

Análise Intervalar de Circuitos Elétricos

P.S. GRIGOLETTI¹, G.P. DIMURO², L.V. BARBOZA³, R.H.S. REISER⁴, Programa de Pós-Graduação em Informática, Escola de Informática, Escola de Engenharia e Arquitetura, Universidade Católica de Pelotas, 96010-000, Pelotas, RS, Brasil.

Resumo. Este trabalho apresenta uma aplicação da Matemática Intervalar à análise de circuitos elétricos, onde consideram-se as incertezas nos dados de entrada devido às tolerâncias existentes nos resistores. Sendo assim, os valores correspondentes às resistências dos resistores são modelados como dados intervalares cujos pontos médios representam as resistências nominais e os raios são especificados pelas tolerâncias percentuais dos resistores. O problema é modelado com a utilização da técnica de análise nodal para circuitos elétricos, que gera um sistema de equações lineares intervalares, nas quais as tensões nodais intervalares do circuito são as incógnitas. A metodologia foi implementada na linguagem Python produzindo uma ferramenta denominada *Free Interval Circuit Analyser* (FINCA). São apresentados resultados para circuitos-teste com algumas topologias, comparando-os com os obtidos através da utilização de um software comercial que executa a análise de circuitos elétricos de forma convencional.

1. Introdução

A modelagem de eventos físicos através de técnicas computacionais possui uma grande limitação em termos da confiabilidade dos parâmetros utilizados. Os dados utilizados na realização da modelagem de qualquer evento físico são obtidos, via de regra, por medições. Estes dados apresentam intrinsecamente um erro, quer seja por falha humana (erro de leitura), quer seja por deficiência mecânica nos próprios instrumentos (erro do instrumento) [9, 10]. Por outro lado, a representação numérica utilizada nos computadores digitais nem sempre permite que os números sejam armazenados com exatidão [10]. Isto pode gerar erros de arredondamento e/ou truncamento, tanto na modelagem quanto na execução dos cálculos numéricos [9].

Devido principalmente a estes fatores, modelar eventos físicos se torna uma tarefa na qual apenas resultados aproximados podem ser obtidos. Em alguns casos, ocorrem aproximações bastante pobres, o que torna necessário a execução de tarefas auxiliares de modo que se possa analisar os erros associados ao problema. Este tipo de análise, além de dispendir um considerável esforço computacional, não é uma tarefa simples de ser operacionalizada.

¹pablogri@atlas.ucpel.tche.br.

²liz@atlas.ucpel.tche.br.

³luciano@atlas.ucpel.tche.br, também atua no Centro Federal de Educação Tecnológica de Pelotas.

⁴reiser@atlas.ucpel.tche.br.

Neste contexto, a Matemática Intervalar [11] se apresenta como uma ferramenta bastante poderosa na análise e controle de erros em computação científica [8]. Pela sua natureza de tratar os números não mais como entes pontuais, e sim como intervalos que encapsulam os números pontuais, pode-se, de uma forma segura, armazenar os dados físicos através de uma medida provável e um possível percentual de erro associado. Assim, ao final do processo matemático-computacional, tem-se uma estimativa da influência dos erros de entrada no resultado final obtido.

Especificamente na área de Engenharia Elétrica, a maioria dos componentes utilizados possui um valor nominal e uma tolerância associada (dados fornecidos pelos fabricantes). O valor nominal corresponde ao provável valor que o componente deve apresentar e a tolerância indica o quanto o valor real do componente pode variar em torno de seu valor nominal. Por exemplo, considerando um resistor de $10\text{ k}\Omega - 5\%$, tem-se que o seu valor nominal corresponde a $10\text{ k}\Omega$, entretanto o seu verdadeiro valor pode ficar situado entre $9,5$ e $10,5\text{ k}\Omega$.

Além disso, as medidas de correntes e tensões são obtidas através da leitura de instrumentos de medição. Estes instrumentos apresentam intrinsecamente um erro associado às medidas. Portanto, por maior que seja a precisão do instrumento, o valor efetivamente medido pode não ser encontrado corretamente.

Sob este paradigma, o presente trabalho apresenta um software para ser utilizado na análise de circuitos elétricos, o qual considera a possível existência de erros de medição como também da tolerância em alguns componentes elétricos. Os dados de entrada são modelados como variáveis intervalares de modo que estes erros sejam encapsulados pelos dados intervalares e seja possível, ao final da análise, avaliar a sua influência no resultado final. A implementação da ferramenta foi realizada na linguagem de programação Python, com características de software livre. Para tanto, foi implementada uma biblioteca Python para Matemática Intervalar.

Optou-se por utilizar uma metodologia conhecida como *análise nodal* [7] para realizar a análise elétrica do circuito. Esta técnica de análise gera um sistema de equações lineares a ser resolvido para a obtenção do resultado final. Neste caso, o sistema linear consiste em um sistema intervalar. Com a aplicação da análise nodal, as incógnitas intervalares são as tensões (ou voltagens) em todos os nós do circuito (pontos do circuito elétrico onde se conectam dois ou mais elementos).

Este trabalho está organizado como segue. Na seção 2 é realizada uma revisão sobre a técnica da análise nodal. Na seção 3 é apresentada uma revisão sobre a Matemática Intervalar e resolução de sistemas lineares intervalares. Na seção 4 apresenta-se a ferramenta FINCA (*Free Interval Circuit Analyser*). A seção 5 mostra alguns resultados obtidos com a aplicação do software FINCA, com a comparação dos resultados obtidos com os resultados fornecidos por um software comercial. Na seção 6 são apresentadas as conclusões e perspectivas futuras.

2. Análise Nodal Aplicada a Circuitos Elétricos

Um circuito elétrico é composto por fontes (de tensão e de corrente), elementos passivos (resistores) e elementos de ligação (conexões elétricas). Cada conexão entre elementos de circuito é chamada *nó*. A literatura apresenta várias técnicas

matemáticas para a modelagem de circuitos elétricos, entre outras podem ser citadas [7]: análise nodal, análise de malhas e transformação de fontes. Neste estudo, foi escolhida a técnica de análise nodal por se tratar da mais adequada para a implementação computacional. Uma vez conhecidos os elementos componentes do circuito elétrico, uma leitura sucessiva destes componentes permite a modelagem do respectivo circuito elétrico.

A análise nodal se baseia na conexão entre os diferentes tipos de elementos do circuito e possui o seu referencial teórico na *Lei das correntes de Kirchhoff*, a qual estabelece que “a soma das correntes que chegam em um nó é igual à soma das correntes que saem deste nó”. Esta lei nada mais é do que a lei da conservação da energia elétrica. Baseadas neste princípio, as equações nodais para cada nó do circuito elétrico podem ser resumidas como [7]:

Lado esquerdo: soma de produtos condutância por tensão. Um destes produtos corresponde à soma de todas as condutâncias conectadas ao nó em análise pela respectiva tensão do nó. Os demais correspondem ao simétrico da tensão de nós adjacentes multiplicados pela condutância que conecta estes dois nós.

Lado direito: corresponde à soma de correntes provenientes de fontes de corrente. Considera-se a corrente positiva se esta chegar ao nó em análise. Caso contrário, esta será considerada negativa.

Observa-se que pela formulação desta técnica de análise, as incógnitas do sistema linear a ser resolvido correspondem às tensões nodais. Considerando-se, por exemplo, o circuito elétrico da Figura 1, utilizando-se o nó “0” como referência para as tensões, tem-se as seguintes equações nodais:

$$\begin{aligned} \left(\frac{1}{R_1} + \frac{1}{R_2}\right) V_A - \frac{1}{R_2} V_B &= I_1 \\ -\frac{1}{R_2} V_A + \left(\frac{1}{R_2} + \frac{1}{R_3} + \frac{1}{R_4}\right) V_B - \frac{1}{R_4} V_C &= 0 \\ -\frac{1}{R_4} V_B + \left(\frac{1}{R_4} + \frac{1}{R_5}\right) V_C &= I_2. \end{aligned}$$

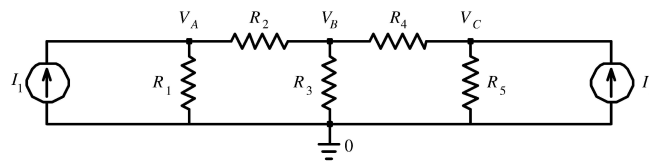


Figura 1: Circuito elétrico hipotético.

Neste exemplo, tem-se um circuito elétrico com quatro nós: A , B , C e 0 . Tomando-se o nó “0” como referência, obtém-se um sistema linear de três equações com três incógnitas para a determinação das outras tensões nodais (V_A , V_B e V_C).

3. A Matemática Intervalar

A Matemática Intervalar é uma teoria matemática introduzida na década de 60 [11], que tem como objetivo fundamental o tratamento automático e garantido de erros

em computação científica, onde os parâmetros e dados iniciais geralmente são incertos. As técnicas intervalares baseiam-se na definição da aritmética intervalar, do produto escalar ótimo [10] e do princípio da máxima exatidão, garantido pelos arredondamentos direcionados.

Define-se um *intervalo real* X como um conjunto não vazio de números reais $X = [x_1; x_2] = \{x \in \mathbb{R} \mid x_1 \leq x \leq x_2\}$, onde x_1 é o extremo inferior e x_2 é o extremo superior. Denota-se por \mathbb{IR} a coleção de todos os intervalos reais.

Para obter uma representação \tilde{X} de um intervalo real $X = [x_1; x_2] \in \mathbb{IR}$ em um sistema de ponto flutuante de uma máquina, os extremos x_1 e x_2 devem ser arredondados “por falta” e “por excesso”, respectivamente, o que denomina-se *arredondamento direcionado*. Tem-se então que $X \subseteq \tilde{X}$.

O *ponto médio*, o *diâmetro* e o *raio* de um intervalo X são definidos, respectivamente, como $mid(X) = \check{X} = \frac{1}{2}(x_1 + x_2)$, $diam(X) = x_2 - x_1$ e $rad(X) = \frac{1}{2}diam(X)$. Uma notação alternativa para um intervalo X é $X = \langle mid(X), rad(X) \rangle$.

As operações aritméticas intervalares são definidas como $X * Y = \{x * y \mid x \in X, y \in Y\}$, para $*$ $\in \{+, -, \times, \div\}$. Se $X = [x_1; x_2]$ e $Y = [y_1; y_2]$, tem-se que [11]:

$$\begin{aligned} X + Y &= [x_1 + y_1; x_2 + y_2], & X - Y &= [x_1 - y_2; x_2 - y_1] \\ X \times Y &= [\min \rho; \max \rho], & \text{com } \rho &= \{x_1 y_1, x_1 y_2, x_2 y_1, x_2 y_2\} \\ X \div Y &= X \times [y_2^{-1}; y_1^{-1}], & \text{se } 0 &\notin Y. \end{aligned}$$

Observa-se que as propriedades da associatividade e comutatividade são válidas para a adição e multiplicação. Entretanto, exceto em casos especiais, a propriedade da distributividade não é válida, o que pode causar superestimação.

O módulo Python, para Matemática Intervalar, que foi implementado neste trabalho, armazena intervalos tanto na forma de “ínfimo – supremo” como na de “ponto médio – raio”. Esta biblioteca possibilita que as operações intervalares básicas sejam executadas sobre escalares intervalares, assim como vetores e matrizes. As funções intervalares clássicas como, por exemplo, as trigonométricas, exponencial, distância, diâmetro etc., também estão disponíveis. Este módulo também implementa os arredondamentos direcionados.⁵

Um sistema linear intervalar do tipo $Ax = b$ possui a matrix de coeficientes A e os vetores x e b como dados intervalares. Como na aritmética pontual, existem também vários métodos para a resolução de SELA’s intervalares (por exemplo, os apresentados em [4, 6, 11, 12]). Os métodos diretos intervalares tendem a ser uma extensão dos métodos diretos pontuais de resolução de SELA’s, pelo uso de operações a argumentos intervalares. Os estudos sobre versões intervalares para métodos indiretos ou iterativos foram iniciados por E. Hansen [3] e posteriormente continuados por um grande número de pesquisadores [4, 5, 11, 12], inclusive em abordagens de programação paralela [1].

O método utilizado neste projeto, que foi o que apresentou melhores resultados considerando o tipo de matrizes geradas pela aplicação, consiste em multipli-

⁵Observa-se a existência de outras bibliotecas para a Matemática Intervalar como, por exemplo, a biblioteca Intlab [6, 13] para o ambiente Matlab, um software proprietário de alto custo. Nosso trabalho, entretanto, está baseado na concepção de software livre, e priorizou a simplicidade da programação orientada a objetos proporcionada pela linguagem Python, que permite com facilidade o desenvolvimentos de extensões por parte da comunidade científica ao módulo desenvolvido.

car os dois lados de $Ax = b$ por uma matriz Y (uma aproximação da inversa da matriz de pontos médios de A) e definir $E = -I - YA$. Se $\|E\| < 1$, sendo $\|E\| = \max_i \sum_j |E_{ij}|$, então a sequência $X^{(k+1)} = \{Yb + EX^{(k)}\} \cap X^{(k)}$, onde $k = 0, 1, 2, \dots$, com $X_i^{(0)} = [-1; 1] \|Yb\| / (1 - \|E\|)$, onde $i = 1, \dots, n$, irá convergir em um número finito de passos para um vetor intervalar x que contém a solução para $Ax = b$ [11].

O módulo intervalar Python disponibiliza também outras rotinas para a resolução de sistemas lineares intervalares, para métodos diretos e iterativos.

4. O Analisador de Circuitos Elétricos Intervalar

O software *Free Interval Circuit Analyser* (FINCA) é uma ferramenta computacional livre para a análise confiável de circuitos elétricos, utilizando a Matemática Intervalar [11] para o controle automático e rigoroso dos erros nos dados de entrada.

O FINCA foi desenvolvido baseado no paradigma de orientação a objetos (OO), e está organizado conforme o diagrama apresentado na Figura 2, sendo composto basicamente pelas seguintes classes: (i) *circuit*, (ii) *iNodalAnalysis*, (iii) *iIterativeMethod* (Python), (iv) *reportMaker* e (v) *finca*.

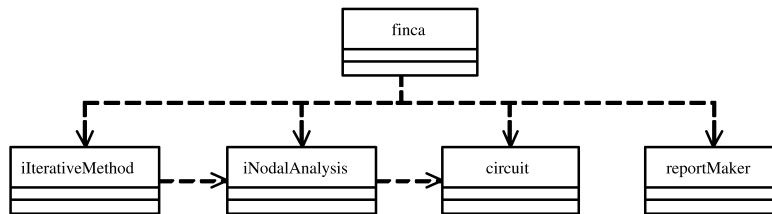


Figura 2: Diagrama de classes da ferramenta FINCA.

A classe *circuit* é utilizada no tratamento da entrada de dados, realizando a leitura das informações (obtidas na interface gráfica ou em um arquivo texto), e posteriormente na análise e na geração das estruturas de dados que serão utilizadas pela ferramenta ao longo de todo processo.

A classe *iNodalAnalysis* utiliza as estruturas de dados geradas pela classe *circuit*. Com estes dados, ela realiza a *análise nodal intervalar* e gera uma matriz intervalar e um vetor intervalar. Internamente, para a realização da análise nodal intervalar, esta classe faz uso do módulo Python para a Matemática Intervalar.

A classe *iIterativeMethod* faz parte do módulo Python para Matemática Intervalar. Esta classe tem como objetivo resolver um SELA intervalar através do método iterativo descrito na seção 3. Este SELA é do tipo $Ax = b$, onde a matriz intervalar A e o vetor intervalar b são fornecidos pela análise nodal intervalar (implementada pela classe *iNodalAnalysis*) e o vetor intervalar x é o resultado que será gerado – as tensões nodais intervalares do circuito analisado.

A classe *reportMaker* é utilizada na geração de relatórios (nos formatos HTML, PDF ou TXT) sobre as atividades realizadas na ferramenta FINCA. Estes relatórios

são baseados nas informações contidas na saída de dados. A classe *finca* gera a interface gráfica e agrupa todas as funcionalidades providas pelo software FINCA.

Na Figura 3 é apresentada a interface gráfica da ferramenta FINCA, que é dividida em três áreas principais: (i) a de entrada de dados, (ii) a de saída de dados e (iii) outra que agrupa funcionalidades diversas (barra de ferramentas).

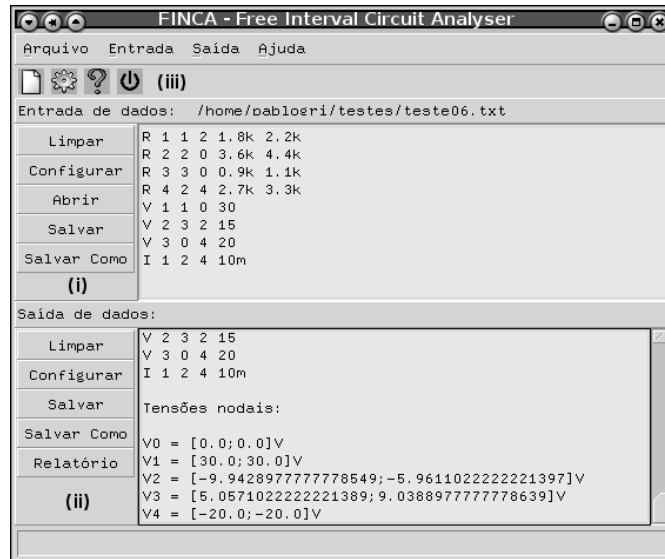


Figura 3: Interface gráfica da ferramenta FINCA.

A inserção dos dados de entrada nesta ferramenta pode ser realizada de duas formas distintas. A definição do circuito elétrico a ser analisado pode ser escrita diretamente na interface gráfica, ou pode ser utilizado um arquivo texto, previamente criado, contendo a topologia do circuito. Na definição do circuito elétrico, cada linha representa um elemento (componente) e possui o formato “elemento numeração nó1 nó2 inf sup”, onde: (i) o campo *elemento* pode ser R (resistor), I (fonte de corrente independente) ou V (fonte de tensão independente), (ii) o campo *numeração* é um índice que identifica cada elemento, (iii) os campos *nó1* e *nó2* são os números dos nós terminais entre os quais o elemento está conectado, e (iv) os campos *inf* e *sup* representam os extremos inferior e superior do intervalo, respectivamente, que representa o valor do elemento.

A Figura 4 apresenta uma interface de configuração da entrada de dados. Nesta se define qual será o formato do intervalo que representa o valor de cada elemento do circuito elétrico. Dependendo do perfil do usuário, os formatos utilizados podem variar. Por exemplo, um aluno do curso de Engenharia Elétrica, provavelmente, preferirá trabalhar com intervalos no formato `valor percentual de variação`.

A saída de dados do FINCA também pode ser configurada pelo usuário. Dentre as opções disponíveis, estão *tensões nodais* e *correntes nos elementos*, como pode ser visualizado na Figura 5. Esta saída pode ser salva em um arquivo texto. Todos os resultados numéricos são apresentados na forma de intervalos.

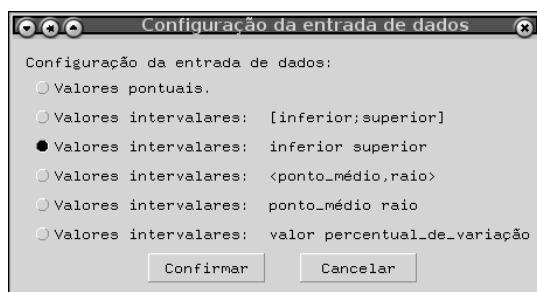


Figura 4: Entrada de dados.

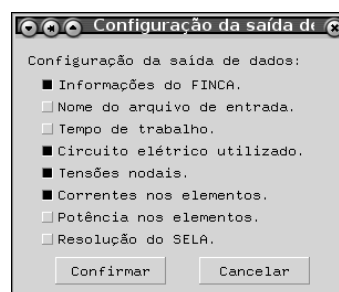


Figura 5: Saída de dados.

A ferramenta FINCA oferece a opção de geração de relatórios sobre os circuitos analisados. Estes relatórios utilizam as informações contidas na saída de dados. A interface gráfica de configuração dos relatórios pode ser visualizada na Figura 6. Um exemplo de relatório no formato PDF é apresentado na Figura 7.

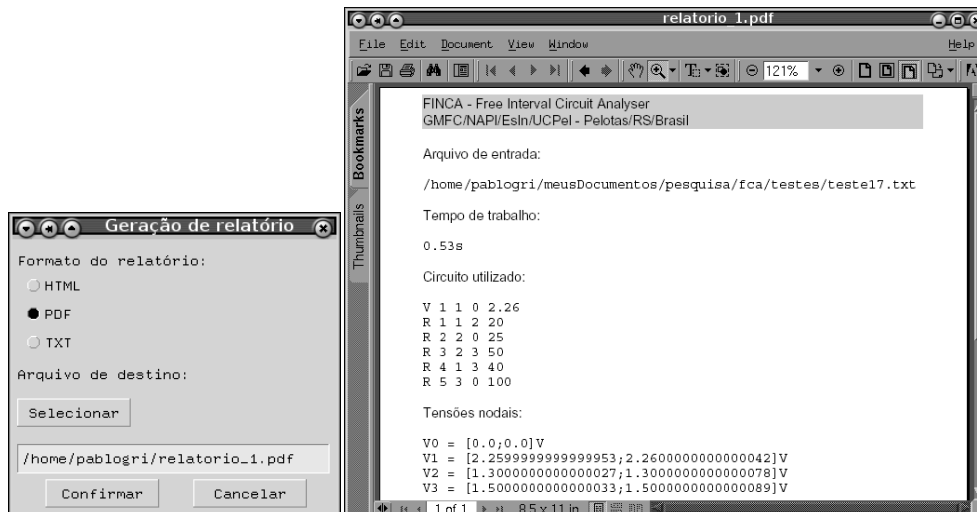


Figura 6: Interface de geração de relatório.

Figura 7: Exemplo de relatório gerado pela ferramenta FINCA.

5. Exemplos de Aplicação

Para testar o desempenho do software proposto, nesta seção serão apresentados resultados numéricos obtidos para a aplicação da ferramenta FINCA a dois circuitos elétricos. A Figura 8 mostra o circuito elétrico para o Teste 1, sendo os valores dos resistores apresentados na Tabela 1. Nesta tabela, a coluna 2 (VN) refere-se ao valor nominal do resistor e a coluna 3, à tolerância. Os resultados obtidos pela ferramenta

FINCA e pelo software comercial *PSpice*® estão apresentados na Tabela 3(a). Com a ferramenta FINCA, as tensões nos nós 2 e 3 devem se situar entre 1.93473 e 2.06527 V e 5.93473 e 6.06527 V, respectivamente. A tensão no nó 1 é mantida fixa em 14 V pois, até o presente momento, está se considerando apenas a incerteza nos dados dos resistores. Comparando os resultados obtidos, observa-se que o resultado intervalar fornecido pela FINCA encapsula o resultado pontual dado pelo software comercial, o qual é apenas uma aproximação do resultado verdadeiro, podendo apresentar um erro que é difícil ou impossível de ser estimado. Dessa forma, em termos de qualidade de solução, os resultados gerados pelo módulo intervalar são aceitáveis e automaticamente já oferecem uma análise do erro presente na solução.

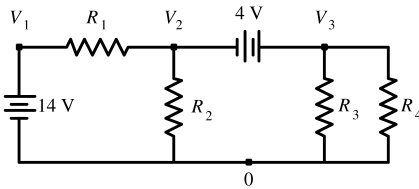


Figura 8: Circuito elétrico do Teste 1.

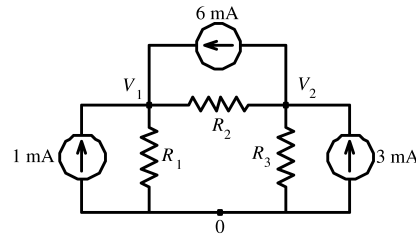


Figura 9: Circuito elétrico do Teste 2.

Tabela 1: Valor dos resistores no Teste 1.

Resistor	Valor Nominal	Tolerância
1	4 Ω	10 %
2	2 Ω	10 %
3	12 Ω	10 %
4	4 Ω	10 %

Tabela 2: Valor dos resistores no Teste 2.

Resistor	Valor Nominal	Tolerância
1	6 k Ω	1 %
2	4 k Ω	1 %
3	2 k Ω	1 %

O circuito elétrico do Teste 2 está mostrado na Figura 9, sendo os valores dos resistores apresentados na Tabela 2. A aplicação do FINCA e do *PSpice*® ao circuito resultou nos valores apresentados na Tabela 3(b). Com o FINCA, observa-se que erros de 1% nos valores nominais das resistências do circuito afetam os valores das tensões nodais em, aproximadamente, 2% para o nó 1 e de 9% para o nó 2. Salienta-se que o FINCA forneceu valores intervalares para as tensões nodais que encapsulam os resultados pontuais do software comercial, e, ao mesmo tempo, fornecem uma estimativa para o erro máximo que pode ter ocorrido.

6. Conclusões

Este trabalho apresentou um software para a análise de circuitos elétricos baseado na filosofia de software livre e na Matemática Intervalar. Este é capaz de avaliar automaticamente a influência das tolerâncias dos valores nominais dos resistores sobre

Tabela 3: Resultados dos testes realizados.

(a) Teste 1			
FINCA		PSPICE [®]	
Entrada de Dados	Tensões Nodais	Entrada de Dados	Tensões Nodais
R 1 1 2 3.96 4.04	V0 = [0.0;0.0] V	R 1 1 2 4	V0 = 0.0 V
R 2 2 0 1.98 2.02	V1 = [14.0;14.0] V	R 2 2 0 2	V1 = 14.0 V
R 3 3 0 11.88 12.12	V2 = [1.93473;2.06527] V	R 3 3 0 12	V2 = 2.0 V
R 4 3 0 3.96 4.04	V3 = [5.93473;6.06527] V	R 4 3 0 4	V3 = 6.0 V
V 1 1 0 14		V 1 1 0 14	
V 2 3 2 4		V 2 3 2 4	

(b) Teste 2			
FINCA		PSPICE [®]	
Entrada de Dados	Tensões Nodais	Entrada de Dados	Tensões Nodais
R 1 1 0 5.94k 6.06k	V0 = [0.0;0.0] V	R 1 1 0 6k	V0 = 0.0 V
R 2 1 2 3.96k 4.04k	V1 = [17.69134;18.27228] V	R 2 1 2 4k	V1 = 18.0 V
R 3 2 0 1.98k 2.02k	V2 = [1.81476;2.18484] V	R 3 2 0 2k	V2 = 2.0 V
I 1 0 1 1m		I 1 0 1 1m	
I 2 2 1 6m		I 2 2 1 6m	
I 3 0 2 3m		I 3 0 2 3m	

as tensões nodais do circuito elétrico. A implementação foi realizada na linguagem Python, na qual também se implementou um módulo para a Matemática Intervalar. Este módulo, que implementa os arredondamentos direcionados, disponibiliza as operações aritméticas intervalares, as funções intervalares clássicas (diâmetro, distância, trigonométricas, exponencial etc.), assim como operações com matrizes intervalares e métodos diretos e iterativos para SELA's intervalares. Testes e comparações deste módulo são apresentados em [2].

Atualmente, além do Curso de Engenharia Elétrica da UCPel, cursos técnicos oferecidos no SENAI/RS também estão utilizando o software FINCA, onde, inclusive, já foram desenvolvidas algumas extensões. Como trabalhos futuros, pretende-se implementar outros tipos de elementos e desenvolver uma interface gráfica para a entrada e saída dos dados dos circuitos.

Agradecimentos

Os autores agradecem as valiosas observações e sugestões dos revisores. Este trabalho foi parcialmente financiado pelo programa CTINFO/CNPq e FAPERGS.

Abstract. This work presents an application of Interval Mathematics techniques to the analysis of electrical circuits where uncertainties due to the resistor tolerances are provided by the manufacturers. The values of the resistor electrical resistances are modelled as interval data whose midpoints represent nominal resistances and radii are specified by the resistor tolerances. The problem is modelled using the nodal analysis technique for electrical circuits, generating a system of interval linear equations, whose solution gives the circuit nodal interval voltages. The methodology was implemented using Python, producing the *Free Interval Circuit Analyser*

(FINCA). Results for test-circuits with different topologies are presented, comparing them with punctual results provided by commercial software for conventional electrical circuit analysis.

Referências

- [1] T. Beelitz, B. Lang, C.H. Bischof, Efficient task scheduling in the parallel result-verifying solution of nonlinear systems, *Reliable Computing*, **12**, No. 2 (2006), 141–151.
- [2] P.S. Grigoletti, “Uma ferramenta computacional intervalar para a análise confiável de circuitos elétricos”, ESIN/UCPel, Pelotas, 2004.
- [3] E.R. Hansen, On solving systems of equations using interval arithmetic, *Mathematics of Computation*, **22**, No. 102 (1968), 374–384.
- [4] E.R. Hansen, Bounding the solution of interval linear equations, *SIAM Journal on Numeric Analysis*, **29**, No. 5 (1992), 1493–1503.
- [5] E.R. Hansen, G. W. Walster, Solving overdetermined systems of interval linear equations, *Reliable Computing*, **12**, No. 3 (2006), 239–243.
- [6] G.I. Hargreaves, “Interval Analysis in Matlab”, Manchester Centre for Comp. Math., Manchester, Numerical Analysis Report No. 416, 2002.
- [7] J.D. Irwin, “Análise de Circuitos em Engenharia”, Makron, São Paulo, 2001.
- [8] R.B. Keafort, V. Kreinovich (eds.), “Applications of Interval Computations”, Kluwer, Boston, 1996.
- [9] U. Kulisch, W.L. Miranker (eds.), “A New Approach to Scientific Computation”, Academic Press, New York, 1983.
- [10] U. Kulisch, Advanced arithmetic for the digital computer, design of arithmetic units, *Electronic Notes in Theoretical Computer Science*, **24** (1999).
- [11] R.E. Moore, “Methods and Applications of Interval Analysis”, SIAM, Philadelphia, 1979.
- [12] A. Neumaier, “Interval Methods for Systems of Equations”, Cambridge University Press, Cambridge, 1990.
- [13] S. M. Rump, IntLab - Interval Laboratory, in T. Csendes (ed), “Developments in Reliable Computing”, Kluwer, Dordrecht, 1999.