

IMPLEMENTAÇÃO COMPUTACIONAL DO ALGORITMO DE FLUXO DE POTÊNCIA PARA ANÁLISE DE SISTEMAS ELÉTRICOS EM REGIME PERMANENTE

1 INTRODUÇÃO

O conhecimento dos principais fundamentos relacionados ao planejamento e operação de sistemas de energia elétrica é imprescindível para a formação dos engenheiros que trabalham em empresas do setor de geração, transmissão, distribuição e utilização de eletricidade e em indústrias em geral. Neste contexto, destaca-se os algoritmos de fluxo de potência, que desempenham papel fundamental para a análise de regime permanente em redes de potência.

No entanto, somente a apresentação teórica de conceitos relativos a esses sistemas já não satisfaz mais as atuais exigências do setor elétrico, que cada vez mais requer aperfeiçoamentos com a inclusão de modernas técnicas de simulação computacional. Dessa forma, torna-se essencial aplicar métodos e rotinas computacionais eficientes, além de linguagens de programação cada vez mais didáticas, à solução das equações iterativas de fluxo de potência em redes elétricas, em especial aquelas de grande dimensão.

Tradicionalmente, a forma empregada para expor estes conceitos aos alunos tem sido através da utilização de alguns pacotes computacionais de análise existentes, como o Power Systems Analysis Toolbox (PSAT) – toolbox do MatLab, e o ANAREDE (Análise de Redes Elétricas) – desenvolvido pelo Centro de Pesquisas em Energia Elétrica (CEPEL). Porém, o acesso e alteração do código-fonte desses programas não é permitido aos usuários, o que dificulta bastante a compreensão de sua funcionalidade principalmente por parte de estudantes acadêmicos.

Assim, observou-se a necessidade de um enorme esforço por parte dos alunos de graduação em Engenharia da Computação do Instituto Federal do Triângulo Mineiro (IFTM) Campus Avançado Parque Tecnológico para assimilar os conceitos relacionados ao desenvolvimento do algoritmo de fluxo de potência, somando-se ainda o tempo reduzido devido às exigências de outras disciplinas do curso. Isso levou os estudantes interessados a possuírem pouco domínio destas ferramentas de análise e, conseqüentemente, a apresentarem grande dificuldade na realização de pesquisas em sistemas elétricos de potência.

Neste sentido, este projeto teve como objetivo a implementação computacional do algoritmo de fluxo de potência para análise de sistemas elétricos em regime permanente. Para tal, fez-se uso de uma linguagem de programação moderna, didática e eficiente para solução de métodos numéricos, além de uma plataforma gratuita, possibilitando que o código-fonte esteja disponível para futuros estudos na área por parte de alunos do próprio IFTM/CAUPT ou de outras instituições. Ao final, de modo a verificar a eficiência do modelo desenvolvido, os resultados foram comparados com aqueles disponíveis por outro software já consolidado no setor elétrico para um mesmo estudo de caso.

2 MATERIAIS E MÉTODOS

A implementação do algoritmo foi realizada utilizando a linguagem de programação Python, por ser uma alternativa de código-fonte aberto, pela facilidade de utilização, sintaxe intuitiva e devido à grande quantidade de bibliotecas disponíveis para seu desenvolvimento, além da enorme comunidade de programadores que a utilizam.

O primeiro passo realizado foi a instalação do editor de código-fonte, a opção escolhida foi o Visual Studio Code, desenvolvido pela Microsoft, assim como a instalação da linguagem Python na máquina. A seguir, foi realizado o desenvolvimento do algoritmo, iniciando-se pelo desenvolvimento das equações de fluxo de potência e métodos numéricos de solução. O método utilizado para o desenvolvimento do algoritmo foi o Método de Newton-Rapshon. Por fim, foi realizada a parametrização dos dados de entrada e forma de visualização dos resultados fornecidos pelo algoritmo.

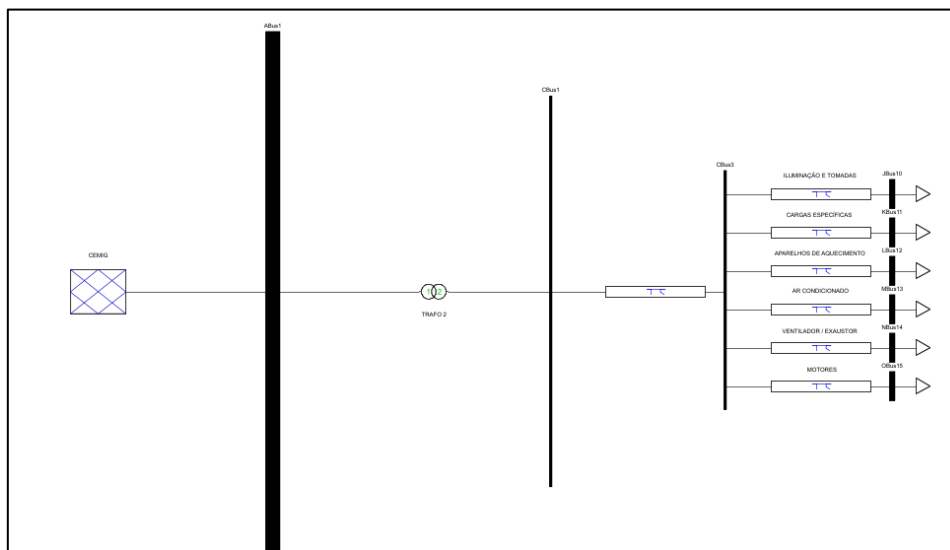
Após a etapa de parametrização e entrada dos dados, foi efetuada uma verificação acerca de possíveis erros no modelo. Posteriormente, foi executada a simulação de fluxo potência para uma rede fictícia e, em seguida foi feita a simulação para a rede modelada em projeto anterior utilizando o campus IFTM/CAUPT para comparação e validação dos resultados com software já reconhecido no setor elétrico nacional: o Power Systems Analysis Toolbox (PSAT) – toolbox do MatLab. Os dados das simulações foram coletados e analisados em software de planilha computacional. Dessa maneira, foi possível validar os resultados encontrados na implementação do algoritmo proposto.

3 RESULTADOS

A implementação do software foi realizada como previsto, utilizando o método de Newton-Rapshon para soluções numéricas. Os dados das simulações foram coletados e analisados em software de planilha computacional, possibilitando a validação dos resultados encontrados no desenvolvimento do algoritmo proposto. Os resultados foram comparados com o software já consolidado e se mostraram bem próximos, o que nos diz mais ainda sobre a eficácia do software desenvolvido.

Abaixo, pode-se verificar as implementações e amostragens dos dois softwares: o primeiro, a biblioteca PSAT, com o sistema do IFTM/CAUPT, sendo modelado em projeto já realizado anteriormente, demonstrado pela figura 1. Logo abaixo, os resultados da modelagem sendo representados para a biblioteca PSAT através da figura 2. A seguir, observa-se, o segundo, a implementação do software sendo desenvolvida em Python, representada pela figura 3, utilizando como Potência Base nos dois casos 100 MVA e as demais informações necessárias como cargas de iluminações e tomadas, cargas específicas, entre outros tipos de cargas do sistema do IFTM. Os resultados obtidos durante o desenvolvimento do software podem ser visualizados através da figura 4 abaixo.

Figura 1 – Modelagem do sistema do campus UPT/IFTM - Software PSAT.



Fonte: Software de modelagem computacional (autoria própria).

Figura 2 – Resultados - Modelagem: Campus UPT/IFTM - Software PSAT.

Bus	Vm [kV]	Va [deg]	P [MW]	Q [MVar]
[1]-ABus1	13.8	0	0.16439	0.06766
[2]-CBus1	0.21998	-0.00903	0	0
[3]-CBus3	0.21998	-0.00994	0	0
[4]-JBus10	0.21998	-0.0104	-0.08472	-0.0361
[5]-KBus11	0.21998	-0.00997	-0.006	-0.00255
[6]-LBus12	0.21998	-0.00997	-0.0055	0
[7]-MBus13	0.21998	-0.01025	-0.05659	-0.02411
[8]-NBus14	0.21998	-0.00995	-0.00202	-0.00087
[9]-OBus15	0.21998	-0.00999	-0.00956	-0.00407

Fonte: Software de modelagem computacional (autoria própria).

É possível constatar, na figura 3, a chamada da classe Newton(), representando o Método de Newton-Rapshon implementado em linguagem Python. Além disso, pode-se verificar as demais informações das barras sendo atribuídas ao método setBarrasSistema(), no qual é definida a numeração e tipo das barras: tensão e ângulo ou barras do tipo PV e PQ, além das tensões e ângulos e a definição do tipo de barra, seja ela de carga ou geração para realização correta do cálculo iterativo do fluxo de potência. A seguir, é mostrado o valor da potência base aparente e demais barras do sistema através da utilização do método mostrarBarras()).

Após isso, é utilizado o método setCalBarrasEspecf() para o cálculo de potência especificada em cada uma das barras. Os valores são calculados automaticamente e, após isso, são plotadas na tela. O próximo método utilizado é o de ligacoesBarras(), realizando ligações nas barras específicas do sistema para dar prosseguimento ao cálculo do fluxo de potência. Logo após, é utilizado o método mostrarLigacoes() para amostragem de informações referentes às ligações das barras do sistema. O próximo método implementado no software é o ResolucaoCircuito(), método para resolução genérica do circuito de fluxo de potência, utilizando parâmetros como erro ou quantidade de iterações, lista tensões e ângulos a serem calculadas representado as barras do tipo PQ e PV.

Por fim, são utilizados, respectivamente, os métodos FluxoPotenciaAparente(), PerdasSistema() e mostrarBarras() para cálculo do fluxo de potência aparente em todas as ligações do sistema, cálculo de perdas do sistema levando em consideração a soma de todas as potências e o método para amostragem das informações com as potências já analisadas e calculadas.

Figura 3 – Modelagem do sistema do campus UPT/IFTM - Implementação do Software.

```
SistemaIF = Newton()

SistemaIF.setBarrasSistema(1, 1, 1.00, 0, 0 + 0 * 1j, 0 + 0 * 1j)
SistemaIF.setBarrasSistema(2, 2, 1.00, 0, 0 + 0 * 1j, 0 + 0 * 1j)
SistemaIF.setBarrasSistema(3, 2, 1.00, 0, 0 + 0 * 1j, 0 + 0 * 1j)
SistemaIF.setBarrasSistema(4, 2, 1.00, 0, 84.72e3 + 36.098e3 * 1j, 0 + 0 * 1j)
SistemaIF.setBarrasSistema(5, 2, 1.00, 0, 6e3 + 2.55e3 * 1j, 0 + 0 * 1j)
SistemaIF.setBarrasSistema(6, 2, 1.00, 0, 5.5e3 + 0 * 1j, 0 + 0 * 1j)
SistemaIF.setBarrasSistema(7, 2, 1.00, 0, 56.59e3 + 24.11e3 * 1j, 0 + 0 * 1j)
SistemaIF.setBarrasSistema(8, 2, 1.00, 0, 2.02e3 + 0.87e3 * 1j, 0 + 0 * 1j)
SistemaIF.setBarrasSistema(9, 2, 1.00, 0, 0.56e3 + 4.07e3 * 1j, 0 + 0 * 1j)

SistemaIF.mostrarBarras()
SistemaIF.setCalBarrasEspecf()

SistemaIF.ligacoesBarras(1, 2, impedancia=0.01 + 0.1j)
SistemaIF.ligacoesBarras(2, 3, impedancia=0.001 + 0.01j)
SistemaIF.ligacoesBarras(3, 4, impedancia=0.001 + 0.01j)
SistemaIF.ligacoesBarras(3, 5, impedancia=0.001 + 0.01j)
SistemaIF.ligacoesBarras(3, 6, impedancia=0.001 + 0.01j)
SistemaIF.ligacoesBarras(3, 7, impedancia=0.001 + 0.01j)
SistemaIF.ligacoesBarras(3, 8, impedancia=0.001 + 0.01j)
SistemaIF.ligacoesBarras(3, 9, impedancia=0.001 + 0.01j)

SistemaIF.mostrarLigacoes()

SistemaIF.ResolucaoCircuito(erro=1e-05, listaBarrasTensoes=[2, 3, 4, 5, 6, 7, 8, 9], listaBarrasAngulos=[2, 3, 4, 5, 6, 7, 8, 9])
SistemaIF.FluxoPotenciaAparente(printTensoes=True, printCorrentes=True)
SistemaIF.PerdasSistema()
SistemaIF.mostrarBarras()
```

Fonte: Implementação computacional do algoritmo de fluxo de potência (autoria própria).

Figura 4 – Resultados - Modelagem: Campus UPT/IFTM - Implementação do Software.

```
##### PERDAS DO SISTEMA: #####
(7.203938160075889e-08+7.203938160043363e-07j) [pu]

##### IMPLEMENTAÇÃO DO SOFTWARE: #####
##### REDE IFTM/CAUPT - UNIDADE 1: #####
Valor Base S (Potência Aparente) = 100000000.0 VA
{'codigo': 1, 'tensao': 1.0, 'angulo': 0.0, 'carga': 0j, 'geracao': (0.0016439360198640673+0.0006773401969433047j)}
{'codigo': 2, 'tensao': 0.9999158390432163, 'angulo': -0.00015763346725316478, 'carga': 0j, 'geracao': 0j}
{'codigo': 2, 'tensao': 0.9999074243142066, 'angulo': -0.00017339827335139134, 'carga': 0j, 'geracao': 0j}
{'codigo': 2, 'tensao': 0.999902966848784, 'angulo': -0.00018151083149118283, 'carga': (0.0008472+0.00036098j), 'geracao': 0j}
{'codigo': 2, 'tensao': 0.9999071092847782, 'angulo': -0.00017397287991665977, 'carga': (6e-05+2.55e-05j), 'geracao': 0j}
{'codigo': 2, 'tensao': 0.9999073693089602, 'angulo': -0.00017394837522908844, 'carga': (5.5e-05+0j), 'geracao': 0j}
{'codigo': 2, 'tensao': 0.9999044471150471, 'angulo': -0.0001788171927570522, 'carga': (0.0005659+0.0002411j), 'geracao': 0j}
{'codigo': 2, 'tensao': 0.9999073171042513, 'angulo': -0.00017359160916686066, 'carga': (2.02e-05+8.7e-06j), 'geracao': 0j}
{'codigo': 2, 'tensao': 0.9999069216670021, 'angulo': -0.00017431374330419702, 'carga': (9.56e-05+4.07e-05j), 'geracao': 0j}
```

Fonte: Implementação computacional do algoritmo de fluxo de potência (autoria própria).

A seguir, pode-se ver uma comparação sendo realizada de forma mais clara, através da tabela 1. Comparando os resultados sendo modelados em projeto anterior e a implementação do software proposto por esse projeto, na primeira barra é possível verificar que a potência ativa do sistema do IFTM é de 164,39 kW no PSAT e o mesmo valor pode ser visualizado no software implementado; já para a potência reativa, o valor encontrado é de 67,66 kVAR no PSAT e um valor muito próximo no software implementado de 67,73 kVAR.

Em relação à potência aparente, o resultado obtido foi de 177,77 kVA no PSAT e 177,8 kVA no software implementado.

Além disso, é possível visualizar outros resultados, como o da barra 4 no software implementado sendo de 84,72 kW e o mesmo valor para a biblioteca PSAT.

Tabela 1 – Tabela comparativa da implementação do sistema do campus UPT/IFTM.

RELATÓRIO ESTÁTICO - FLUXO DE POTÊNCIA: CAMPUS UPT/IFTM							
Power System Analysis Toolbox (PSAT)				Implementação do Software			
BARRA	POTÊNCIA ATIVA (kW)	POTÊNCIA REATIVA (kVar)	POTÊNCIA APARENTE (KVA)	BARRA	POTÊNCIA ATIVA (kW)	POTÊNCIA REATIVA (kVar)	POTÊNCIA APARENTE (KVA)
ABarra 1	184,39	67,66	177,77	Barra 1	184,39	67,73	177,80
BBarra 2	0,00	0,00	0,00	Barra 2	0,00	0,00	0,00
CBarra 3	0,00	0,00	0,00	Barra 3	0,00	0,00	0,00
JBarra 10	-84,72	-36,10	92,09	Barra 4	-84,72	-36,10	92,09
KBarra 11	-8,00	-2,55	8,52	Barra 5	-8,00	-2,55	8,52
LBarra 12	-5,50	0,00	5,50	Barra 6	-5,50	0,00	5,50
MBarra 13	-56,59	-24,11	61,51	Barra 7	-56,59	-24,11	61,51
NBarra 14	-2,02	-0,87	2,20	Barra 8	-2,02	-0,87	2,20
OBarra 15	-9,56	-4,07	10,39	Barra 9	-9,56	-4,07	10,39

Fonte: Software de planilha computacional (autoria própria).

Dessa forma, os valores calculados mostram que os resultados estão bem próximos, para não dizer os mesmos, por questões de aproximações decimais. Por fim, é visto que na implementação do software as potências estão sendo representadas com sinais negativos, indicando que essas barras em questão são barras de carga e não de geração, assim como é representado na biblioteca PSAT.

4 DISCUSSÃO

A forma empregada para expor conceitos relacionados a Sistemas Elétricos de Potência a alunos do ensino superior tem sido através da utilização de alguns pacotes computacionais de análise existentes, como o Power Systems Analysis Toolbox (PSAT) – toolbox do MatLab, e o ANAREDE (Análise de Redes Elétricas) – desenvolvido pelo Centro de Pesquisas em Energia Elétrica (CEPEL). Porém, o acesso e alteração do código-fonte desses programas não é permitido aos usuários, o que dificulta bastante a compreensão de sua funcionalidade, principalmente por parte de estudantes acadêmicos.

Portanto, verificou-se um grande esforço por parte dos graduandos do curso de Engenharia de Computação do Instituto Federal do Triângulo Mineiro (IFTM) Campus Avançado Uberaba Parque Tecnológico para absorver conceitos relacionados ao desenvolvimento do algoritmo de fluxo de potência, havendo ainda outras exigências que demandam tempo e esforço relacionadas às outras disciplinas do curso. Consequentemente, isso fez com que os alunos interessados tivessem pouco domínio sobre essas ferramentas de análise, encontrando grandes dificuldades ao conduzir pesquisas de sistemas de elétricos de potência.

Desse modo, a implementação de um algoritmo de fluxo de potência para análise em regime permanente foi vista como necessária, para melhor explorar e facilitar o entendimento dos estudantes, pois através do modelo implementado o acesso a seu código fonte foi disponibilizado para realização de possíveis testes e implementações de diferentes redes elétricas.

Os resultados alcançados e as análises desenvolvidas ao longo da pesquisa permitiram concluir que o desenvolvimento do programa retratou um bom comportamento do sistema quando comparado com software já estabelecido no mercado para um mesmo estudo de caso. Os dados obtidos foram organizados em planilhas computacionais para realização de comparações e estudos futuros.

Existem também outras atualizações possíveis de serem realizadas em projetos posteriores, como o desenvolvimento de uma interface gráfica para que a visualização dos dados se torne mais amigável e a implementação do Fluxo de Potência Ótimo, que leva em consideração diversos outros fatores importantes e essenciais para a realização do cálculo do fluxo de potência.

Além disso, reitera-se a importância desse projeto de iniciação científica, que representa uma maneira de aproximar os alunos da vida profissional e científica. Desse modo, o aluno tem maior motivação para continuar seus estudos, promovendo, assim, seu crescimento como acadêmico.

REFERÊNCIAS

- ALEXANDER, C. K.; Sadiku, M. N. O.. **Fundamentos de Circuitos Elétricos**, 5ª edição, McGraw Hill, 2013.
- BARROSO, L. C. et al. **Cálculo Numérico com Aplicações**. 2 ed. São Paulo: Harbra, 1987.
- BOYLESTAD, R. L.. **Introdução à Análise de Circuitos Elétricos**, 12ª edição, Pearson, 2012.
- CLÁUDIO, D. M.; MARINS, J. M.. **Cálculo Numérico Computacional - Teoria e Prática**. 1 ed. São Paulo: Atlas, 1989.
- DEL TORO, V.. **Electric Machines and Transformers**. Prentice Hall do Brasil, Rio de Janeiro-RJ, 1994.
- FITZGERALD, A. E.. **Máquinas Elétricas**. McGraw-Hill, 1979.
- FRANCO, N. B.. **Cálculo Numérico**. 1 ed. São Paulo: Pearson Prentice Hall, 2007.
- RUGGIERO, M. A. G.; LOPES, V. L.. **Cálculo Numérico: Aspectos Teóricos e Computacionais**. 2 ed. São Paulo: Pearson Makron Books, 1998.
- SANTOS, M. **Fluxo De Potência por Newton-Raphson em Python**. Youtube, 2 de mar. de 2021. Disponível em: <<https://www.youtube.com/playlist?list=PLNOc51Usq3jgqOe4zmCNrXJ71SOBpa-TI>>. Acesso em: 15 jul. de 2021.
- MONTICELLI, Alcir José; GARCIA, Arioaldo. **Introdução a Sistemas de Energia Elétrica**. Campinas, SP: Ed. UNICAMP, c2011. VIII.

COMPUTATIONAL IMPLEMENTATION OF THE POWER FLOW ALGORITHM FOR ANALYSIS OF ELECTRICAL SYSTEMS IN PERMANENT REGIME

Knowledge of the main fundamentals related to the planning and operation of electric power systems is essential for the training of engineers working in the electricity sector. In this one, the power flow algorithms are highlighted, which play a great role for the steady-state analysis in electrical networks. Currently, only theoretical knowledge is not enough to achieve the desired results, thus, it is essential to apply efficient computational methods, in addition to didactic programming languages, to the solution of iterative power flow equations. Traditionally, some computer packages are used to simulate power flow in networks, such as the Power Systems Analysis Toolbox (PSAT), however, the licenses of the mentioned software do not allow the edition of the source code, thus making it difficult to understand. Thus, it has been observed the need for an enormous effort on the part of undergraduate students in Computer Engineering at the IFTM Campus UPT to assimilate the concepts related to the implementation of this type of algorithm and, thus, carry out research in the area. In this sense, this project aimed at the computational implementation of the power flow algorithm for the analysis of electrical systems in steady state. To this end, the source code editor Visual Studio Code, developed by Microsoft, was used, in addition to the Python programming language, as it is an open-source option allowing future studies in the area by students from the IFTM itself. CAUPT or other institutions. Ease of use was another decisive factor in choosing the language, in addition to the intuitive syntax and large number of available libraries. The power flow, as it is a non-linear problem, requires that its solution be based on numerical methods, with emphasis on the Newton-Raphson algorithm. In the end, it was possible to verify the efficiency of the model developed from the comparison of its results with those available by other software already consolidated in the electricity sector for the same case study.

Keywords: *Power Flow, Numerical Method, Newton-Raphson, Python, Visual Studio Code.*