočet provádí pomocí numerických metod, které kontrolují chybu metody, a od které se odvíjí případné zjemnění kroku. Jedná se pouze o doporučení pro simulátor, jaký krok se má zvolit. Dalším parametrem může být počáteční čas. Není-li zadán, bere se za nulový. Je-li zadán, stejně se simulace počítá z  $t = 0$ , ale výsledky jsou ukládány až od daného času.

Důležitým parametrem při spouštění analýzy je *uic* (use initial conditions), tedy použití počátečních podmínek. Pokud není *uic* dáno, spočítá se operační bod odvodu a následně se z něho vychází (jako hodnota  $t = 0$ ). Je-li *uic* zadán, berou se v úvahu uživatelem zadané počáteční podmínky (IC). Nejsou-li zadány, považují se jako nulové. IC na rozdíl od *nodeset* neovlivňují výpočet pracovního bodu (konvergenci), ale slouží pouze jako počáteční stav transientní analýzy [34] [12] [4].

Na tuto analýzu navazují spektrální analýzy typu FFT (Fast Fourier transform) a Fourier viz obrázek 2.12.



Obrázek 2.12: Struktura transientní analýzy [12].

### ■ 2.2.5 Fourierova analýza

Periodický signál může být rozložen na řadu sinusových/kosinusových složek, které jsou násobky základní frekvence signálu. Tyto složky signálu se také označují jako spektrální nebo harmonické složky. Jinými slovy, periodický signál může být reprezentován Fourierovou řadou.

Obecně nelineární obvody způsobují zkreslení signálu, tedy vznik vyšších harmonických. Pomocí Fourierovy analýzy lze toto zkreslení zkoumat, jak graficky, tak i kvantitativními parametry. Tím může být například THD (total harmonic distortion), což je poměr součtu výkonů všech vyšších harmonických složek ku výkonu základní harmonické. SPICE tuto hodnotu standardně vypisuje [34].

### ■ 2.2.6 Spektrální analýza

Fourierova analýza je zaměřena na periodické signály. Běžnější analýza ve frekvenční oblasti pro obecně neperiodické signály se provádí pomocí Fourierovy transformace. Zpravidla se používá výpočetní algoritmus FFT (Fast Fourier transform), který efektivně převede časový signál do frekvenční oblasti.

Nutností pro výpočet spektra pomocí FFT je signál s ekvidistantními časovými vzorky. SPICE obecně toto nezaručuje, protože časový krok se odvíjí od numerické chyby integrace. Je tedy potřeba provést linearizaci resp. interpolaci signálu.

Dalším problémem při spektrální analýze je konečnost signálu, kde na začátku a na konci vzniká diskontinuita. Výsledkem jsou vysokofrekvenční složky, které nejsou přítomny v původním signálu. Tyto frekvence mohou být mnohem vyšší než Nyquistova frekvence a jsou aliasingem mezi 0 a polovinou vzorkovací frekvence. Spektrum, které se získá pomocí FFT, tedy není skutečné spektrum původního signálu, ale „rozmazaná“ verze. Tento jev se nazývá prosakování ve spektru. Prosakování lze předejít použitím celočíselného počtu period (je-li signál periodický) nebo vynásobením okenní funkcí (Blackman, Hamming, Hann a jiné) [4].

### ■ 2.2.7 Citlivostní analýza

Citlivostní analýza počítá citlivost zadané výstupní veličiny na parametrech obvodu. V principu jde o to, že výstupní veličina je dána nějakým vzorcem, obsahující parametry obvodu. Parciální derivací výstupní veličiny podle jednotlivých parametrů vzniká soubor hodnot, které představují citlivosti výstupní veličiny na dané parametry. Pro parametry s hodnotou nula se citlivost nepočítá. Výstupní hodnoty citlivostí se mohou udávat v procentech nebo v absolutní změně výstupní veličiny na jednotkovou změnu parametru [34].

Citlivostní analýza je vhodná k zjišťování, do jaké míry mají vliv tolerance hodnot součástek na danou sledovanou veličinu obvodu [20].

### 2.2.8 PZ analýza

PZ (Pole-Zero) analýza počítá póly a nuly přenosové funkce obvodu, kde póly jsou kořeny jmenovatele a nuly kořeny čitatele dané přenosové funkce. Obvod je nejprve linearizován v pracovním bodě. Při analýze PZ se s nezávislými zdroji napětí zachází trochu jinak, než při AC analýze. Nezávislý zdroj napětí, který nemá parametr *acmag*, představuje zkrat v PZ analýze. Pokud je však zadán parametr *acmag*, zdroj je odstraněn z obvodu. To se liší od analýzy AC, kde takový zdroj není odstraněn, protože představuje komplexní zdroj signálu.

Důvody jsou následující. Nezávislé zdroje napětí bez parametru *acmag* dodávají energii analyzovanému obvodu. Tyto zdroje by měly být zastoupeny zkratem, když je obvod linearizován. Nezávislé zdroje napětí, které mají parametr *acmag*, jsou zdroje signálu. Takový zdroj je třeba před výpočtem přenosové funkce z obvodu odstranit, protože je (téměř vždy) umístěn na vstupu obvodu. Pokud takový zdroj není odstraněn (a nahrazen přerušným obvodem), ale nahrazen zkratem, způsobí to zkrat na vstupu a zabrání se vyhodnocení funkce přenosu.

PZ analýza je užitečná metoda pro studium chování lineárních časově invariantních (LTI) obvodů a lze ji použít při návrhu analogových obvodů, jako jsou zesilovače a filtry. Může být použita pro stanovení stability z polohy pólů a nul dané přenosové funkce [4].

## Kapitola 3

### Simulátory

#### 3.1 Typy simulátorů

Typy simulátorů se liší tím, pro jaké obvody jsou určeny. Jsou dvě základní kategorie: analogové obvody a digitální obvody. Existují také simulátory, které dokáží pracovat jak s analogovými, tak digitálními obvody. Těm se říká smíšené simulátory. Jako další speciální druhy simulátorů se můžou považovat symbolické a semisymbolické.

##### 3.1.1 Simulátory analogových obvodů

Analogové simulátory se zaměřují na lineární a nelineární chování obvodů nejen v časové oblasti. Řeší se Kirchhoffovy zákony, jejichž výsledkem jsou průběhy napětí a proudů daných součástí. Řeší se soustavy obecně nelineárních rovnic. Pro časové odezvy to jsou soustavy obyčejných diferenciálních rovnic. Používají se různé numerické metody a maticové výpočty, které jsou časově náročnější.

Obvody je možné zkoumat třemi základními analýzami. První je DC analýza, ve které se zanedbávají reakční prvky. Počítá se operační bod obvodu, případně jeho závislost na rozmítání určitého prvku obvodu. Další je AC analýza, která zkoumá chování obvodu v závislosti na frekvenci. Poslední je transientní analýza, jejíž výsledkem je časová odezva obvodu na budící zdroje. Zdroje mohou být konstantní, sinusové, pulsní a jiné speciální zdroje.

Typickým zástupcem analogových simulátorů je SPICE, který byl původně určen pro návrh integrovaných obvodů (IC). Dnes je hojně používán nejen pro návrh IC, ale i pro diskrétní obvody a obvodové moduly [39].

##### 3.1.2 Simulátory digitálních obvodů

Simulace digitálních obvodů se od analogových liší v mnoha pohledech. Hlavní rozdíl je v tom, že se nemusí řešit Kirchhoffovy zákony, protože u digitálních obvodů není třeba řešit konkrétní hodnoty napětí a proudů. Podstatné jsou logické úrovně a jejich propagace dál do obvodu. Logické úrovně jsou většinou reprezentovány dvěma hodnotami 0 a 1, které jsou dány nějakými intervaly hodnot napětí popřípadě proudů. Řeší se hlavně změny logických úrovní



Symbolické výsledky lze získat například ze simulátorů TINA [35], SNAP [31] nebo PraCAn [28].

Krátce jen o PraCAnu, protože tato práce s ním souvisí. PraCAn je zkratkou Prague Circuit Analyser, a jedná se o knihovnu pro Maple [16], umožňující symbolickou analýzu elektrických obvodů. Více je zmíněno v předchozí práci [1] nebo v [28].

## 3.2 Použitelné simulátory

Cílem této práce je zabývat se analýzami v online editoru elektrických obvodů (GEEC). Není účelem naprogramovat vlastní simulační software, ale nalézt vhodný program, který poběží na straně serveru. Na základě požadavků ze strany editoru se bude tento program spouštět a provádět dané analýzy. V současné době se numerické analýzy provádí ve SPICE OPUS, jak bylo zmíněno v pracích [11] a [1].

### 3.2.1 WINSPICE

WINSPICE je univerzální program pro simulaci obvodů pro nelineární DC, nelineární přechodné a lineární AC analýzy. Obvody mohou obsahovat rezistory, kapacitory, induktory, vázané induktory, nezávislé zdroje napětí a proudu, čtyři typy závislých zdrojů, bezeztrátová a ztrátová přenosová vedení (dvě samostatné implementace), přepínače, stejnoměrně distribuovaná RC vedení a pět nejběžnějších polovodičových součástek: diody, BJT, JFET, MESFET a MOSFET.

WINSPICE3 je založen na SPICE3F41, který byl zase vyvinut z SPICE2G.6. Zatímco WINSPICE3 je vyvíjen tak, aby zahrnoval nové funkce. Nadále podporuje funkce a modely, které se stále hojně používají v programu SPICE2.

Samotný WINSPICE neobsahuje schematický editor, je tedy ideální pro spouštění z příkazové řádky. Na druhou stranu jde o placený software, takže není volně dostupný [38].

### 3.2.2 HSPICE

HSPICE je průmyslovým standardem pro přesnou simulaci obvodů. Nabízí MOS modely certifikované výrobcí polovodičových součástek. Obsahuje nejmodernější simulační a analytické algoritmy. Díky rozsáhlému využití v simulaci integrity signálu čipů, pouzder, desek plošných spojů, charakterizaci buněk a pamětí, návrhu integrovaných obvodů je HSPICE nejpopulárnějším, důvěryhodným a komplexním obvodovým simulátorem na trhu.

Nicméně jedná se o komerční produkt s nezanedbatelnou cenou. Pro výukové a běžné potřeby je zbytečně sofistikovaný [29].





Následně je pak možné jedním parametrem měnit hodnoty více součástí najednou.

Dalšími výhodami jsou příkazy *measure* (postprocessing odsimulovaných výsledků), *fft* (má přímo implementovanou funkci FFT, a ne jen *spec*), umožňuje výpočet PSD (power spectral density) a v neposlední řadě *compose*, který dokáže vygenerovat vektor s daným rozložením bodů - lineárně, oktávově nebo logaritmicky). Program je také dobře zdokumentován ve volně dostupném PDF na internetu [23].

NGSPICE pro své převažující výhody je uvažován jako vhodný kandidát pro numerické analýzy v GEEC. Dobrým ukazatelem je také to, že ho využívají jiné známé softwary jako Altium designer, Eagle, KiCad nebo online simulátor PartSim (viz kapitola 3.3.4).

### ■ 3.2.5 Shrnutí

V tabulce 3.1 byla shrnuta důležitá kritéria, která se musí zvážit při volbě vhodného simulátoru pro numerické analýzy GEEC. Zcela zásadním požadavkem je neplacený software. To splňují NGSPICE a SPICE OPUS. Zda je daný software open-source, není až tak důležité. Podstatnější je dostupnost volné dokumentace, která je pro SPICE OPUS ve formě knihy, nikoliv online PDF. Dobré je zvážit možnosti provádět pokročilejší operace, které základní SPICE nenabízí. NGSPICE se zdá být ideální volbou pro numerické analýzy GEEC.

**Tabulka 3.1:** Stručné shrnutí vlastností použitelných simulátorů.

	NGSPICE	SPICE OPUS	WINSPICE	HSPICE
zdarma	ano	ano	ne	ne
open-source	ano	ne	ne	ne
volná dokumentace	ano	ne	ano	ano
pokročilé funkce	ano	ano	ne	ano



### ■ 3.3.4 PartSim

PartSim je bezplatný a snadno použitelný online simulátor obvodů, který běží ve webovém prohlížeči a umožňuje komukoliv navrhnout, simulovat a sdílet obvod. Obsahuje také databázi s velkým počtem součástek od různých výrobců.

Opět v pozadí běží NGSPICE pro simulaci obvodů. Implementovány jsou pouze základní analýzy OP, DC, AC a transientní. Bohužel není možné dále pracovat s výsledky, jen lze zobrazovat pevně dané výstupy. Je možné zobrazit vstupní skript do NGSPICE i výstupní log [27].

### ■ 3.3.5 CircuitLab

CircuitLab je webový schématický editor a simulátor, který byl vytvořen studenty Massachusettského technologického institutu (MIT) pro podporu výuky elektrotechnických obvodů. Poskytuje online nástroje v prohlížeči pro schématické kreslení a simulaci obvodů. Tyto nástroje umožňují studentům, amatérům i profesionálním technikům navrhnout a analyzovat analogové i digitální systémy, ještě před vytvořením prototypu. Online schématický editor umožňuje snadno sdílet a diskutovat o návrzích obvodů. Zatímco online simulace obvodů umožňují rychlou iteraci návrhu a zrychlené učení o elektronice.

Implementovány jsou OP, DC, AC a transientní analýza. Simulované grafy a výstupní výsledky lze exportovat jako soubor CSV pro další analýzu.

Tento software je placený. Jsou tři možnosti předplatného, z čehož je jedna určena pro školy [26].

### ■ 3.3.6 TinaCloud

TINACloud je online webová verze populárního softwaru TINA, který nyní běží v prohlížeči bez jakékoli instalace a kdekoli na světě. Je to výkonný průmyslový online simulátor obvodů, který umožňuje analyzovat a navrhovat analogové, digitální, VHDL, Verilog, Verilog A, MCU a smíšené elektronické obvody včetně SMPS, RF, komunikačních a optoelektronických obvodů, testovat aplikace s mikrokontroléry v prostředí smíšených obvodů.

TINACloud podporuje většinu funkcí jako TINA. Základem je schématický editor. Umožňuje mnoho analýz jako DC, AC, transientí, Fourierovu, Šumovou a jiné dalších. Dokáže také zobrazit symbolickou přenosovou funkci lineárních obvodů v DC, AC a transientním režimu.

Stejně jako TINA je i TINACloud placený software, což také odpovídá kvalitě a možnostem simulátoru [35].

### 3.3.7 Multisim Live

Multisim Live je online verze desktopové aplikace Multisim od společnosti National Instruments. Jde opět o placený simulátor. Umožňuje uživatelům využívat stejnou simulační technologii, jakou dnes používají akademické instituce i průmyslový výzkum. Používat ji kdekoli, kdykoli a na jakémkoli zařízení.

Co se analýz týče, nemá takové možnosti jako desktopová verze. K dispozici jsou základní analýzy OP, DC, AC a transientní [19].

### 3.3.8 GEEC

Posledním zmíněným online webovým simulátorem je GEEC. Jelikož tato aplikace je hlavním předmětem této práce, bude popsána trochu podrobněji než předchozí. Nicméně ne zase tak podrobně, protože to již bylo popsáno pracích [11] a [1].

Jedná se o webovou aplikaci napsanou v jazyce JavaScript, která umožňuje kreslení součástek i celých obvodů. Ty lze dále ukládat, načítat, editovat a exportovat do různých formátů jak vektorových (PDF, PostScript), tak rastrových (JPG, PNG), popřípadě do  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ u.

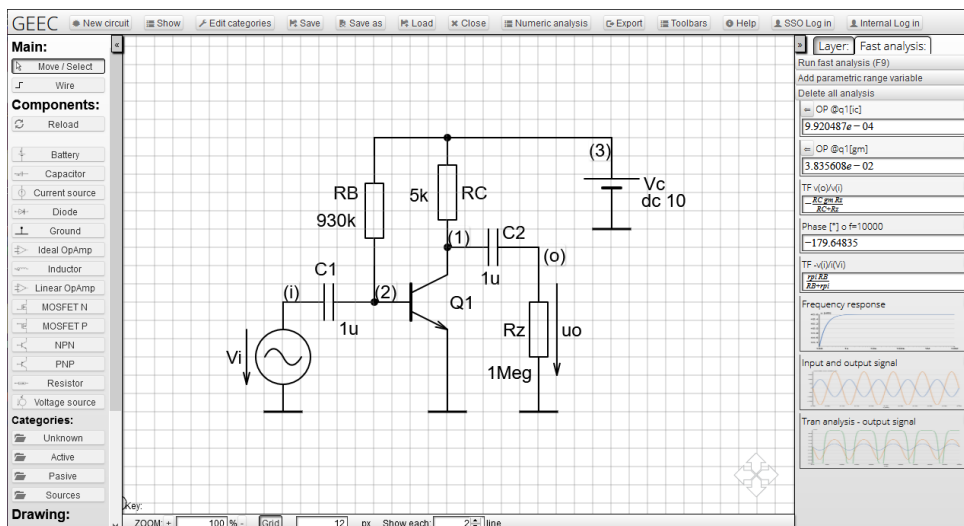
Vytvořené obvody lze také analyzovat pomocí zaintegrováných numerických (SPICE) a symbolických (PraCAN) simulátorů. Výsledky je možné zobrazit ve formě symbolických a semisymbolických výrazů, dále jako numerické hodnoty nebo v grafické podobě. Tyto výsledky se dají uložit do pravé lišty editoru, která se nazývá *Fast analysis*. K dispozici jsou základní typy analýz OP, DC, AC a transientní, podrobně popsané v [1]. Dále přibyly nové pokročilejší analýzy, kterými se mimo jiné bude dále zabývat tato práce. Opomenout se nesmí ani možnost analýzy spínaných obvodů v analyzátoru PraCAN.

Výhodou této a obecně všech ostatních webových aplikací je, že není požadována instalace a lze k ní přistupovat odkudkoli s přístupem k internetu.

Velké uplatnění tato aplikace našla při výuce elektrických obvodů. Je k dispozici databáze obvodů, která čítá stovky zanalyzovaných příkladů s uloženými výsledky. To lze považovat za velice dobrý interaktivní výukový materiál, kde si student může daný příklad zanalyzovat na papíře a výsledky porovnat se simulátorem. Dále si může editovat parametry obvodu i analýz a zkoumat chování daného obvodu. K příkladům je možné přistupovat pomocí speciálních odkazů. To přináší příležitost v prezentacích a elektronických studijních materiálech vložit například obrázek s obvodem a odkazem na daný příklad v aplikaci. Snadno se pak lze dostat pouhým kliknutím k obvodu otevřenému v editoru, a zjistit mnohé další informace, které by se do prezentace nevešly.

Využití ale nemusí být pouze pro teorii. Pomocí editoru lze navrhovat i simulovat obvody s reálnými modely součástek. Například pro tranzistory, je možné výběr modelu z databáze, případně vložení vlastního modelu. Pro návrh obvodů je dobrým pomocníkem možnost parametrické analýzy vytvořené a popsané v práci [1].

Na Obrázku 3.1 je pro představu ukázán jak zmiňovaný editor vypadá. Vizuálně se skládá ze čtyřech částí. Levý panel obsahuje součástky a kreslicí nástroje. Horní panel je obecné menu, obsahující hlavní ovládací prvky. Pravý panel je již zmíněná Fast analysis. A nejdůležitější částí je kreslicí plocha, ve které vzniká výsledný obvod.



Obrázek 3.1: Grafický editor elektrických obvodů (GEEC).



## Kapitola 4

### Knihovny pro vizualizaci dat

Pro zobrazení grafických výsledků analýz je potřeba zvolit vhodný nástroj, který splní dané požadavky a výsledná vizualizace bude vypadat na úrovni. Jelikož je GEEC psán v jazyce JavaScript, bude tato rešerše zaměřena na knihovny pro vizualizaci dat určené právě pro JavaScript.

Některé knihovny nabízejí bezplatné verze pro osobní použití, ale za jejich použití v komerčních aplikacích je třeba zaplatit. Níže budou představeny dvě komerční knihovny (amCharts a CanvasJS), ale jinak bude důraz kladen na volné knihovny.

#### 4.1 amCharts

amCharts je jedna z nejzajímavějších vizualizačních knihoven. Její design značně převyšuje konkurenční produkty. Společnost amCharts uvádí jako své klienty Microsoft, Apple, Samsung, Amazon, NASA a mnoho dalších významných společností. Umožňuje vykreslení všech typů grafů, včetně geografických map.

Současná verze 4 byla napsána v TypeScriptu, lze ji ale použít v jakémkoli prostředí kompatibilním s JavaScriptem, tedy v aplikacích TypeScript, React nebo Angular2 a dokonce i ve původním JavaScriptu [15].

#### 4.2 CanvasJS

CanvasJS je další komerční nástroj. Podporuje 30 různých typů grafů a vykreslení napříč zařízeními, včetně iPhone, iPad, Android, Mac a PC. Grafy zahrnují několik dobře vypadajících předloh a jsou 10krát rychlejší než běžné knihovny grafů založené na Flash/SVG. Bezproblémově funguje i s 100 000 a více datovými body. Jejich seznam klientů je také působivý: Apple, Bosch, Siemens, HP, Microsoft, Intel, BMW a mnoho dalších.

Grafy lze snadno integrovat do populárních frameworků jako React, Angular, Vue atd. Knihovna obsahuje více než 30 typů grafů včetně spojnicových, sloupcových, výsečových, prstencových a skládaných. Grafy podporují různé interaktivní funkce, jako jsou popisky, přiblížení, posouvání, animace, události,

procházení a export. Možná je i integrace s různými technologiemi na straně serveru, jako je PHP, Ruby, Python, ASP.Net, Node.JS a Java [3].

## 4.3 D3

D3 (Data-Driven Documents) je jednou z nejpobulárnějších JavaScriptových knihoven nejen pro vizualizaci dat, ale také pro animace, analýzu dat a geografické zobrazování.

D3 umožňuje vázat libovolná data na model DOM (Document Object Model) a poté na dokument použít transformace založené na datech. Například lze D3 použít ke generování tabulky HTML z řady čísel či použitím stejných dat vytvořit interaktivní sloupec grafu SVG s plynulými přechody a interakcí.

D3 není jednoduší framework, který se snaží poskytnout všechny myslitelné funkce. Místo toho řeší podstatu problému: efektivní manipulaci s dokumenty založenými na datech. Tím se zabráni proprietární reprezentaci a poskytuje mimořádnou flexibilitu a odhalí veškeré možnosti webových standardů, jako jsou HTML, SVG a CSS. S minimální režii je D3 extrémně rychlý a podporuje velké datové sady a dynamické chování pro interakci i animaci. Funkční styl D3 umožňuje opětovné použití kódu prostřednictvím rozmanité kolekce oficiálních a komunitně vyvinutých modulů [6].

### 4.3.1 VEGA

D3 je knihovna nižší úrovně. Během raného vývoje se D3 označovalo spíše jako „vizualizační jádro“ než jako toolkit nebo framework. Kromě vlastního designu je D3 zamýšlen jako podpůrná vrstva pro vizualizační nástroje vyšší úrovně. Vega je jedním z nástrojů, který při své implementaci hojně využívá D3.

Vega je deklarativní formát pro vytváření, ukládání a sdílení vizualizačních návrhů. S Vegou jsou vizualizace popsány v JSON formátu a generují interaktivní zobrazení pomocí HTML5 Canvas nebo SVG.

Vega-Lite je vysokoúrovňová gramatika interaktivní grafiky. Poskytuje stručnou, deklarativní syntaxi JSON k vytvoření vizualizací pro analýzu a prezentaci dat. Je také používána celou řadou společností jako Apple, Google, Microsoft, organizací Cern, univerzitami MIT a Berkeley, nebo například deníkem Los Angeles Times [36].

## 4.4 Recharts

Recharts je knihovna grafů vytvořená pomocí React a D3. Jedná se o knihovnu, která umožňuje rychle vytvářet grafy s oddělenými, opakovaně použitelnými komponentami React. Jedním z jeho hlavních silných stránek je, že jeho komponenty jsou postaveny na prvcích SVG s lehkou závislostí na submodulech D3.



Grafy lze přizpůsobovat. Knihovna poskytuje skvělý výkon pro statické grafy a obsahuje vestavěné nástroje pro grafy, jako jsou legendy a popisy. Při práci s více animovanými grafy na stejné stránce s velkými datovými sadami může být zobrazování neplynulé, ale pro většinu případů to bude fungovat [30].

## 4.5 Chart.js

Chart.js je poměrně snadno použitelný a také velmi přizpůsobitelný. Umožňuje vytvářet graf pomocí prvků HTML5 canvas, díky čemuž je vykreslování poměrně rychlé. Knihovna jako taková nabízí podporu pro většinu prohlížečů, protože prvek canvas je podporován ve všech moderních prohlížečích. Výhodou je dobrá integrace s populárními knihovnami JavaScriptu.

Jde o open-source projekt s rozsáhlou dokumentací udržovanou vývojáři. Obsahuje více než osm typů grafů, včetně vestavěných typů grafů [13].

## 4.6 Victory

Victory je sada modulárních grafických komponent pro React a React Native. Díky Victory je snadné začít bez obětování flexibility a vytvořit jedinečnou vizualizaci dat s plně přizpůsobitelnými styly a chováním. Victory používá stejné API pro web a aplikace React Native pro snadné mapování mezi platformami [37].

## 4.7 Google Charts

Google Charts je cloudové řešení business intelligence navržené tak, aby pomohlo týmům vizualizovat data na svých webech ve formě piktogramů, koláčových grafů, histogramů a dalších. Mezi klíčové funkce patří správa obsahu, vlastní řídicí panely, správa datových sad a vizuální zjišťování.

Uživatelé využívající grafy Google mohou využívat interaktivní grafy, které se vykreslují pomocí technologie HTML5 / SVG a poskytují přenositelnost napříč platformami na více zařízeních. Galerie grafů nabízí celou řadu připravených ukázek zhotovených vizualizací k použití, které se mohou dále přizpůsobit podle individuálních požadavků.

Google Charts je zdarma, ale nikoliv open-source (skoro jako každý produkt Google). Grafy jsou ve výchozím nastavení neresponsivní, ale pomocí vlastního kódu lze grafům měnit velikost [14].

### 4.7.1 Shrnutí

V tabulce 4.1 byly shrnuty důležité vlastnosti výše zmíněných grafických knihoven pro JavaScript. Důležitým požadavkem je volné použití a nezávislost na jiných webových technologiích. Protože D3 je velice známá i používaná knihovna a spolu s nástrojem VEGA umožňuje elegantně vytvářet vizualizace dat, byla tedy vybrána pro zobrazování výsledků grafických analýz.

**Tabulka 4.1:** Stručné shrnutí vlastností grafických knihoven.

	zdarma	open-source	závislost na jiné technologii
D3 (VEGA)	ano	ano	ne
amCharts	ne	ne	ne
CanvasJS	ne	ne	ne
Recharts	ano	ano	ano (React)
Chart.js	ano	ano	ne
Victory	ano	ano	ano (React)
Google Charts	ano	ne	ne



## **Část II**

### **Vlastní práce**



## Kapitola 5

### Grafy

Dříve pro zobrazování grafů byla používána knihovna D3. Vzhledem k nutnosti v analýzách i grafech měnit a přidávat nové funkcionality se původně uvažovalo o aktualizaci D3 knihovny na nejnovější verzi. Po provedení rešerše viz kapitola 4, však bylo rozhodnuto o použití Vega-Lite. Ta je sice nadstavbou D3, nicméně její použití vedlo k redukci kódu, protože spousta věcí se řeší automaticky a nemusí se jimi zabývat programátor. Výhodou Vega-Lite je také to, že je ve formátu JSON, se kterým je snadná manipulace v JavaScriptu.

Novinkou, kterou uživatel pocítí ihned je interaktivní manipulace s grafem. Pomocí myši lze s grafem libovolně pohybovat a měnit přiblížení nebo oddálení. Pokud si tedy uživatel zobrazí například nějaký časový průběh signálu, může snadno a rychle zobrazit určitý detail nebo úsek, který ho zajímá.

Pokud je v okně analýzy pouze jeden graf, jeho velikost se sama přizpůsobuje velikosti okna jak v horizontálním, tak vertikálním směru. Dříve to bylo možné pouze ve směru horizontálním.

Novým prvkem je legenda zobrazovaných průběhů, která je na pravé straně mimo oblast grafu (viz obrázky kapitoly 7). Legenda je interaktivní a kliknutím na danou položku lze zobrazit pouze daný průběh, ostatní se zneviditelní. Možný je také výběr více položek pomocí klávesy shift a klikem myši. Toto funkci ocení uživatel v případě, kdy má zobrazeno mnoho průběhů a v daném okamžiku ho bude zajímat jen některý nebo pár některých.

Stejně jako doposud i nadále se při pohybu myši v oblasti grafu zobrazuje popisek dat v grafu (tooltip). Vega-Lite sice umožňuje zobrazení tooltipu, nicméně ten není vyhovující, proto byl použit vlastně vytvořený pomocí HTML elementů, který má úplně stejnou podobu jako tomu bylo doposud.

Implementován byl také export grafů do PNG a SVG. Dále se pracuje na implementaci exportu do  $\text{\LaTeX}$ u a CSV.

Co ještě není vyřešeno a stálo by za to implementovat, jsou markery. V dosavadní verzi markery také nejsou, ale určitě budou implementovány, protože je to velice užitečný nástroj pro odečítání z grafu a zvýrazňování zajímavých bodů.

Jak Vega-Lite, tak i jiné nástroje dokáží zobrazovat nejen spojnicové grafy. Pro zobrazování výsledků analýz byly použity například sloupcové, čárové nebo bodové grafy. Vytvořena byla také struktura pro zobrazování fázorů (napětových i proudových). Opět je vše možné vidět v kapitole 7.

Úplnou novinkou, je možnost kompletního nastavení grafů. Pro tento účel bylo navrženo samostatné okno, které obsahuje čtyři sekce.

První sekcí je *General*, tedy obecné nastavení (obrázek 5.1). Obsahuje nastavení jako barevný režim (tmavý a světlý), nadpis grafu, možnost zobrazení legendy a pravé osy. Pokud se zaškrtně použití pravé osy, zobrazí se dodatková sekce, kde je možné si vybrat, jaké signály budou souviset s jakou osou. Toto má velké opodstatnění v případě, že se do jednoho grafu společně vykreslují napětí i proudy. Ve většině případů je proud o několik řádů menší a nebylo by tedy zobrazení viditelné. Proto lze zobrazit pravou osu, která může mít jiný rozsah.



**Obrázek 5.1:** Nastavení grafů - záložka General.

Druhou sekcí je *Axis*, což je nastavení os (obrázek 5.2). Zde lze nalézt defaultně nastavení levé a spodní osy. V případě použití pravé osy, se zobrazí i třetí blok právě pro pravou osu. Nastavovat lze titulek osy, měřítko (lineární, logaritmické a decibelové), to zda se má použít automatický nebo manuální rozsah os a případně jaký rozsah. Poslední možností je zadání vlastních hodnot na ose.

**Obrázek 5.2:** Nastavení grafů - záložka Axis.

Třetí sekcí je *Grid*, což je nastavení mřížky (obrázek 5.3). Má podobnou strukturu a chování jako sekce *Axis*. Volitelné ze zobrazení vykreslení mřížky, barva, tloušťka čáry a styl. Stylem je myšleno typ čáry. To je prozatím reprezentováno posloupností čísel, které udávají spojitost čáry, tedy zda je to plná nebo přerušovaná čára a s jakými rozestupy.

**Obrázek 5.3:** Nastavení grafů - záložka Grid.

Poslední a docela velmi důležitou sekcí je *Data*, což je nastavení už konkrétních průběhů (obrázek 5.4). V této sekci se vždy objeví tabulka se zobrazovanými průběhy, kterým lze měnit nastavení.

Měnit je možné barvu. Je dáno deset předdefinovaných barev, pokud počet průběhů přesáhne tento počet, tak pro každý další je náhodně vygenerována jiná. Nevyhovují-li uživateli dané barvy, má možnost si je změnit.

Nastavit lze také typ průběhu. Na výběr jsou čtyři možnosti : prostá čára, vyplněná oblast pod průběhem (to může sloužit pro ilustraci výkonu), svislá čára (např Fourierova analýza) a prostý bod. Každá z předchozích možností má pak ještě své vlastní nastavení, což je také vidět na Obrázku 5.4. Například nastavení prosté čáry zahrnuje typ interpolace, šířku, styl a zda mají být zvýrazněny jednotlivé datové body.

Dále bylo umožněno seskupení všech průběhů do jednoho, čímž lze aplikovat stejné nastavení pro všechny průběhy. To tak lze nechat nebo po odškrtnutí možnosti se opět zobrazí všechny položky s již aplikovaným nastavením a lze dále upřesnit nastavení jednotlivých průběhů. Toto nalézá uplatnění například u rozsáhlých parametrických analýzách.

		Name	Color	Mark type	Mark settings			
General	Group all: <input type="checkbox"/>							
Axis	v(8)			Line	Interpolation: Linear	Line width: 2	Stroke-Dash: [1,0]	Points: <input checked="" type="checkbox"/>
	v(9)			Area	Interpolation: Linear			
Grid	v(9,8)			Bar	Width: 2			
	i(V1)			Point	Size: 30			
Data								

**Obrázek 5.4:** Nastavení grafů - záložka Data.





## Kapitola 6

### NGSPICE

Na základě provedení rešerše, konkrétně pak kapitoly 3.2.5 a její tabulky 3.1 bylo rozhodnuto o využití NGSPICE pro numerické analýzy GEEC. Na výpočetní server byl nainstalován NGSPICE verze 32. V PHP bylo vytvořeno rozhraní mezi GEEC a výpočetním serverem. Na rozdíl od předchozího řešení, kde se ještě využíval SPICE OPUS, se toto rozhraní výrazně zjednodušilo. Základní funkcí je již jen přeposílání dat a jednoduché parsování. Dříve v tomto rozhraní probíhalo generování SPICE skriptů a následné parsování přijatých dat. Částečné parsování tam probíhá i nadále, ale velkou úlevou bylo vytvoření vlastního SPICE skriptu pro výpis dat (viz kapitola 6.2). Jak bylo zmíněno, generování skriptu pro spice se z PHP přesunulo do JavaScript. Důvodů bylo několik. Jedním z nich je kompletní předělání celé struktury analýz v JavaScriptu a přidání nových funkcionalit viz kapitola 7.

## 6.1 Nastavení NGSPICE

Ve většině případu není nutné měnit nastavení simulátoru SPICE. V některých případech se však najde obvod, který má problém s konvergencí nebo je nutné použít jinou numerickou metodu (více v kapitole Metodika 2.1). V takovýchto případech se hodí možnost změnit nastavení simulátoru.

### 6.1.1 Obecné nastavení simulátoru

Pro obecné nastavení NGSPICE bylo vytvořeno okno s vybranými možnostmi nastavení. Tyto možnosti jsou dále rozděleny do čtyřech sekcí. Jak celé okno vypadá je zachyceno na obrázku 6.1, kde z důvodu velikosti nebyla třetí a čtvrtá sekce rozbalena.

První sekce je obecné nastavení, které obsahuje například teplotu obvodu nebo počet desetinných míst, na které se mají zobrazit numerické výsledky.

Druhá sekce se týká OP a DC analýzy. Nicméně OP analýza se provádí téměř před každou jinou analýzou, takže je tato sekce asi nejdůležitější. Lze zde nalézt parametry jako *reltol*, *abstol* a *vntol*. Zajímavým parametrem je také *rseries*, který ke každému induktoru přidá sériový rezistor.

Třetí sekcí je nastavení transientní analýzy. Naleznout tam lze nastavení metody numerické integrace (*method*), parametr *interp* pro interpolaci výsledných průběhů nebo například parametr *ramptime*, který určuje na náběh nezávislých zdrojů při spuštění simulace.

Poslední sekcí je nastavení AC analýzy, kde je jediný parametr *noopac*, který určuje zda se má před AC analýzou spouštět OP analýza.

Spice settings

▼ General

Name	Value	Description
numdgt	6	The number of digits to use when printing tables of data. If the number is negative, one fewer digit is printed.
temp	27	Operating temperature of the circuit (in Degrees C)
tnom	27	Nominal temperature for which device models are created (in Degrees C)

▼ OP and DC Options (Transient can be affected too)

Name	Value	Description
reltol	1m	Sets relative error tolerance of the program. The value must be between 0 and 1.
abstol	1p	Sets the absolute current error tolerance (in Amps).
vntol	1u	Sets the absolute voltage tolerance of the program (in Volts)
rshunt	100T	Adds a resistor from each analog node to the ground. The value of the resistor should be high enough to not interfere with circuit operations (in Ohms).
rseries	0	Adds a series resistor to each inductor in the circuit
gmin	1p	Sets min conductance (max resistance) of any device in the circuit (in A/V).
gminsteps	10	Sets the number of steps in the GMIN stepping algorithm. If the value is set to zero, the standard gmin stepping algorithm is skipped. The standard behavior is that gmin stepping is tried before going to the source stepping algorithm.
noopiter	false	Go directly to gmin stepping, skipping the first iteration.
it1	100	Sets Operating Point Analysis iteration limit.
it2	50	Sets DC Analysis iteration limit.
pivrel	1m	Sets the relative ratio between the largest column entry and an acceptable pivot value. The value must be between 0 and 1.
pivot	100f	Sets the absolute minimum value for a matrix entry to be accepted as a pivot.

▶ Transient Analysis Options

▶ AC Analysis Options

Default Save

Obrázek 6.1: Možnosti nastavení NGSPICE.

## 6.1.2 IC a Nodeset

Další možností nastavení SPICE jsou počáteční podmínky IC (initial conditions) a nastavení hodnoty uzlových napětí Nodeset. Jak tyto parametry fungují bylo popsáno dříve, ale v krátkosti. IC udává počáteční hodnotu ze které vychází transientní analýza, kdežto Nodeset ovlivňuje výpočet pracovního bodu, respektive jeho konvergenci.

Pro toto nastavení bylo opět navrženo okno viz obrázek 6.2. V případě, že se toto nastavení vyvolává v transientní analýze, zobrazují se sekce IC i Nodeset. V případě jakékoliv jiné analýzy se zobrazuje pouze sekce Nodeset.

Zobrazují se dostupná uzlová napětí, k nim je přiřazené textové pole pro zadání hodnoty. Také bylo přidáno zaškrtačací políčko, které určuje, zda má být daná hodnota použita. To nachází uplatnění v situaci kdy si uživatel zadá nějaké hodnoty, třeba je použije. Poté si to rozmyslí, z důvodu že mu výsledek nevyhovuje, tak aniž by byl nucen zadané hodnoty mazat, pouze odškrtně použití. Pro ulehčení práce a času byly přidány tlačítka na zaškrtnutí všeho nebo naopak ničeho.

IC and Nodeset			
▼ Nodeset			
Name	Value	use all	use none
v(1)	5	<input type="checkbox"/>	<input checked="" type="checkbox"/>
v(2)		<input type="checkbox"/>	<input type="checkbox"/>
v(3)		<input type="checkbox"/>	<input type="checkbox"/>
▼ IC			
Name	Value	use all	use none
v(1)		<input type="checkbox"/>	<input type="checkbox"/>
v(2)		<input type="checkbox"/>	<input type="checkbox"/>
v(3)		<input type="checkbox"/>	<input type="checkbox"/>
			Save

Obrázek 6.2: Nastavení IC a Nodeset.

## 6.2 Vlastní skripty

NGSPICE nebo obecně SPICE není dokonalý a neumí vše. Od toho je tu možnost si vytvořit vlastní skripty/příkazy, které lze v jiném skriptu spouštět. Tato možnost byla využita i ve vstupních souborech numerických analýz pro GEEC. Zatím byly vytvořeny čtyři skripty, ale určitě se v průběhu času a přidávání nových funkcionalit naskytne příležitost vytvořit další.

Prvním velice důležitým příkazem je *jsonprint*, který umožňuje vypsat data v JSON formátu. Využití tohoto příkazu vedlo k značnému zjednodušení parsování dat přijímaných z NGSPICE. Příkaz má následující syntaxi

```
jsonprint (-c/-r) vector1 [vector2 ...],
```

kde příznak *-c* udává, že se jedná o komplexní data. Naopak příznak *-r* říká, že to jsou reálná data. Dále je požadován minimálně jeden vektor, ale může jich být i více.

Další nadefinovaný skript počítá THD. SPICE sice při Fourierově analýze také zobrazí hodnotu THD, ale v textové formě a nelze s ní jakkoliv manipulovat ani ji vypisovat. Proto byl vytvořen jednoduchý příkaz *thd* se syntaxí

```
thd vector,
```

kde parametr *vector* představuje amplitudové spektrum, získané Fourierovou analýzou.

Třetím skriptem je *geecinit*. Tento skript nic nepočítá, ale definuje užitečné funkce, které se mohou v průběhu analýzy hodit. Obsahuje například funkci pro výpočet hodnoty  $V_{pp}$  (napětí peak-to-peak), dále výpočet proudů rezistorem a kapacitorem, které AC analýza nepočítá.

Zatím posledním, ale docela významným příkazem je *acsweep*. SPICE obecně neumí provádět AC analýzu při jedné frekvenci, při níž by se rozmítal nějaký jiný parametr obvodu. Tato analýza dokáže také nalézt své uplatnění (více kapitola 7.3), proto byl vytvořen příkaz se syntaxí

```
acsweep freq param (lin/dec/oct) points from to tosave [tosave2 ...],
```

kde *freq* je při jaké frekvenci se bude analýza provádět, *param* je jaký parametr se bude rozmítat. Dále je volba rozložení bodů. Následuje počet bodů *points*. Rozsah od (*from*) do (*to*). A na konec veličiny, které se mají uložit *tosave*. Požadována je minimálně jedna.

# Kapitola 7

## Analýzy

Analýzy v GEEC jsou hlavním předmětem této práce. Nejen že přibyly nové typy analýz, došlo také k vylepšení stávajících. Jelikož stará struktura kódu analýz v JavaScriptu byla nevyhovující a již nepřehledná, bylo rozhodnuto o jejím kompletním přepsání. Při té příležitosti se přidali nové analýzy a funkcionality.

V kapitole 7.1 budou popsány některé nové možnosti týkající se obecně analýz. V dalších kapitolách pak budou představeny konkrétní nové nebo vylepšené analýzy.

### 7.1 Nové možnosti v analýzách

V porovnání s ostatními online simulátory je tento již na poměrně dobré úrovni, nicméně je tu stále ještě co vylepšovat a přidávat. Některé z nových vylepšení budou zde dále představeny.

Ne úplně znatelnou změnou je, že nabízené výsledky nejsou dány na základě výpisu ze SPICE, ale jsou generovány na základě nakresleného obvodu a jeho součástek. Tím se předešlo chaosu způsobeného pojmenováváním uzlových napětí pomocí textového řetězce. SPICE totiž standardně vypisuje uzlové napětí ve tvaru „v(číslo uzlu)“, pokud je místo čísla zadán text, což je možné. Nevypisuje se „v(text)“, ale pouze „text“, a to může být velmi matoucí.

#### 7.1.1 Menu

Ve všech okenních analýzách přibylo horizontální menu. Důvod je prostý. Přidáním nových funkcionalit a možností by bylo nepřehledné a nevhledné mít v okně analýzy mnoho tlačítek. Jednotlivé možnosti lze seskupit do sekcí menu, čímž se celá struktura stává přehlednější a elegantnější. Jak takové menu vypadá je možné vidět na obrázcích z následujících kapitol, kde jsou ukázána celá okna analýz.

Obecně nelze popsat obsah menu, protože se může lišit podle dané analýzy. Nicméně pro představu, takové menu grafické analýzy obsahuje čtyři sekce *File*, *Graph*, *Analysis* a *Help*.

Sekce *File* většinou v sobě zahrnuje možnost uložení do Fast analysis, export grafů ve formě PNG, SVG,  $\text{\LaTeX}$  nebo nastavení jména dané analýzy.

Sekce *Graph* zahrnuje nastavení grafů, přidání subgrafu (kapitola 7.1.2) nebo možnost překreslení všech grafů v daném okně.

V sekci *Analysis* jsou možnosti otevření Design variables (kapitola 7.1.3), nastavení SPICE nebo IC a Nodeset (jedná-li se o SPICE analýzu). Dále například odkaz na FFT a Fourierovu analýzu, jde-li o transientní analýzu.

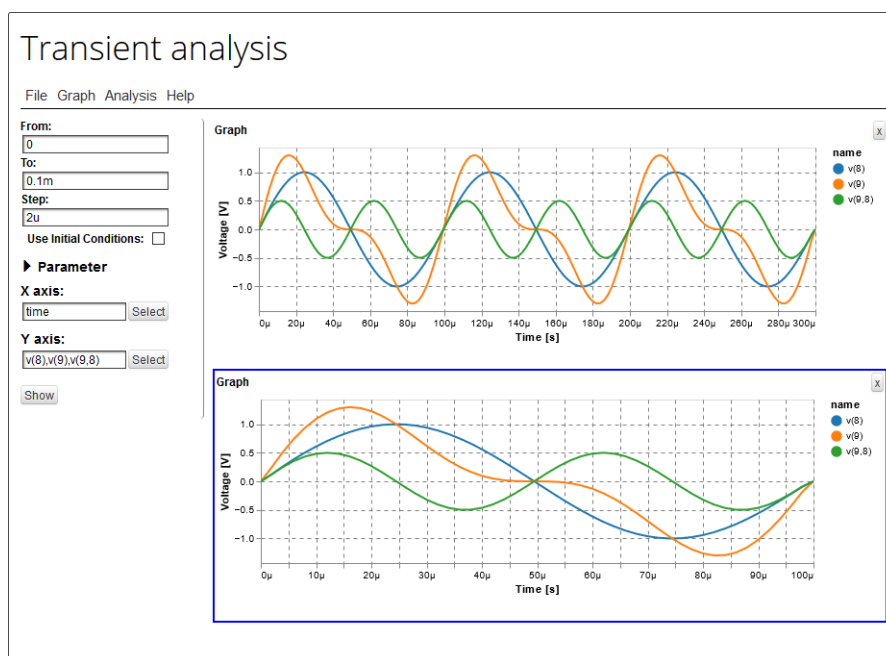
Nesmí chybět sekce *Help*, kde jsou nějaké informace o dané analýze, posílaném skriptu do SPICE i Maple nebo chybový log.

## 7.1.2 Více grafů v jednom okně

Požadavkem také bylo umožnit přidání více grafů do jednoho okna analýzy. Je sice možné si otevřít více oken daného typu analýzy a zobrazit si různé výsledky, nicméně někdy je vhodné, až potřebné, si zobrazit více grafů v jednom okně pod sebou. Proto tato možnost byla implementována a výsledek je vidět na obrázku 7.1.

Jde vlastně o více nezávislých analýz stejného typu v jednom okně. Analýzy respektive grafy lze jednoduše přidávat pomocí menu *Graph/Add sub graph*. Odebírání je ještě jednodušší, pomocí křížku v pravém horním rohu grafu.

Modrým orámováním je zvýrazněno, jaký graf je právě aktivní. Pro aktivní graf jsou vždy platné hodnoty v konfiguračním panelu analýzy i položky v menu (export obrázků, nastavení grafu, ...). Výjimkou je položka *Graph/Refresh all graphs*, díky níž lze přepočítat všechny analýzy v daném okně (například při změně v obvodu).



Obrázek 7.1: Více grafů v jednom okně.

### 7.1.3 Design variables

Po vzoru jiných simulátorů a to nejen online bylo umožněno zadání symbolické hodnoty parametru součástky. Respektive se jedná o parametrické zadání hodnoty. Toho lze využít například zadáním stejného parametru více součástkám. Následně změnou tohoto parametru se mění několik součástí najednou. Velice užitečná novinka.

Symbolický simulátor PraCAN je na tuto funkci přímo stavěn, takže tam je to bez problému. Obecně numerické simulátory typu SPICE už na tom jsou hůře, ale není to neřešitelný problém. Parametrické zadávání je umožněno příkazem `.param`, ale vždy se musí jednat o numerickou hodnotu. Proto bylo vytvořeno okno se seznamem takzvaných Design variables, které jsou automaticky aktualizovány podle zadaných symbolických hodnot součástek. Pro SPICE se musí vždy symbolické hodnotě přiřadit numerická. Pro PraCAN tomu tak není, tam je možné mít jak symbolickou, tak numerickou hodnotu. Z toho důvodu bylo ke každé *Design variable* přidáno zatrhávací políčko, které určuje zda má být daná numerická hodnota přiřazena (pouze pro PraCAN).

Name	Value	assign value
R	2k	<input checked="" type="checkbox"/>
c	1u	<input type="checkbox"/>

Save

Obrázek 7.2: Nastavení hodnot Design variables.

### 7.1.4 Substitute a přiřazení

Po vzoru symbolických analýz, kde se umožnilo zadávání substitucí a přiřazení již před delší dobou. Byly tyto možnosti vylepšeny a rozšířeny i do numerické analýzy. Zadávání substitucí i přiřazení bylo vyřešeno pomocí kalkulatoru viz kapitola 8.1.

Substituce jsou vlastně takovým ulehčením práce. Při analýze, pokud uživatel potřeboval změnit hodnotu parametru nějaké součástky, se musel překlíknout do schématu a změnit ji. Ještě horším případem byla změna stejné hodnoty u více součástek. To se substitucemi odpadá a lze měnit hodnoty přímo v analýze. Co se týče příkladu s více součástkami je řešením použití *Design variable* (viz předchozí kapitola) a její následné substituce.

Co se přiřazení týče, jde vlastně o definování vlastní proměnné, se kterou lze dále počítat nebo jí pouze vypsát/vykreslit. Přiřazení je bráno jako samostatný výsledek, který může mít své vlastní substituce. Proto se při zadávání nového nebo editaci stávajícího přiřazení otevírá zcela nový kalkulator.

Takovým příkladem použití substitucí i přiřazení může být výpočet výstupního odporu zesilovače. Prvním přiřazením bude výstupní napětí naprázdno  $U_o$ , které bude mít substituci zatěžovacího odporu ideálně nekonečno, reálně v rádech mega/giga Ohmů. Druhým přiřazením bude výstupní proud

nakrátko  $I_k$ , který bude mít substituci zatěžovacího odporu rovnu nule nebo v řádech mili Ohmů. Výsledný výstupní odpor pak bude  $U_o/I_k$ , což může být další přiřazení nebo už rovnou výsledný výraz.

## 7.2 AC analýza jedné frekvence

Jak bylo v kapitole 7.1 zmíněno, výsledky se nenabízejí podle toho, co SPICE vrátí, ale jsou generovány v editoru na základě vytvořeného obvodu. To přináší možnost vytvořit AC analýzu jedné frekvence, po vzoru již existující symbolické a semisymbolické v simulátoru PraCAN. Jak okno takové analýzy vypadá je ukázáno na obrázku 7.3. Jelikož se jedná o numerickou analýzu, je potřeba vždy zadat frekvenci. To je řešeno textovým polem nad výsledky. Pro přepočítání výsledků je třeba kliknout v menu na *Analysis/Refresh results*.

Výsledky se defaultně zobrazují ve formátu Magnitude-Phase. Je-li třeba zobrazit výsledky v jiné podobě. V menu *Results type* je další možnost Real-Imaginary.

Dalším problémem bylo to, že SPICE nepočítá proudy součástkami v AC analýze, tedy mimo zdroj napětí a induktor. Proto byly nadefinovány funkce pro výpočet proudů rezistorem a kapacitorem

$$\text{define ir}(v,r) \ v/r \ ,$$

$$\text{define ic}(v,\text{cap},f) \ v*2*\text{const.pi}*f*\text{cap}*const.i \ .$$

Spice AC one frequency analysis

File Analysis Results type Help

Frequency: 1k

▼ Nodes

Name	Magnitude	Fast analysis	Phase	Fast analysis
v(1)	1	Add	0	Add
v(3)	157.18m	Add	-80.96	Add

▼ C1

Name	Magnitude	Fast analysis	Phase	Fast analysis
i(C1)	987.57u	Add	9.04	Add

▼ R1

Name	Magnitude	Fast analysis	Phase	Fast analysis
i(R1)	987.57u	Add	9.04	Add

▼ V1

Name	Magnitude	Fast analysis	Phase	Fast analysis
i(V1)	987.57u	Add	-170.96	Add

Obrázek 7.3: Okno AC analýzy jedné frekvence.

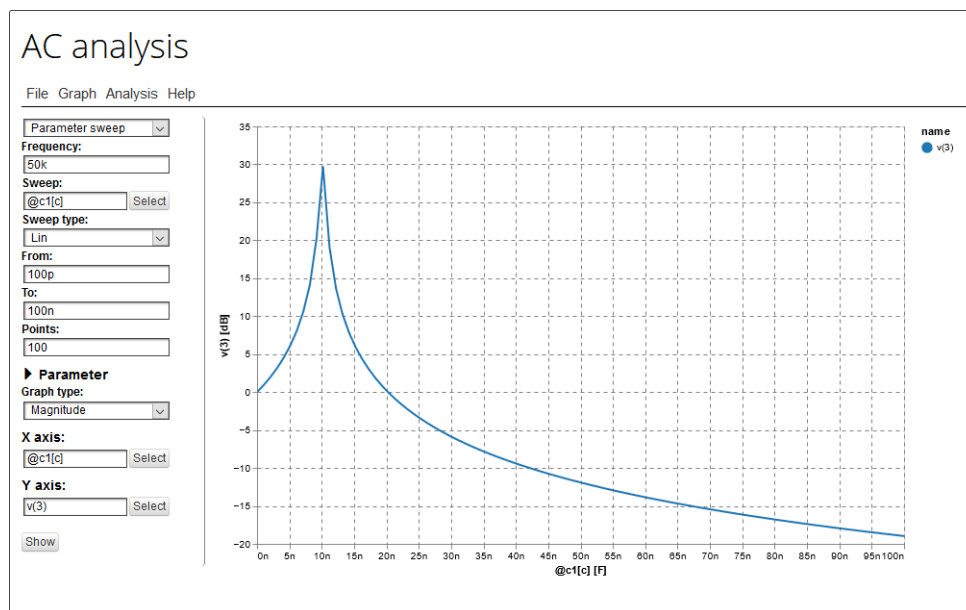


## 7.3 AC analýza - rozmítání parametru

Jednou z nových možností v AC analýze je analýza jedné frekvence při rozmítání jiného parametru než frekvence. Tato možnost není ve SPICE přímo možná. Byl proto vytvořen speciální příkaz *acsweep*, zmíněný v kapitole 6.2, který provedení takovéto analýzy umožní.

V konfiguračním panelu AC analýzy na obrázku 7.4, je třeba vybrat *Parameter sweep* (první položka v nastavení od shora). Po vybrání této možnosti se přizpůsobí zbytek konfiguračního panelu. Zobrazí se položka *Frequency* udávající při jaké frekvenci se bude analýza provádět. Dále výběr parametru (*Sweep*), jeho rozsah a počet bodů. Samozřejmostí je možnost parametrické analýzy. Následuje typ grafu (amplituda, fáze, ...) a specifikování x-ové i y-ové osy. Defaultně se x-ová osa nastaví na tu samou hodnotu, jako je *Sweep*.

Typickým uplatněním této analýzy může být například zjištění, pro jakou hodnotu parametru při dané frekvenci nastává rezonance. To je též ukázáno na obrázku 7.4, kde je dána frekvence 50 kHz. Přibližně pro hodnotu 10 nF kapacitoru C1 nastává při 50 kHz rezonance.



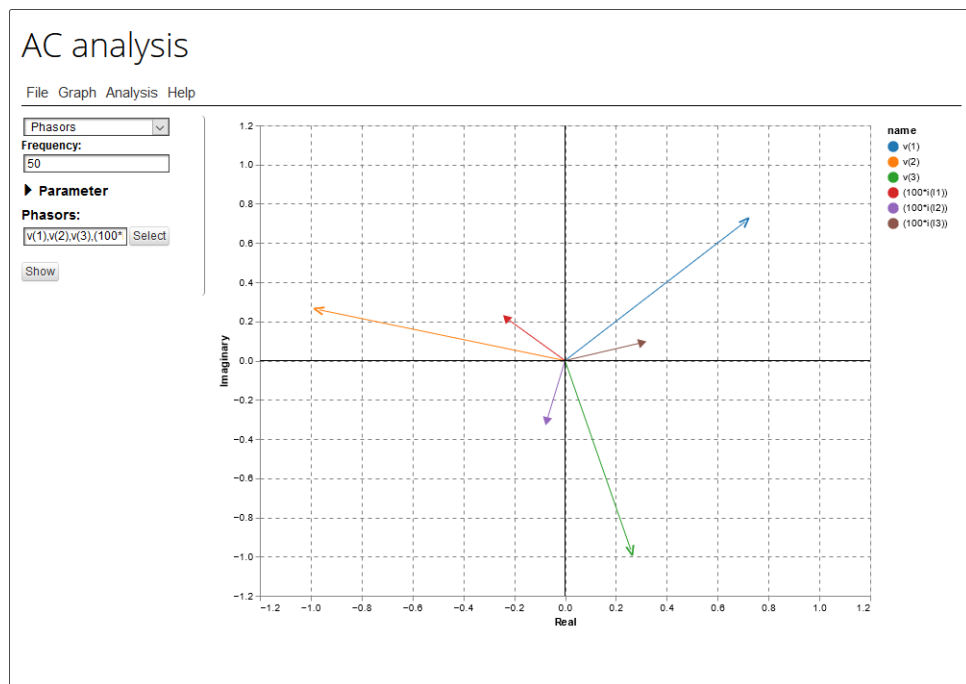
Obrázek 7.4: Okno AC analýzy při rozmítání parametru.

## 7.4 AC analýza - fázory

Další novinkou je opět grafická AC analýza jedné frekvence, kdy se výsledná data zobrazují ve formě fázorů v komplexní rovině. Fázory jsou zobrazeny ve formě šipek, které není možné standardně vykreslit v knihovně Vega. Byly proto dodatečně dodefinovány a při té příležitosti se vytvořili jak napěťové, tak proudové šipky.

V konfiguračním panelu je třeba vybrat položku *Phasors*, poté se zobrazí možnost zadání frekvence a konkrétních fázorů, které chce uživatel zobrazit. Vše je opět ukázáno na obrázku 7.5 včetně fázoru napětí a proudů.

Vzhledem k tomu, že se do jedné souřadnicové soustavy dají zobrazit jak napětí, tak proudy. A ve většině situací jsou proudy o několik řádů menší. Dají se vynásobit konstantou, aby byly lépe viditelné viz legenda grafu na obrázku 7.5.



Obrázek 7.5: Okno AC analýzy se zobrazenými fázory.

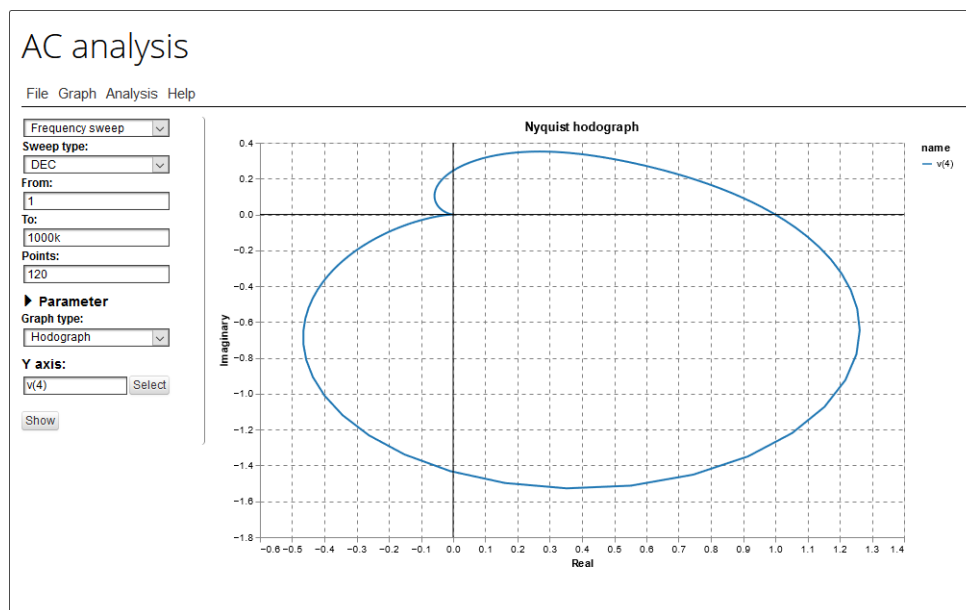
## 7.5 AC analýza - hodograf

Hodograf je alternativním zobrazením frekvenční charakteristiky obvodu v komplexní rovině. Jedná se o křivku, jejíž každý bod odpovídá konkrétní hodnotě kmitočtu. Vzdálenost určitého bodu od počátku souřadnicového systému určuje modul přenosu. Úhel, který svírá reálná osa se spojnicí konkrétního bodu a počátku, udává fázi přenosu.

Je-li obvod pasivní, pak zobrazená charakteristika leží uvnitř jednotkové kružnice. V případě aktivních a rezonančních obvodů tomu tak být nemusí.

Hodograf lze využít například k posouzení stability obvodů se zpětnou vazbou (Nyquistovo kritérium stability).

Konfigurační panel analýzy je de facto totožný s klasickou frekvenční charakteristikou implementovanou již dříve. Pouze se musí v položce *Graph type* zvolit možnost *Hodograph*. Konfigurační panel i konkrétní příklad vykresleného hodografu je na obrázku 7.6.



Obrázek 7.6: Okno AC analýzy se zobrazeným hodografem.

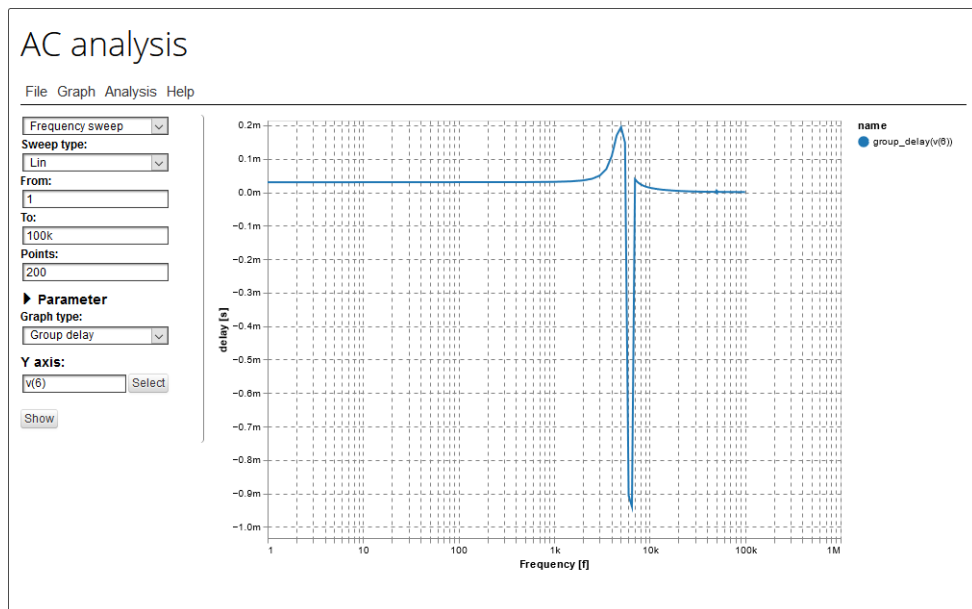
## 7.6 AC analýza - skupinové zpoždění

Stejně jako hodograf i zobrazení skupinového zpoždění je pouze nastavbou klasické AC analýzy. Zobrazí se zvolením možnosti *Group delay* v položce *Graph type*.

Nejedná se o nic jiného, než o záporně vzatou derivaci fázové charakteristiky podle úhlové frekvence. Nicméně v NGSPICE je k tomu určena speciální funkce *group\_delay*, takže není potřeba nic počítat.

Skupinové zpoždění je časové zpoždění signálu procházející daným obvodem v závislosti na frekvenci. Jinými slovy, jde o zpoždění jednotlivých frekvenčních složek signálu. Díky zobrazení skupinového zpoždění lze předem odhadnout, zda bude signál zkreslen, bez použití Fourierovy analýzy.

Ukázka zobrazení skupinového zpoždění viz následující obrázek 7.7.



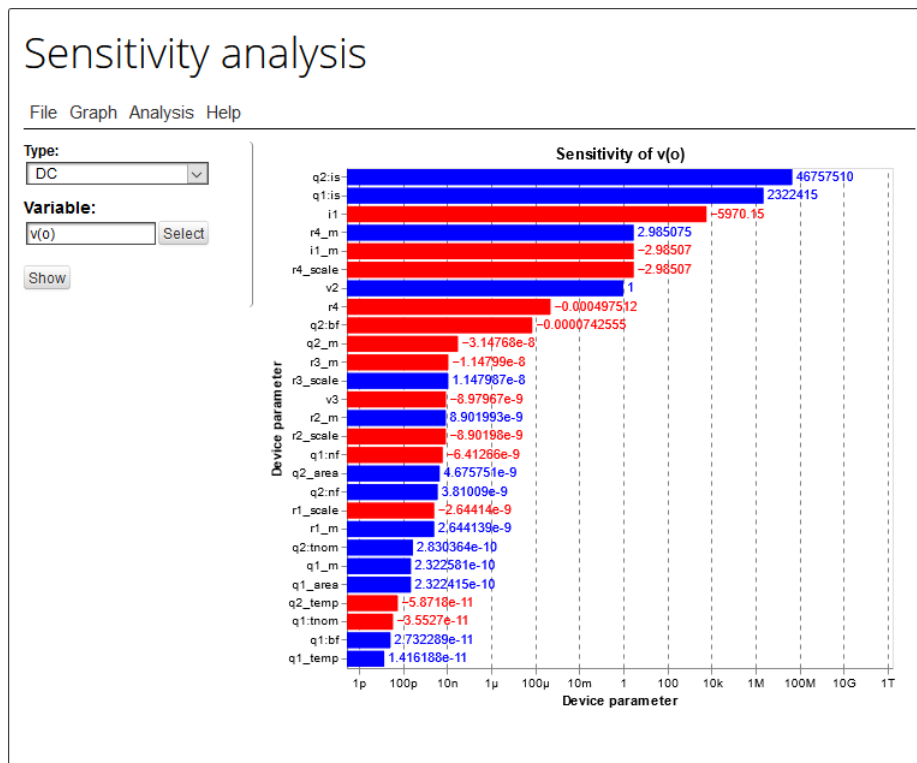
Obrázek 7.7: Okno AC analýzy s výsledkem skupinového zpoždění.

## 7.7 Citlivostní analýza

Úplně nově přidanou analýzou je analýza citlivosti. Citlivostní analýza ve SPICE má dvě možnosti. V doméně AC a DC, což lze vybrat pomocí položky *Type* v konfiguračním okně viz obrázek 7.8.

Pro možnost DC se vypočítají citlivosti zadané proměnné (*Variable*) na všechny nenulové parametry obvodu. Výsledkem jsou numerické hodnoty reprezentující absolutní změnu proměnné na jednotkovou změnu daného parametru. Pro lepší a elegantnější reprezentaci výsledků byl použit horizontální sloupcový graf, nikoliv prostý výpis numerických hodnot. Tím, že hodnoty mohou být i záporné, zobrazuje se absolutní hodnota a znaménko je rozlišeno barvou. Defaultně jsou kladné hodnoty modré a záporné červené viz obrázek 7.8. Samozřejmě uživatel si může barvy změnit v nastavení grafu.

Pro možnost AC se vypočítají citlivosti v zavitosti na frekvenci. Při zvolení *AC* v položce *Type*, se v konfiguračním panelu přidávají položky pro zadání rozsahu frekvence. Výsledkem je pak nikoliv sloupcový, ale klasický spojnicový graf. Kde na x-ové ose je vynesena frekvence a na y-nové ose jsou citlivosti parametrů.



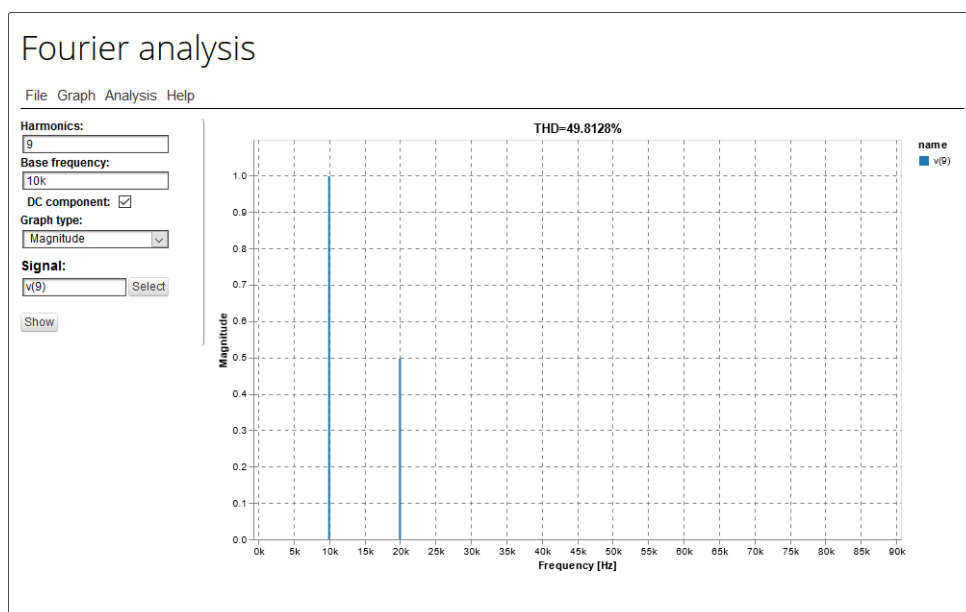
Obrázek 7.8: Okno citlivostní analýzy.

## 7.8 Fourierova analýza

Fourierova analýza vychází z transientní analýzy, kde je na ní odkaz v menu *Analysis/Fourier*. Zobrazí se nové okno Fourierovy analýzy, kde je možné dále zkoumat časové průběhy ve frekvenční oblasti.

Konfigurační okno obsahuje volbu počtu harmonických složek (*Harmonics*), základní harmonickou (*Base frequency*), možnost zahrnutí nebo vynechání DC složky, což má uplatnění při volbě logaritmické frekvenční osy. Dále tam lze nalézt typ grafu (*Graph type*), což ze opět výběr, zda má být zobrazené amplitudové nebo fázové spektrum, případně obojí. Nakonec se volí signál (*Signal*), který byl vypočítán v transientní analýze a bude zde dále analyzován.

S Fourierovou analýzou úzce souvisí parametr THD, jehož hodnota je počítána zvlášť příkazem *thd* viz kapitola 6.2. Tato hodnota je následně zobrazena v titulku grafu, jak je možné vidět na obrázku 7.9. V grafu jsou zobrazeny diskretní spektrální složky ve formě svislých čar s odpovídající velikostí.

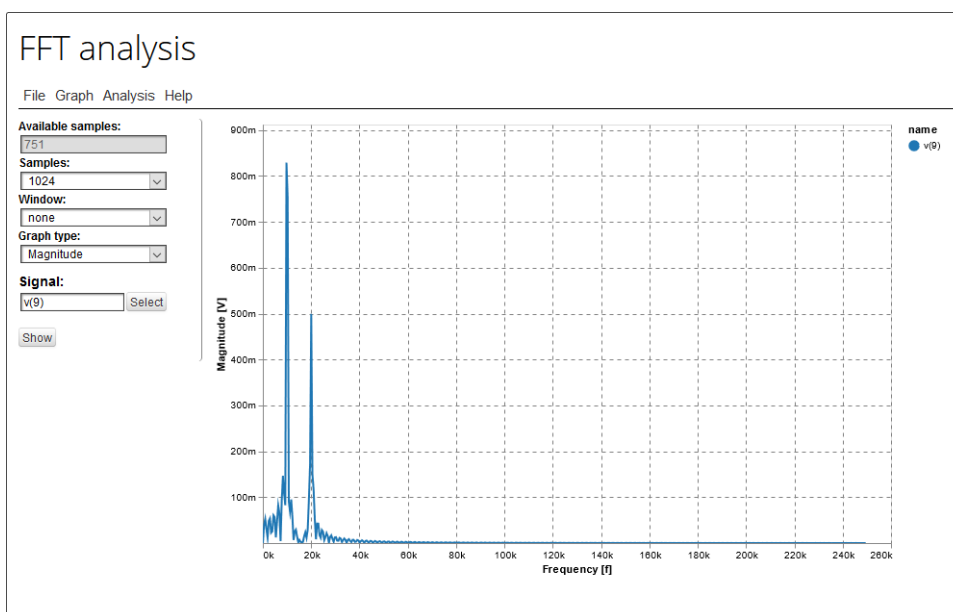


Obrázek 7.9: Okno Fourierovy analýzy.

## 7.9 FFT analýza

Stejně jako Fourierova, tak i FFT analýza vychází z transientní. Přístupná je v menu *Analysis/FFT*. Tato analýza se provádí přímo v JavaScriptu, což přináší mnohé výhody. Vše je ještě ve fázi testování, takže se tento přístup může ukázat nevhodným. Hlavní výhodou je, že se do SPICE nebo Maple nemusí posílat zpátky vypočtené průběhy, aby se nad nimi dále spustila FFT analýza. Jiným řešením by bylo opětovné spuštění transientní analýzy a následná FFT, což také není časově optimální řešení, zvláště pro delší simulace. Proto byla zvolena možnost počítat FFT přímo v JavaScriptu, kde se vezmou data z transientní analýzy a provede se nad nimi výpočet spektra.

Jak okno FFT analýzy vypadá je ukázáno na obrázku 7.10. Konfigurační panel obsahuje informaci o počtu bodů časové analýzy (*Available samples*). Toho je využíváno při výběru počtu vzorku pro výpočet FFT, který musí být mocninou čísla dvě. Je-li vybraný počet menší, signál je oříznut danou na délku. Je-li však vybraný počet větší, než dostupný počet vzorků, je na konec signálu dodáno potřebné množství nulových hodnot. Dále je možné použití váhovacího okna (*Window*), typ zobrazení (stejně jako u Fourierovy analýzy) a signál pro výpočet spektra.



Obrázek 7.10: Okno FFT analýzy.

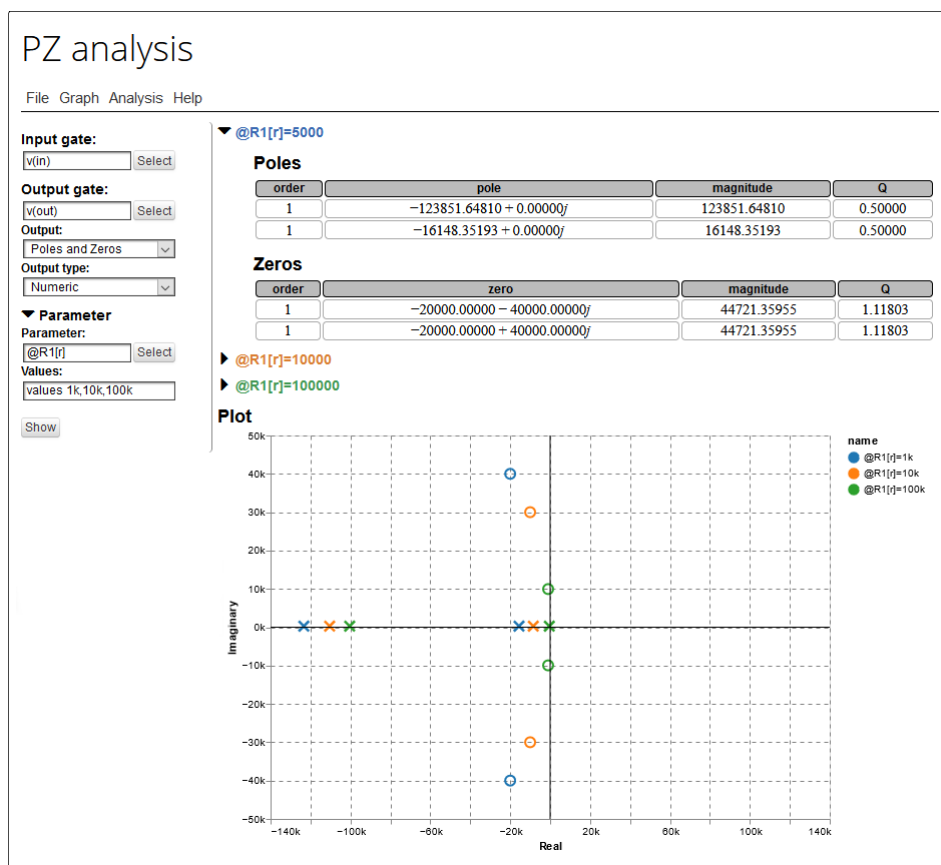
## 7.10 PZ analýza

PZ analýza, neboli analýza pólů a nul, je implementována jak pro SPICE, tak i pro PraCAN. Výhodou analýzy v PraCAN je výpis pólů a nul v symbolickém tvaru.

Výsledkem této analýzy je jednak výpis pólů a nul, buď v numerické nebo symbolické podobě. V případě numerických výsledků je zobrazena i grafická podoba pólů a nul v komplexní rovině. Dále pro numerické hodnoty je vypočítána velikost (*magnitude*) a činitel jakosti ( $Q$ ) a nechybí ani násobnost (*order*).

V konfiguračním panelu je nutné zadat vstupní a výstupní bránu daného obvodu (*Input a Output gate*). Vybírá se možnost výpočtu pouze pólů, nul nebo obojího (*Output*). Pro PraCAN je možný výběr symbolických nebo numerických hodnot (*Output type*). K dispozici je i parametrická analýza.

Ukázka okna numerické PZ analýzy je na obrázku 7.11, kde je zároveň prezentována parametrická analýza. Jednotlivé výsledky pro daný parametr jsou jak v grafu, tak v ve výpisu barevně odlišeny.



Obrázek 7.11: Numerické výsledky parametrické PZ analýzy.



Jak bylo zmíněno, PraCAN umožňuje vrátit i symbolicky vypočtené póly a nuly. To je ukázáno na obrázku 7.12. Pro symbolické hodnoty už nedává smysl počítat velikost a činitel jakosti, proto také tyto položky nejsou zahrnuty.

V PraCAN existuje PZ analýza jako taková, nicméně se ukázalo výhodnějším provést TF analýzu v s-doméně. Sestavit přenosovou funkci a vyjádřit kořeny čitatele a jmenovatele, tedy nuly a póly.

PZ analysis

File Graph Analysis Help

**Input gate:**  
v(in) Select

**Output gate:**  
v(out) Select

**Output:**  
Poles and Zeros

**Output type:**  
Symbolic

**Parameter**  
Show

**Poles**

order	pole
1	$-1/2 \frac{R_2 C_1 - C_2 R_1 - C_2 R_2 - \sqrt{C_1^2 R_2^2 - 2 C_1 C_2 R_1 R_2 + 2 C_1 C_2 R_1 R_2^2 - C_1^2 R_1^2 - 2 C_1^2 R_1 R_2 - C_2^2 R_1^2 - 2 C_1^2 R_1 R_2 - C_2^2 R_2^2}}{C_1 C_2 R_1 R_2}$
1	$-1/2 \frac{R_2 C_1 - C_2 R_1 - C_2 R_2 + \sqrt{C_1^2 R_2^2 - 2 C_1 C_2 R_1 R_2 + 2 C_1 C_2 R_1 R_2^2 - C_1^2 R_1^2 - 2 C_1^2 R_1 R_2 - C_2^2 R_1^2 - 2 C_1^2 R_1 R_2 - C_2^2 R_2^2}}{C_1 C_2 R_1 R_2}$

**Zeros**

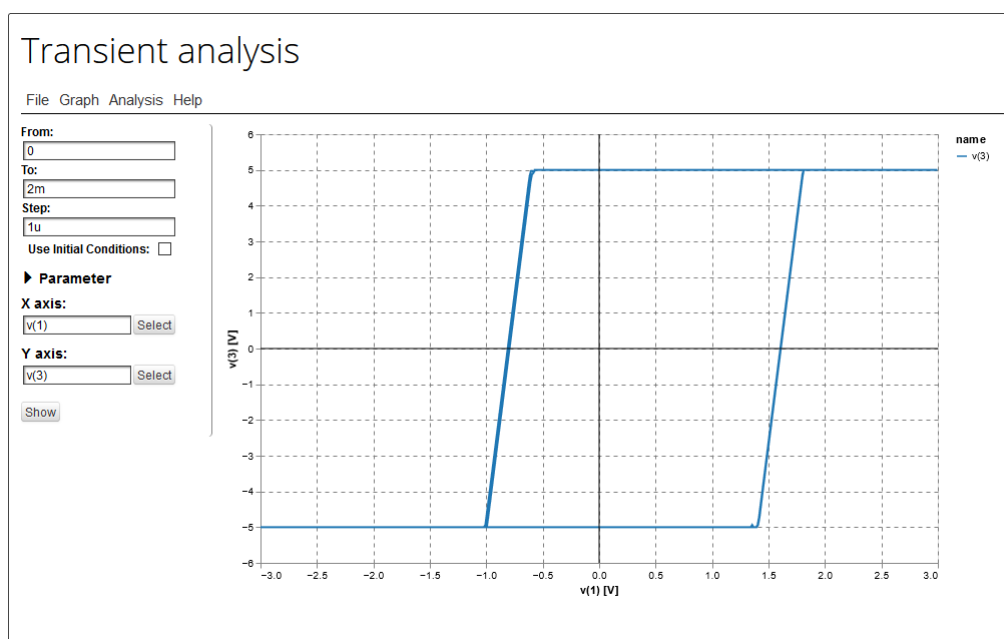
order	zero
1	$-1/2 \frac{R_2 C_1 - C_2 R_2 - \sqrt{C_1^2 R_2^2 - 4 C_1 C_2 R_1 R_2 + 2 C_1 C_2 R_2^2 - C_2^2 R_2^2}}{C_1 C_2 R_1 R_2}$
1	$-1/2 \frac{R_2 C_1 - C_2 R_2 + \sqrt{C_1^2 R_2^2 - 4 C_1 C_2 R_1 R_2 + 2 C_1 C_2 R_2^2 - C_2^2 R_2^2}}{C_1 C_2 R_1 R_2}$

Obrázek 7.12: Symbolické výsledky PZ analýzy.

## 7.11 Transientní analýza - XY mód

Vylepšení se dočkala také transientní analýza, kde přibyla možnost výběru x-ové osy (*X axis*) v konfiguračním panelu analýzy. V principu transientní analýza počítá časové průběhy napětí a proudů, je tedy logické mít na x-ové ose čas. V úvahu také přichází vynést závislost jednoho časového průběhu na jiném, což je de facto obdobou XY módu osciloskopu.

Díky této možnosti lze nyní vykreslovat zajímavé věci jako Lissajouseovy obrazce. V obvodech generující chaotické signály (generátory chaosu), lze vykreslit zajímavé „šnekovitě“ útvary. Takovým praktičtějším použitím může být zobrazení hysterézní křivky, jak je to možné vidět na obrázku 7.13.



Obrázek 7.13: Hysterézní křivka zobrazená pomocí transientní analýzy.

## Kapitola 8

### Výběr výsledků

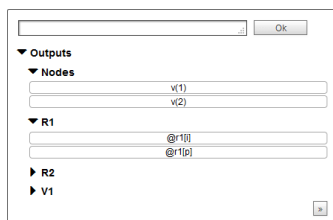
Ne vždy uživateli stačí vypsát nebo vykreslit pouze nabízený výsledek simulace. Často je potřeba s výsledky dále pracovat a zobrazovat celé výrazy.

V práci [1] bylo představeno základní okno s možností výběru výsledků. Pokud uživatel znal syntaxi a příkazy SPICE i Maple, mohl si sám zadat nějaký výraz. V této kapitole bude představen kalkulátor právě pro zadávání i složitějších výrazů. Dále pak zde bude představeno nové zadávání parametrických analýz, které vychází z podobné struktury.

#### 8.1 Kalkulátor

Jak bylo zmíněno, kalkulátor vznikl za účelem uživateli ulehčit práci při sestavování složitějších výrazů, které mají být vypočítány nebo vykresleny. Tím, že se jedná o zadávání výsledků, bylo do kalkulátoru zahrnuto zadávání substitucí i nových přiřazení.

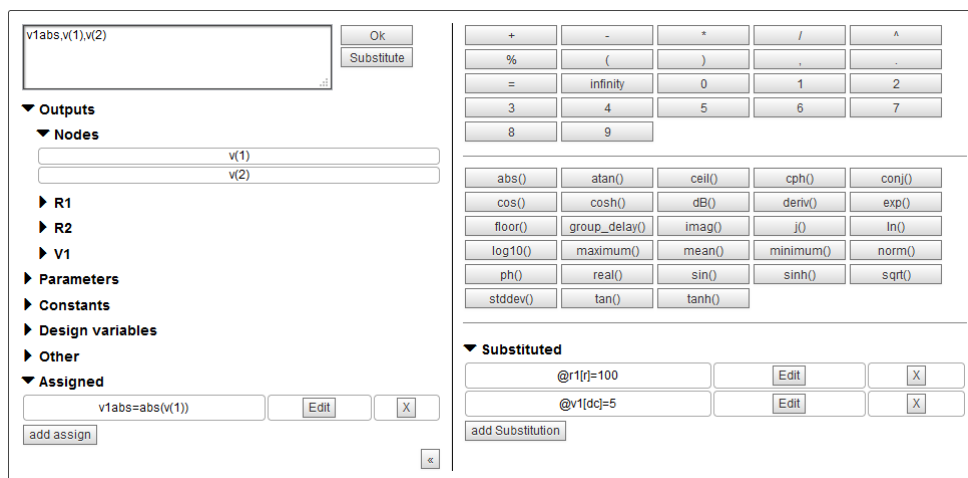
V konfiguračním panelu dané analýzy většinou bývá možnost určit, co se má vypočítat a vykreslit. Pro výběr daných položek slouží textové pole, do kterého lze vypsát chtěné výrazy. Častější postup je, že vedle textového pole je tlačítko, které otevře kalkulátor v minimalizované verzi, pro výběr prostých výsledků viz obrázek 8.1. V pravém dolním rohu je tlačítko, které rozbalí kalkulátor a všechny jeho možnosti. Rozbalený kalkulátor je na obrázku 8.2. Běžný nenáročný uživatel většinou chce zobrazit jen uzlová napětí nebo proudy součástkami. Kdyby se pro takového uživatele zobrazil celý kalkulátor, mohl by být zmatený nebo by to na něj bylo složité. Z toho důvodu se defaultně zobrazuje minimalizovaný kalkulátor a pokud chce více, tak si ho rozbalí.



Obrázek 8.1: Minimalizovaný kalkulátor.

Kalkulátor se skládá ze dvou částí. Na levé straně jsou záložky s možnostmi zobrazení a výpočtu. Záložka *Outputs* obsahuje možné zobrazitelné výsledky analýz (napětí a proudy). V záložce *Parameters* jsou parametry součástek. *Constants* obsahují definované konstanty (Pí, e, elementární náboj,...). *Design variables* nabízí pro obvod definované Design variables viz kapitola 7.1.3. V *Others* jsou nijak nezařazené položky jako teplota obvodu, čas, frekvence a jiné. Poslední záložkou je *Assigned*, kde jsou k dispozici definovaná přiřazení (o přiřazení kapitola 7.1.4). Ty lze v kalkulátoru přidávat, odebírat i upravovat. Kliknutím na jakoukoliv položku z jakékoliv sekce se do zadávacího textového pole v levé horní části kalkulátoru přidá daná položka.

Pravá část kalkulátoru obsahuje symboly a čísla, které lze k výpočtu použít. Dále funkce, ať už matematické (sinus, cosinus,...) nebo jiné (group\_delay, maximum,...). Je zde také záložka *Substituted*, ve které se nacházejí substituce, jež mají být použity pro daný výraz. Substituce je možné taktéž přidávat, odebírat i upravovat. Přidávat je lze dvěma způsoby. První přichází v úvahu, když v textovém poli není ještě zadán žádný výraz. Zapiše se tam daná substituce ve tvaru: „název parametru=hodnota“, a stiskne tlačítko *Substitute*. V opačném případě, když už textové pole obsahuje nějaký výraz, je třeba kliknout na tlačítko *add substitution*. Tím se zapamatuje již zadaný výraz a textové pole se vyprázdní. Další postup je už obdobný s tím rozdílem, že po stisku tlačítka *Substitute* se do textového pole vrátí předtím uložený výraz.



Obrázek 8.2: Kalkulátor.

## 8.2 Nastavení parametrické analýzy

Parametrické analýzy už byly představeny v předchozí práci [1]. V této práci přichází další vylepšení, týkající se zadávání parametrické analýzy.

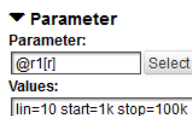
Zadávací okno vychází ze struktury kalkulátoru (obrázek 8.3). V levé části jsou opět záložky položek, které lze parametricky měnit. V pravé části je pak zadávání, jakých hodnot bude vybraný parametr nabývat. Pomocí comboboxu (rozbalovací seznam) lze vybrat typ rozmítání daného parametru. Na výběr je možnost *Values*, kde si uživatel sám zadá konkrétní hodnoty. Dalšími možnostmi jsou automatické metody generování hodnot: lineární, logaritmické, oktákové, dekadické a pomocí Gaussova rozložení.



Obrázek 8.3: Okno pro nastavení parametrické analýzy.

Ve spodní části je vždy pole *Output*, ve kterém se zformuluje výraz obsahující informace z předchozích polí. Tento výraz je shodný se syntaxí NGSPICE příkazu *compose*, který vygeneruje vektor hodnot s daným rozdělením bodů.

Po potvrzení výběru pomocí tlačítka *Ok*, se zvolené hodnoty přesunou do konfiguračního panelu analýz, konkrétně do sekce *Parameter* viz obrázek 8.4.



Obrázek 8.4: Předání nastavení do konfiguračního panelu analýzy.



## Kapitola 9

### Interaktivní analýza

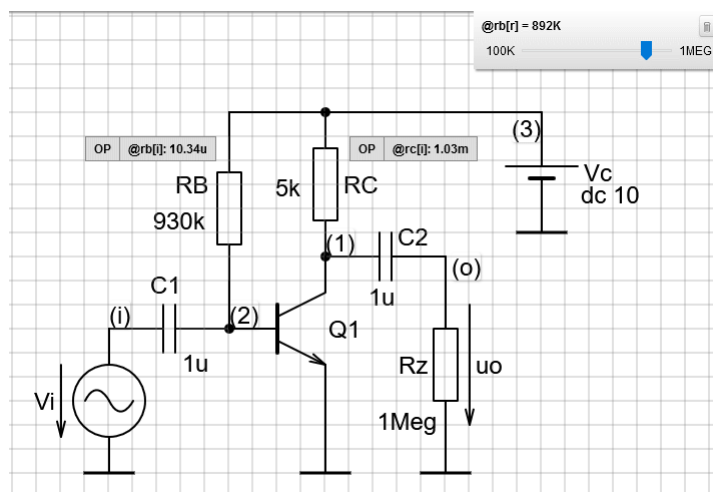
Další novinkou, která je stále ve vývoji, je interaktivní analýza. Mou hlavní náplní bylo zprovoznění funkčnosti z hlediska volání a vykonávání analýz.

Jedná se o rámečky s výsledky dané analýzy (zatím jen OP), které jsou uloženy ve Fast analysis. Z Fast analysis lze pomocí tlačítka přesunout výsledek do schématu a dále přesunout například k součástce, se kterou souvisí.

Je to prvotní verze rámečků. Dalším krokem bude pro zlepšení interaktivity bude možnost zadání podmínky, při jejíž vyhodnocení se bude měnit barva rámečku. Toto nalezne využití například pro kontrolu režimu tranzistoru, výkonového zatížení dané součástky a mnohé další aplikace.

Dalším interaktivním prvkem je změna hodnoty parametru součástky nebo obvodu pomocí posuvníku. Pro přidání posuvníku do obvodu je nutné zadat parametr, který se bude měnit. Dále rozsah hodnot a počet kroků, mezi kterými se bude hodnota pohybovat. Při detekci změny se přepočítají výsledky nejen v obvodu, ale také ve Fast analysis.

Jak vypadají rámečky s výsledky a posuvník je ukázáno na Obrázku 9.1.



**Obrázek 9.1:** Výsledky analýzy ve schématu a interaktivní změna parametru.





# Kapitola 10

## Závěr

Cílem této práce bylo navázat na práce [11] a [1], a dále pokračovat ve zdokonalování Grafického editoru elektrotechnických obvodů (GEEC).

Práce se zabývala především analýzami elektrických obvodů. Jednak numerickými v programu NGSPICE, a jednak symbolickými v programu Maple, využívající knihovnou PraCAN. Byly vylepšeny stávající analýzy a přidány nové. Novými analýzami jsou citlivostní, Fourierova, FFT a PZ. Velkých změn se dočkala AC analýza, kde přibyly možnosti zobrazení fázorů, hodografu, skupinového zpoždění a rozmítání jiného parametru, než je frekvence.

Předělána byla celá struktura zobrazování grafů včetně umožnění kompletního nastavení. Rozsáhlé nastavení bylo vytvořeno pro simulátor NGSPICE. Implementovalo se mnoho dalších věcí týkající se zadávání i obecně analýz.

Během této práce se napsalo tisíce řádků kódu v desítkách souborů, a to nejen v JavaScriptu, ale i v PHP, CSS, HTML, SPICE a Maple. Přepsána byla celá kódová struktura analýz v GEEC. Nastudováno bylo detailní fungování NGSPICE a Maple. Nutností bylo seznámení se s knihovnou VEGA pro generování grafů. Provedl se průzkum ostatních simulátorů, a to nejen webových, za účelem inspirace pro implementaci nových funkcionalit.

Dalo by se říci, že zadání práce bylo obecně splněno. Nicméně aplikace je stále ve fázi vývoje a testování. Nějakou dobu bude ještě trvat, než se změny objeví na stabilním přístupovém bodě <http://asinus.feld.cvut.cz/geec/>.

Ani poté práce na GEEC neskončí. V budoucnu je plánováno další vylepšování a přidávání nových funkcionalit. V kapitole 9 byla zmíněna interaktivní analýza, která se bude dále vylepšovat. Měl by se zmodernizovat celkový vzhled aplikace. Struktura analýz je připravená i na možné přidání nového symbolického simulátoru. Stejně jako SPICE OPUS i NGSPICE v sobě zahrnuje nadstavbu XSPICE, která podporuje pokročilejší součástky jako digitální obvody nebo zdroje napětí typu *filesource*, umožňující například práci se zvukovými signály. Plánováno je přidání právě takovýchto součástek.

Byla vynaložena snaha o celkové vylepšení aplikace GEEC s cílem ještě většího využití ve výuce i mimo ni. Databáze GEEC čítá stovky uložených obvodů, k nimž lze přistupovat speciálním odkazem, které lze vkládat do prezentací a elektronických materiálů. Tohoto konceptu hojně využívá doc. Hospodka a z vlastní zkušenosti lze prohlásit, že je výrazným pomocníkem při studiu elektrických obvodů.





## Literatura

- [1] BUKOVSKÝ, Ondřej. *Analýzy v grafickém editoru elektrotechnických obvodů*. ČVUT, 2019. Bakalářská práce. ČVUT FEL.
- [2] SÜLI, Endre a David F. MAYERS. *An Introduction to Numerical Analysis* [online]. Cambridge University Press, 2012 [cit. 2021-5-15]. ISBN 9780511801181. Dostupné z: doi:10.1017/CBO9780511801181
- [3] *Beautiful HTML5 Charts & Graphs* [online]. c2021 [cit. 2021-5-15]. Dostupné z: <https://canvasjs.com/>
- [4] TUMA, Tadej a Árpád BÜRMEIN. *Circuit simulation with SPICE OPUS: theory and practice*. Boston: Birkhäuser, c2009. Modeling and simulation in science, engineering & technology. ISBN 9780817648671.
- [5] *Circuit Simulator Applet* [online]. c2021 [cit. 2021-5-15]. Dostupné z: <https://www.falstad.com/circuit/>
- [6] *D3: Data-Driven Documents* [online]. c2019 [cit. 2019-4-16]. Dostupné z: <https://d3js.org/>
- [7] *EasyEDA - Online PCB design & circuit simulator* [online]. c2021 [cit. 2021-5-15]. Dostupné z: <https://easyeda.com/>
- [8] HAVLÍČEK, Václav, Martin POKORNÝ a Ivan ZEMÁNEK. *Elektrické obvody*. 2. vyd. Praha: České vysoké učení technické v Praze, 2014-. ISBN 978-80-01-05492-5.
- [9] MCCALLA, W. a D. PEDERSON. Elements of Computer-Aided Circuit Analysis. *IEEE Transactions on Circuit Theory* [online]. 1971, **18**(1), 14-26 [cit. 2021-5-15]. ISSN 0018-9324. Dostupné z: doi:10.1109/TCT.1971.1083238
- [10] *EveryCircuit* [online]. c2021 [cit. 2021-5-15]. Dostupné z: <https://everycircuit.com/>
- [11] PAULŮ, Filip. *Grafický editor elektrotechnických obvodů*. ČVUT, 2015. Diplomová práce. ČVUT FEL.

- [12] SYNOPSISYS®. *HSPICE® User Guide: Basic Simulation and Analysis*. Mountain View, CA: Synopsys, 2014.
- [13] *Chart.js / Open source HTML5 Charts for your website* [online]. c2021 [cit. 2021-5-15]. Dostupné z: <https://www.chartjs.org/>
- [14] *Charts / Google Developers* [online]. c2021 [cit. 2021-5-15]. Dostupné z: <https://developers.google.com/chart/>
- [15] *JavaScript Charts & Maps - amCharts* [online]. c2021 [cit. 2021-5-15]. Dostupné z: <https://www.amcharts.com/>
- [16] *Maplesoft: Software for Mathematics, Online Learning, Engineering* [online]. c2019 [cit. 2019-4-2]. Dostupné z: <https://www.maplesoft.com/>
- [17] MADEC, Morgan, Christophe LALLEMENT, Jacques HAIECH a Jun MA. Modeling and simulation of biological systems using SPICE language. *PLOS ONE* [online]. 2017, **12**(8) [cit. 2021-5-15]. ISSN 1932-6203. Dostupné z: doi:10.1371/journal.pone.0182385
- [18] BIOLEK, Dalibor. *Modelování a simulace v mikroelektronice*. VUT FEKT Brno, 2005.
- [19] *Multisim Live Online Circuit Simulator* [online]. c2021 [cit. 2021-5-15]. Dostupné z: <https://www.multisim.com/>
- [20] BIOLEK, Dalibor. *Navrhování elektronických obvodů počítačem: Počítačové modelování, analýza a simulace*. VA Brno, c2004.
- [21] DOBEŠ, Josef. *Návrh vysokofrekvenčních a mikrovlnných obvodů počítačem*. Praha: Vydavatelství ČVUT, 2003. ISBN 80-01-02696-5.
- [22] *Ngspice, the open source Spice circuit simulator* [online]. c2021 [cit. 2021-5-15]. Dostupné z: <http://ngspice.sourceforge.net/>
- [23] VOGT, Holger, Marcel HENDRIX a Paolo NENZI. *Ngspice User-s Manual*. <http://ngspice.sourceforge.net/docs.html>, 2020.
- [24] NAVARA, Mirko a Aleš NĚMEČEK. *Numerické metody*. Praha: Vydavatelství ČVUT, 2003. ISBN isbn80-01-02689-2.
- [25] VONDRÁK, V. a L. POSPÍŠIL. *Numerické metody 1*. VŠB-TU Ostrava, <http://mi21.vsb.cz/modul/numericke-metody-1>.
- [26] *Online circuit simulator & schematic editor - CircuitLab* [online]. c2021 [cit. 2021-5-15]. Dostupné z: <https://www.circuitlab.com/>
- [27] *PartSim* [online]. c2021 [cit. 2021-5-15]. Dostupné z: <https://www.partsim.com/>

- [28] HOSPODKA, Jiří a Jan BÍČÁK. PraCAn — Maple Package for Symbolic Circuit Analysis. In: *Digital Technologies 2008* [online]. Žilinská univerzita, Elektrotechnická fakulta, 2008 [cit. 2019-4-14]. ISBN 978-80-8070-953-2.
- [29] *PrimeSim HSPICE the Gold Standard for Accurate Circuit Simulation* [online]. c2012 [cit. 2021-5-15]. Dostupné z: <https://www.synopsys.com/implementation-and-signoff/ams-simulation/primesim-hspice.html>
- [30] *Recharts* [online]. c2021 [cit. 2021-5-15]. Dostupné z: <https://recharts.org/en-US/>
- [31] BIOLEK, Dalibor. *SNAP: program pro analýzu lineárních obvodů* [online]. [cit. 2021-5-15]. Dostupné z: <https://user.unob.cz/biolek/vyukaVUT/skripta/priruckaSNAP.pdf>
- [32] *SPICE OPUS* [online]. c2008 [cit. 2021-5-15]. Dostupné z: <http://www.spiceopus.si/>
- [33] CHUNG-WEN HO, A. RUEHLI a P. BRENNAN. The modified nodal approach to network analysis. *IEEE Transactions on Circuits and Systems* [online]. 1975, **22**(6), 504-509 [cit. 2021-5-15]. ISSN 0098-4094. Dostupné z: doi:10.1109/TCS.1975.1084079
- [34] VLADIMIRESCU, Andrei. *The SPICE Book*. John Wiley & Sons, January 1994. ISBN 978-0-471-60926-1.
- [35] *TINACloud on-line Circuit Simulator* [online]. c2020 [cit. 2021-5-15]. Dostupné z: <https://www.tina.com/tinacloud/>
- [36] *Vega* [online]. c2021 [cit. 2021-5-15]. Dostupné z: <https://vega.github.io/>
- [37] *Victory* [online]. c2021 [cit. 2021-5-15]. Dostupné z: <https://formidable.com/open-source/victory/>
- [38] *WinSpice Circuit Simulátor* [online]. c2016 [cit. 2021-5-15]. Dostupné z: <http://www.winspice.com/index.html>
- [39] COX, F.L., W.B. KUHN, H.W. LI, J.P. MURRAY, S.D. TYNOR a M.J. WILLIS. *XSPICE: Software user's manual*. Atlanta, Georgia: Georgia Tech Research Corporation, 1992.
- [40] WANG, Ruye. *Newton-Raphson method* [online]. 2015 [cit. 2021-5-15]. Dostupné z: <http://fourier.eng.hmc.edu/e176/lectures/NM/node20.html>





## Seznam použitých zkratk

Zkratka	Význam
AC	Alternating current
AJAX	Asynchronous JavaScript and XML
BDF	Backward Differentiation Formula
CSS	Cascading Style Sheets
CSV	Comma Separated Values
DC	Direct current
FFT	Fast Fourier Transform
FPGA	Field Programmable Gate Array
GEEC	Graphic editor of electrical circuits
GEM	Gaussova eliminací metoda
LTE	Local Truncation Error
HDL	Hardware Description Languages
HTML	HyperText Markup Language
HUS	Harmonický ustálený stav
MMUN	Modifikovaná metoda uzlových napětí
MUN	Metoda uzlových napětí
JPEG	Joint Photographic Experts Group
ODE	Obyčejná diferenciální rovnice
OP	Operating point
OPUS	OPTimalization UtilitieS
PDF	Portable Document Format
PHP	Personal Home Page
PNG	Portable Network Graphics
PraCAN	Prague Circuit Analyser
PZ	Pole-Zero
SPICE	Simulation Program with Integrated Circuit Emphasis
SUS	Stacionární ustálený stav
SVG	Scalable Vector Graphics
TF	Transfer function
THD	Total Harmonic Distortion
UIC	Use initial conditions
VHDL	VHSIC Hardware Description Language