

UMA FERRAMENTA PARA A ANÁLISE DE CIRCUITOS ELÉTRICOS

José Ubirajara N. de Nunes¹, Pablo S. Grigoletti²,
Graçaliz P. Dimuro, Luciano V. Barboza, Renata H. S. Reiser
Universidade Católica de Pelotas - UCPel
Rua Félix da Cunha, 412 - CEP 96010-000 - Pelotas / RS
{junnunez,pablogri,liz,lvb,reiser}@ucpel.tche.br

Resumo. *O projeto tem como objetivo desenvolver um software para a análise de circuitos elétricos. Este programa servirá à concepção de software livre, sendo que Python foi a linguagem escolhida para sua implementação. Foi desenvolvido um módulo para a resolução de sistemas de equações lineares algébricas (SELA's). Este módulo foi implementado utilizando o método da decomposição LU por possuir maior robustez numérica minimizando, por exemplo, os erros de arredondamento ocasionados pelas divisões por números muito pequenos. Implementou-se também um módulo para a análise de circuitos elétricos, baseado na análise nodal. Atualmente, está em desenvolvimento uma nova versão da decomposição LU, utilizando técnicas intervalares para que os erros sejam automática e rigorosamente controlados. Como trabalhos futuros, serão desenvolvidos uma interface gráfica e uma versão web para o sistema.*

Palavras-chave: *Análise de circuitos elétricos, Decomposição LU, Python*

1. INTRODUÇÃO

Um dos grandes problemas enfrentados pelos alunos do curso de Engenharia Elétrica é o alto custo dos softwares utilizados. É importante salientar também que existem poucos programas de qualidade baseados na filosofia de software livre, desenvolvidos para a área de análise de circuitos elétricos.

Outros problemas enfrentados na área de computação numérica são os erros gerados pela incerteza dos dados de entrada, bem como, erros oriundos de arredondamentos e truncamentos. São esses processos que causam a perda da exatidão dos resultados teoricamente esperados.

Por estes motivos, o objetivo geral do Projeto onde está inserido este trabalho é desenvolver uma ferramenta computacional para auxiliar na análise de circuitos elétricos, baseada na concepção de software livre, e utilizando técnicas intervalares no controle dos erros de computação numérica. Este projeto recebe apoio financeiro do programa CTINFO/CNPq e FAPERGS.

Em particular, este artigo descreve a primeira etapa deste Projeto. Esta consiste no desenvolvimento de um software livre para aplicação em análise de circuitos elétricos em sua versão pontual.

¹Bolsista de Iniciação Científica PROBIC/FAPERGS

²Bolsista de Iniciação Científica IC/CNPq

2. ANÁLISE DE CIRCUITOS ELÉTRICOS

A análise de circuitos elétricos utiliza-se fundamentalmente de métodos da Álgebra Linear que podem exigir um grande esforço computacional. As técnicas para análise de circuitos mais conhecidas são: *análise de malhas* e *análise nodal*, que se baseiam nas Leis de Kirchhoff [1].

Estes métodos de análise geram sistemas de equações lineares de n equações e n incógnitas, cuja solução estima os valores das incógnitas de um circuito elétrico, que podem ser de dois tipos: tensão ou corrente.

Ao percorrer as malhas de um circuito elétrico, obtém-se, como incógnitas, as tensões sobre os elementos que compõe as malhas (*Lei das Tensões de Kirchhoff*). Por outro lado, se for realizada uma análise das correntes que entram ou saem de um nó, têm-se, como incógnitas, as correntes que percorrem os ramos do circuito (*Lei das Correntes de Kirchhoff*).

Por motivos computacionais, foi escolhido implementar o sistema utilizando a *técnica de análise nodal*. Observa-se que é exigido muito mais esforço computacional para identificar as malhas de um circuito do que seus nós.

Nesta versão do software, os circuitos analisados podem conter apenas resistores e fontes independentes de corrente e de tensão.

2.1. Análise nodal

A análise nodal é um método de análise de circuitos baseado na Lei das Correntes de Kirchhoff. Nela deve-se escolher um ponto como referência (ponto 0) e após arbitrar, aos demais nós, as respectivas tensões do circuito a serem calculadas. No lado esquerdo das equações nodais, tem-se o somatório dos produtos *condutância \times tensão*. Do lado direito dessas equações tem-se uma corrente positiva (proveniente de uma fonte de

corrente) se a corrente estiver chegando ao nó, e uma corrente negativa, se a corrente estiver saindo do nó.

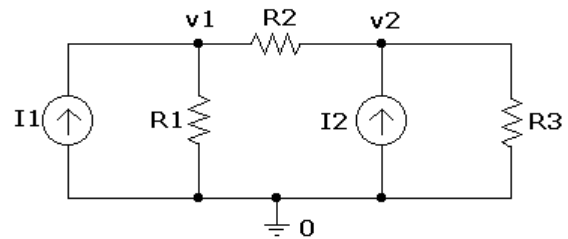


Figura 1. Circuito elétrico sem fontes de tensão.

Análise nodal com fontes de tensão. No método de análise nodal, a utilização de *fontes de tensão* nos circuitos elétricos introduz equações vinculares. Quando a fonte de tensão estiver com um de seus terminais ligados a referência (que é arbitrada), então se tem um *nó aparente*. Quando esta não possui nenhum de seus terminais ligados à referência, então se possui um *supernó*.

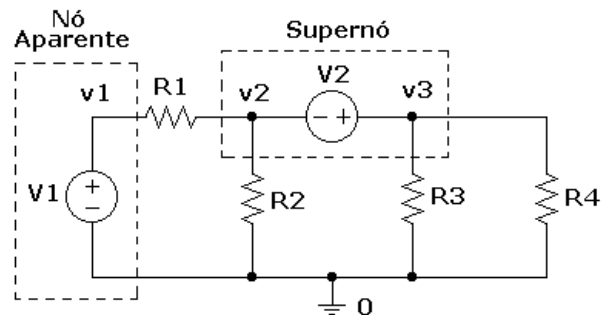


Figura 2. Circuito elétrico com fontes de tensão.

3. RESOLUÇÃO DE SELA'S

A *análise nodal* [1] é uma técnica de análise de circuitos que gera um número de equações igual ao número de nós (incógnitas) menos um (nó de referência).

Sabe-se que no caso de sistemas $n \times n$, a solução de um sistema do tipo $Ax=b$ é dado por $x=A^{-1}b$, onde os valores encontrados para x , correspondem à solução do sistema. Porém

o cálculo da matriz inversa A^{-1} requer um grande esforço computacional, pelo número de operações envolvidas e pela complexidade dessas operações.

Portanto, para a resolução de SELA's no aplicativo em desenvolvimento, foram estudados alguns métodos, por exemplo, eliminação de Gauss e decomposição LU.

3.1. Eliminação de Gauss

Eliminação Gaussiana [2] é a técnica mais conhecida e mais usada para a resolução de SELA's densos de pequeno a médio porte (dimensão máxima 30). Este método consiste de uma aplicação esquemática das propriedades básicas da álgebra linear. O *Método de Gauss* transforma o sistema linear original num sistema linear equivalente, cuja matriz dos coeficientes é triangular superior. A resolução deste sistema equivalente é imediata.

O método divide-se em duas etapas, o que torna a implementação mais fácil. A primeira etapa é chamada de *triangularização* e consiste em transformar a matriz dos coeficientes A numa matriz triangular superior, mediante permutações e combinações lineares. Observa-se que as operações realizadas sobre a matriz A devem também ser aplicadas sobre o vetor independente b . A segunda etapa do método é conhecida como *substituição inversa* e tem por objetivo o cálculo dos componentes do vetor x , solução de $Ax=b$, a partir da solução imediata do último componente de x , e a substituição inversa nas equações anteriores.

3.2. Decomposição LU

A *Decomposição LU* [3] consiste na decomposição da matriz A em um produto de duas matrizes triangulares e, em seguida, na solução de dois sistemas triangulares que fornecem a solução do sistema original.

Considerando um sistema do tipo $Ax=b$, e sendo A uma matriz quadrada, então se pode

escrever $A=LU$, onde L é uma matriz triangular inferior unitária e U uma matriz triangular superior.

Se $Ax=b$ e $A=LU$ então $LUx=b$ e considerando que $Ux=y$, obtém-se dois novos sistemas lineares:

$$Ly=b \quad (1)$$

$$Ux=y \quad (2)$$

A partir da solução da Eq. (1), são obtidos os componentes de y , e, portanto, com a substituição do vetor y na Eq. (2), obtém-se a solução desta, encontrando os valores dos componentes do vetor x . A Eq. (1) é solucionada por *substituição direta* e a Eq. (2) por *substituição inversa*.

Na prática, na busca de soluções para minimizar a instabilidade numérica, deve-se optar por uma modificação deste método, denominada *Decomposição LUP* [4]. O objetivo desta decomposição é encontrar três matrizes L , U e P , com dimensão $n \times n$, de modo que $PA = LU$, onde L é uma matriz triangular inferior unitária, U é uma matriz triangular superior e P é uma matriz de permutação (inicialmente uma matriz identidade).

Considerando um sistema do tipo $Ax=b$ e multiplicando ambos os membros da equação por uma matriz P , obtém-se $P Ax=P b$ e se $PA=LU$, então $LUx=P b$. Considerando $Ux=y$, obtém-se:

$$Ly=Pb \quad (3)$$

$$Ux=y \quad (4)$$

Na Eq. (3), tem-se um sistema triangular inferior e obtém-se a solução para o vetor y através de uma substituição direta. De posse da solução para y , resolve-se o sistema da Eq. (4). Este sistema é triangular superior e dele obtém-se a solução para o vetor x através de uma substituição inversa.

Na prática, observa-se que com o pivotamento, a matriz A (matriz original) será

alterada em função das trocas de linhas, portanto o vetor b também deveria ser modificado para que a solução do sistema não se altere. É para resolver este problema que existe a matriz P , para poder realizar no vetor b todas as trocas de linha que ocorreram no processo de decomposição da matriz A .

Portanto, a decomposição LUP oferece uma maior estabilidade e robustez numérica na resolução dos SELA's do que os métodos anteriores (método de Gauss e decomposição LU), permitindo o pivotamento parcial de linhas e possui um esforço computacional equivalente.

4. POR QUE USAR PYTHON?

Para a implementação, escolheu-se a linguagem *Python* [5], por possuir estruturas de dados de alto-nível eficientes e adotar uma abordagem simples e efetiva para a programação orientada a objetos. Sua sintaxe elegante e tipagem dinâmica, em adição a sua natureza interpretada, tornam Python ideal para *scripting* e para o desenvolvimento rápido de aplicações (RAD - Rapid Application Development) em diversas áreas e na maioria das plataformas. O interpretador de Python e sua extensa biblioteca padrão estão disponíveis na forma de código fonte ou binário para a maioria das plataformas, e podem ser distribuídos livremente.

Além disso, este interpretador é facilmente extensível incorporando novas funções e tipos de dados implementados em C ou C++ (ou qualquer outra linguagem acessível a partir de C), seja para desempenhar operações críticas em máxima velocidade, ou para vincular programas Python a bibliotecas que só estejam disponíveis em formato binário.

Esta linguagem permite organizar programas em módulos que podem ser reutilizados em outros programas escritos em Python. Além disso, a linguagem Python possui vários módulos desenvolvidos para a área de

computação científica. No desenvolvimento do *analisador de circuitos elétricos* foi utilizada sua biblioteca numérica.

5. ESTUDO DE CASO

A entrada de dados é feita a partir de um arquivo texto especificando o tipo de elemento (resistor, fonte de tensão ou de corrente), a sua numeração, os terminais entre os quais o elemento está conectado e o seu valor. Considere o exemplo da Fig. 3.

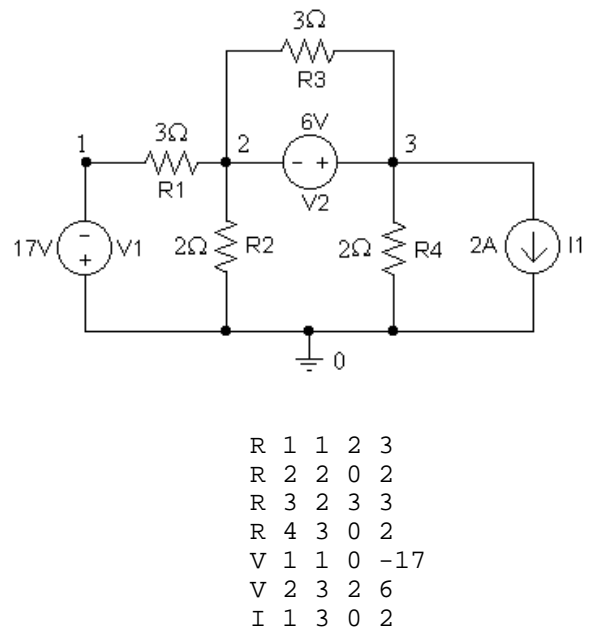


Figura 3. Circuito elétrico e entrada de dados em arquivo texto para o exemplo.

Cada linha do arquivo texto da Fig. 3 é uma lista composta da seguinte forma:

[elemento,numeração,nó1,nó2,valor], onde elemento pode ser resistor (R), fonte de tensão (V) ou fonte de corrente (I); numeração é o índice que identifica o elemento; nó1 e nó2 são os terminais entre os quais o elemento está conectado; valor é o valor numérico do elemento.

Na saída, teremos como resposta, as tensões nodais do circuito, que são as tensões de cada ponto (nó) do circuito em relação à referência (ponto 0). Neste exemplo, as

incógnitas são as tensões V_1 , V_2 e V_3 . A Fig. 4 apresenta o arquivo de saída, onde o vetor X consiste dos valores dessas tensões.

O elemento R1 é um resistor.
Está entre os nós 1 e 2.
Possui valor 3 ohm(s).

O elemento R2 é um resistor.
Está entre os nós 0 e 2.
Possui valor 2 ohm(s).

O elemento R3 é um resistor.
Está entre os nós 2 e 3.
Possui valor 3 ohm(s).

O elemento R4 é um resistor.
Está entre os nós 0 e 3.
Possui valor 2 ohm(s).

O elemento V1 é uma fonte de tensão.
Está entre os nós 0 e 1.
Possui valor 17 volt(s).

O elemento V2 é uma fonte de tensão.
Está entre os nós 2 e 3.
Possui valor -6 volt(s).

O elemento I1 é uma fonte de corrente.
Está entre os nós 3 e 0.
Possui valor 2 ampere(s).

Vetor X:

[-17. -8. -2.]

Figura 4. Arquivo texto de saída para o exemplo da Fig. 3.

7. CONSIDERAÇÕES FINAIS

Os estudos desenvolvidos propiciaram a implementação de uma versão inicial do sistema para a análise de circuitos elétricos.

Como continuidade do trabalho, pretende-se aplicar as técnicas da Matemática Intervalar [6], para possibilitar o tratamento da incerteza dos dados de entrada, com o controle rigoroso e automático dos erros de computações numéricas.

Pretende-se desenvolver uma interface gráfica para o software, semelhante as dos

programas comerciais utilizados na análise de circuitos elétricos. Para a implementação desta interface, será utilizada a biblioteca gráfica wxPython.

Sendo o Python uma linguagem interpretada, com a utilização de uma biblioteca especial, é possível compilar seus programas. Com o programa compilado, não existe a necessidade de ter a linguagem Python e nem as bibliotecas utilizadas pelo aplicativo instaladas no computador. Neste contexto, outro trabalho a ser desenvolvido futuramente é disponibilizar o sistema em uma versão compilada.

Propõe-se também o desenvolvimento de uma versão para web, de tal forma que o analisador possa ser utilizado principalmente em cursos on-line. A versão web será uma segunda alternativa para quem não possui a linguagem Python instalada.

8. REFERÊNCIAS

- [1] D. E. Johnson, J. L. Hilburn, J. R. Johnson, Fundamentos de Análise de Circuitos Elétricos, LTC, Rio de Janeiro: 1994, p. 1-90.
- [2] D. M. Claudio, J. M. Marins, Cálculo Numérico Computacional: Teoria e Prática, Atlas, São Paulo: 2000.
- [3] H. Anton, Álgebra Linear com Aplicações, Bookman, São Paulo: 2001, p. 321-332.
- [4] T. H. Cormen, C. E. Leiserson, R. L. Rivest, Algoritmos: Teoria e Prática, Campus, Rio de Janeiro: 2002, p. 585-596.
- [5] M. C. Brown, Python, McGraw-Hill Companies, s.l: 2001.
- [6] R. E. Moore, Methods and Applications of Interval Analysis, SIAM, Philadelphia: 1979.