HABILIDADES DE RESOLUÇÃO DE PROBLEMAS COM MATHCAD PRIME SOB O ESTÍMULO DO PENSAMENTO COMPUTACIONAL

Eduardo H. Banaczewski *– eduardohbanaczewski@aluno.santoangelo.uri.br*
Universidade Regional Integrada do Alto Uruguai e das Missões
Rua Universidade das Missões, 464.
98.802-470 – Santo Ângelo – RS

Eliani Retzlaff *– elianir@san.uri.br*
Rua Universidade das Missões, 464.
98.802-470 – Santo Ângelo – RS

Iuri C. Figueiró *– iuricastroff@san.uri.br*
Rua Universidade das Missões, 464.
98.802-470 – Santo Ângelo – RS

Nelson K. Neto *– nelsonknak@san.uri.br*
Rua Universidade das Missões, 464.
98.802-470 – Santo Ângelo – RS

Resumo: *A capacidade de cálculo oferecida pelos softwares matemáticos permite a simulação de problemas, processo que envolve a modelagem matemática, a implementação de equações e análise dos resultados produzidos. Essa aplicabilidade e automatização de cálculos remetem elementos importantes do pensamento computacional. Por uma experiência vivenciada no desenvolvimento de um tópico de estudo do componente curricular Cálculo Numérico Computacional, do curso de Engenharia Elétrica, através da resolução de um problema de fluxo de potência, pretende-se enfatizar a potencialidade do software Mathcad Prime para o desenvolvimento do pensamento computacional. O planejamento das aulas tem como eixo norteador a Resolução de Problemas, cuja experiência realizada transforma-se em aprendizagem através de um ciclo de aprendizagem envolvendo as seguintes etapas: compreensão do problema, seguida da elaboração de um plano e execução, e por fim, a verificação dos resultados. Entre as atividades propostas nesse componente curricular, destaca-se que a resolução de problemas acontece de forma transversal, estabelecendo a relação entre o estudo teórico e aplicação prática, onde a matemática envolvida foi favorecida pela área da computação.*

Palavras-chave: Métodos Numéricos. Pensamento Computacional. Resolução de Problemas.

1 INTRODUÇÃO

A Matemática é usada como uma ferramenta essencial que permite, entre outros, subsídios para visualização gráfica, análise e representação de modelos necessários para resolução de problemas, processos fundamentais para o bom desempenho profissional nas mais diversas áreas do conhecimento.

No que tange o estudo dos Métodos Numéricos, apresentam caráter transversal, pois podem ser utilizados para a obtenção da solução de problemas reais. O processo de resolução envolve o levantamento de dados, construção do modelo matemático, escolha de um método, implementação computacional e análise de resultado. As ferramentas matemáticas dispostas nos métodos numéricos envolvem operações aritméticas simples (porém repetitivas), tornando possível o desenvolvimento de algoritmos, nas mais variadas linguagens de programação.

As ferramentas tecnológicas evidenciam participação efetiva na educação, com a tendência de buscar um ensino mais dinâmico e que propicie a construção de conhecimento de forma autônoma e interativa. Desta forma, os softwares baseados em álgebra computacional tornam-se grandes aliados. A sua utilização na resolução de problemas é indissociável, pois embora se realize cálculos manuais, desenvolver resultados em termos computacionais permite explorar domínios como a área de análise numérica, problemas esses que muitas vezes não poderiam ser abordados sem a tecnologia. Nesse sentido, observa-se uma importante relação entre o desenvolvimento de algoritmos para resolução de problemas e o conceito de pensamento computacional. Wing [7] aponta competências relacionadas à abstração e decomposição de problemas de forma a permitir sua resolução usando recursos computacionais e estratégias algorítmicas. Pelas condições ora apresentadas, esse trabalho constituiu-se a partir de aulas de Cálculo Numérico Computacional, do Curso de Engenharia Elétrica, da Universidade Regional Integrada do Alto Uruguai e das Missões (URI). As experiências vivenciadas utilizando o software *Mathcad Prime*, tem o propósito de promover o envolvimento dos acadêmicos tornando-os ativos no seu processo de ensino e aprendizagem e, buscar maior interação entre professores e acadêmicos dos cursos de engenharias pela resolução de problemas na área.

2. PENSAMENTO COMPUTACIONAL E A RESOLUÇÃO DE PROBLEMAS

A definição operacional de pensamento computacional que se adota nesse trabalho é referenciada pela *International Society for Technology in Education* (ISTE) e *Computer Science Teachers Association* (CSTA) [2], como um processo de resolução de problemas que inclui: a) formulação de problemas de forma que permita a usar o computador e outras ferramentas para contribuir na solução; b) organização e análise lógica dos dados; representação dos dados por meio de abstrações, como modelos e simulações; c) automatização de soluções por meio do pensamento algorítmico; d) identificação, análise e implementação de possíveis soluções com o objetivo de alcançar o máximo de combinação eficiente e eficaz de etapas e recursos; e) generalização e transferência deste processo de resolução de problemas para uma grande variedade de problemas.

Ao ministrar o componente curricular de Cálculo Numérico Computacional, observa-se que muitos acadêmicos dos cursos de engenharias apresentam dificuldades em relação a conceitos de pensamento computacional. Esses conceitos envolvem a coleta, análise e representação de dados, modelação de problemas, abstração, algoritmos e simulação – elementos importantes para o desenvolvimento de habilidades como o raciocínio lógico e a resolução de problemas. Para que esse processo tenha êxito, segue-se de conceitos previamente aprendidos, buscando criar estratégias para a solução. No entanto, além do conhecimento matemático, necessitam pensar em termos abstratos e hipotéticos, utilizando-se da lógica para ordenar o pensamento. A estratégia de produzir algoritmos busca contribuir para a estruturação do pensamento lógico, constituindo-se o pensamento computacional pode servir para a resolução de muitos outros problemas. Cabe como exemplo, a resolução de um sistema linear de ordem 3, generalizando para a ordem n. Autores como Sica [5], apontam que o raciocínio lógico e o Pensamento Computacional deveriam ser ensinados desde cedo, pois aumentam a capacidade de dedução e conclusão de problemas. Para Valente [6], o processo de criação de um programa para a resolução de um problema acontece por intermédio de um ciclo de ações descrição-execução-reflexão-depuração.

Ao desenvolver o algoritmo na forma de um programa específico, em um *software* matemático, pode-se observar a fase da descrição. O resultado gerado pelo computador pode ser obtido pela execução dos comandos utilizados. Com isso, o acadêmico poderá realizar a ação de reflexão, abstraindo o que se tem e o esperado, avaliando se o objetivo foi atingido ou o programa necessita depuração, para a busca de melhores resultados numéricos.

3. SOFTWARE *MATHCAD PRIME* E OS MÉTODOS NUMÉRICOS

Embora os métodos numéricos tenham sido concebidos antes dos computadores, ambos se relacionam à interdisciplinaridade, onde a colaboração entre a matemática e a tecnologia da informação aumenta a capacidade de compreensão e desenvolvimento de princípios científicos até então difíceis de serem estudados ou compreendidos [1]. O uso de *softwares* matemáticos específicos para o desenvolvimento de métodos numéricos possibilita a construção de programas que, através de passos lógicos e sequências programadas, efetuam os cálculos relacionados aos mais diversos problemas. Nesse contexto, o *Mathcad Prime* é um *software* baseado em Álgebra computacional, que permite a entrada e manipulação de equações matemáticas, realização de cálculos, análise de dados e geração de gráficos. Essa combinação torna-o uma importante ferramenta para acadêmicos e profissionais que necessitam da matemática na resolução de problemas, com a possibilidade de redução de erros.

3.1 O ALGORITMO DO MÉTODO GAUSS-SEIDEL

O método iterativo de Gauss-Seidel é utilizado para resolução de sistemas de equações lineares algébricas de ordem n, que consiste em isolar a variável x_1 da primeira equação, a variável x_2 da segunda equação e assim por diante, visto na Figura 1 (a). Supondo $a_{ii} \neq 0, i = 1,2,...,n$, isola-se o vetor mediante a separação pela diagonal da matriz dos coeficientes, obtendo-se o indicado na Figura 1 (b). Caso o sistema garanta convergência, parte-se de uma aproximação inicial (arbitrária), utilizando sempre o último vetor obtido em cada nova iteração k. O processo iterativo é repetido até que o vetor $x^{(k+1)}$ esteja suficientemente próximo do vetor $x^{(k)}$. A relação recursiva é dada pela Equação 1.

$$x_n^{(k+1)} = (b_i - \sum_{\substack{j=1 \\ j<i}}^{i-1} a_{ij} x_j^{(k+1)} - \sum_{\substack{j=i+1 \\ j>i}}^{n} a_{ij} x_j^{(k)}) / a_{ii} \qquad (1)$$

Figura 1. (a) Sistema de equações lineares de ordem "n" (b) Vetor para iterações

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + ...\,..a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + ...\,..a_{2n}x_n = b_2 \\ .... \; .... \; .... \; .... \; .... \; .... \; ...... \\ .... \; .... \; .... \; .... \; .... \; .... \; ...... \\ a_{n1}x_1 + a_{n2}x_2 + a_{n3}x_3 + ...\,..a_{nn}x_n = b_n \end{cases}$$

$$\begin{aligned} x_1^{(k+1)} &= \frac{1}{a_{11}}\left[b_1 - a_{12}x_2^{(k)} - a_{13}x_3^{(k)} - ...... - a_{1n}x_n^{(k)}\right] \\ x_2^{(k+1)} &= \frac{1}{a_{22}}\left[b_2 - a_{21}x_1^{(k+1)} - a_{13}x_3^{(k)} - ...... - a_{1n}x_n^{(k)}\right] \\ x_3^{(k+1)} &= \frac{1}{a_{33}}\left[b_3 - a_{31}x_1^{(k+1)} - a_{32}x_2^{(k+1)} - ...... - a_{1n}x_n^{(k)}\right] \\ &......... \; .... \; .... \; .... \; .... \; .... \; .... \\ x_n^{(k+1)} &= \frac{1}{a_{11}}\left[b_n - a_{n1}x_1^{(k+1)} - a_{n2}x_2^{(k+1)} - ...... - a_{n,n-1}x_{n-1}^{(k+1)}\right] \end{aligned}$$

(a) (b)

Fonte: Autores

Para funcionamento do algoritmo é necessário determinar a entrada dos dados – apresentados em forma de matriz (Figura 2), que são separadas em matriz dos coeficientes (A), matriz dos termos independentes (B) e matriz de valores iniciais (X).

Figura 2. Matrizes com valores de entrada no Mathcad

$$A := \begin{bmatrix} 3 & 2 & 2 \\ 1 & 4 & 2 \\ 2 & 1 & 6 \end{bmatrix} \qquad B := \begin{bmatrix} 480 \\ 400 \\ 400 \end{bmatrix} \qquad X := \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

Fonte: Autores

A apresentação do algoritmo como resolução do método Gauss-Seidel resulta em um aprendizado lógico e sequencial de passos que levam a compreensão do método. No código apresentado na Figura 3 (a), tem-se a representação da função "G(t,n)" que assume "t" (é usado como critério de parada um número máximo de iterações t) e "n" (define a ordem da matriz) como variáveis de entrada para o método. O uso do operador de repetição "for" indica que determinada expressão está dentro de um laço com parâmetros de "i" que vai de 1 até "n" e "i,j" a linha e coluna da matriz, respectivamente. Em seguida, no início desta repetição usa-se a matriz "S" como auxiliar para receber valores temporários da equação que funciona como um somatório dela mesma mais o respectivo elemento da matriz "A" (matriz dos coeficientes) multiplicada por "X" (vetor de valores aleatórios iniciais na iteração k). Fora da primeira repetição "X" assume o seu valor parcial da iteração (k+1), como mostrado na Equação 1, através da atribuição. A resposta é apresentada após atribuir valores para "t" e "n" na função "G", mostrado na Figura 3 (b).

Figura 3. (a)Desenvolvimento do método de Gauss-Seidel no *Software Mathcad Prime*; (b) Valores finais após o uso do método de Gauss-Seidel

```
G(t,n) := for k ∈ 1..t
            for i ∈ 1..n
              S ← 0
              for j ∈ 1..n
                if j≠i
                  S ← S + A_{i,j}·X_j
              X_i ← (1/A_{i,i})·(B_i − S)
          X
```

$$G(16,3) = \begin{bmatrix} 103.333 \\ 63.333 \\ 21.667 \end{bmatrix}$$

(a) (b)

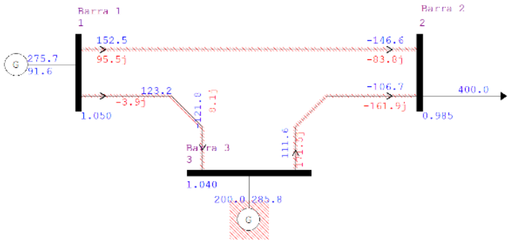
Fonte: Autores

4. FLUXO DE POTÊNCIA E O MÉTODO NUMÉRICO DE GAUSS-SEIDEL

Métodos de fluxo de potência em sistema de energia são utilizados para a determinação dos estados de operação de um sistema, como a potência a ser demandada por um gerador de energia, assim como as perdas nas linhas de transmissão. Além disso, tais métodos auxiliam como norteadores para obtenção das melhores condições de operação e controle. Nos sistemas de potência em condição de regime permanente, as análises numéricas são extensas e acabam sendo necessárias técnicas de simplificação para sistemas de grande porte. A notação por unidade (pu), as componentes simétricas e métodos de resolução, como Gauss-Seidel e Newton-Raphson, facilitam o entendimento e simplificam os cálculos. Na representação dos sistemas de potência os componentes podem estar ligados de duas formas: entre as barras do sistema, como os transformadores e linhas de transmissão ou nas extremidades, como geradores e cargas.

A partir dos parâmetros de impedância do sistema é formulada a matriz de admitância das barras (Y_{bus}), e dos elementos ligados nas barras são obtidos os valores necessários para o cálculo de resolução do sistema. As equações básicas de fluxo de potência são obtidas impondo-se a primeira lei de Kirchhoff, no tocante à conservação das potências ativa e reativa em cada barra da rede, isto é, a potência líquida injetada (S_i) em uma barra deve ser igual à soma das potências que fluem pelos componentes conectados a esta barra [4]. A segunda lei de Kirchhoff é utilizada para expressar os fluxos de potência nos ramos como função das suas tensões terminais. Quatro grandezas estão associadas a cada barra da rede: Módulo da tensão na barra i (V_i); Ângulo da tensão na barra i (θ_i); Potência ativa líquida injetada na barra i (P_i); Potência reativa líquida injetada na barra i (Q_i) [4]. A análise do problema, que envolve o fluxo de potência apresentada na Figura 4 e da referida solução, tem como objetivo especificar as tensões das barras, as iterações necessárias para convergência, os fluxos de potência ativas e reativas nas barras, a potência total dos geradores e as perdas totais dos sistemas.

Figura 4. Esquema do fluxo de potência desenvolvido no *Software* Anarede



Fonte: Autores

O motor da resolução desse tipo de sistema é o uso do método de Gauss-Seidel, porém, para poder aplicá-lo é preciso preparar as equações e grandezas envolvidas. O primeiro passo para a resolução do sistema é identificar cada barra com os seus respectivos atributos para facilitar a visualização do que é preciso calcular. Em seguida, deve-se assumir a ordem (n) da matriz Y_{bus}, composta com elementos da admitância. Sabendo-se que admitância, assume a variável Y, e é o inverso da impedância Z, podemos obter a matriz Y_{bus} utilizando as equações (2) e (3):

$$Y_{ii} = \sum_{\substack{j=0 \\ j\neq 1}}^{n} Y_{ij} \tag{2}$$

$$Y_{ij} = Y_{ji} = -Y_{ij} \tag{3}$$

A próxima etapa é determinar as potências líquidas, tanto das barras com potência gerada (P_g, Q_g) ou demandada (P_d, Q_d); considerando também o S_b como 100 MVA, o qual é apresentado no fluxo de potência. Através da equação (4).

$$Si = \frac{(P_{gi} - P_{di}) + j(Q_{gi} - Q_{di})}{Sb} \quad (4)$$

O método de Gauss-Seidel é um método utilizado para a resolução de sistemas de equações lineares, que obedece a um critério de convergência. Cada iteração consiste na repetição da equação inicial, porém com a atualização do valor das grandezas que já foram calculadas na iteração anterior, obtendo assim uma aproximação do resultado. Após certo número de iterações o sistema apresenta a resposta aproximada, com exatidão a pré-definida. Para encontrar as tensões de barra assumem-se estimativas iniciais padrões para as barras onde a tensão é desconhecida e aplica-se essas informações nas equações que deverão ser repetidas até a convergência. Para barras de carga, utiliza-se a equação (5).

$$V_i^{(k+1)} = \left[\frac{P_i^{esp} - jQ_i^{esp}}{V_i^{*(k)}} - \sum_{\substack{j=1 \\ j \neq i}}^{n} Y_{ij} V_j \right] * \frac{1}{Y_{ii}} \quad (5)$$

Para barras de geração é usada a mesma equação, porém, a carga reativa (Q) também precisa ser determinada pelo método de Gauss-Seidel, por meio da equação (6).

$$Q_i^{(k+1)} = -j\{V_i^{*(k)}.[Y_{ii}V_i^{(k)} + \sum_{\substack{j=1 \\ j \neq i}}^{n} Y_{ij} V_j^{(k)}]\} \quad (6)$$

Com as tensões de cada barra encontradas, pode ser determinada a injeção de potência nas barras de referência e geração, usando a equação (7).

$$P_i - jQ_i = V_i^*.[Y_{ii}V_i \quad + \sum_{\substack{j=1 \\ j \neq i}}^{n} Y_{ij} V_j] \quad (7)$$

Para obter as correntes de linha (I_{ij}), partimos da Lei de Ohm (V=RI), adaptada na equação (8), e usamos as grandezas da última iteração de Gauss-Seidel.

$$I_{ij} = Y_{ij} * (V_i - V_j) \quad (8)$$

Os fluxos de potência das linhas de transmissões (LTs) são obtidos através da equação (9)

$$S_{ij} = V_i * I_{ij}^* \quad (9)$$

E as perdas nas LTs são calculadas usando a Equação 8.

$$S_{ij} = S_{ij} + S_{ji} \quad (10)$$

Os resultados obtidos através da resolução do problema proposto permitem modelar um sistema com fluxo de potência. Este modelo de resolução será usado para mostrar como o s*oftware Mathcad Prime* traz uma facilidade para o ensino em diversas disciplinas.

5. METODOLOGIA

O presente estudo investiga o desenvolvimento do pensamento computacional, utilizando a metodologia de aprendizagem baseada na Resolução de Problemas por meio de atividades realizadas no componente curricular de Cálculo Numérico Computacional, dos cursos de Engenharias, com carga horária de 72h/a, e o do *software Mathcad Prime*, como recurso de apoio ao ensino e aprendizagem. O desenvolvimento das aulas de Cálculo Numérico tem como referência a metodologia de resolução de problemas proposta pelo matemático Polya [3], que apresenta quatro fases em seu processo: 1) Entender o problema; 2) Construir um plano para solucionar o problema; 3) Colocar o plano em funcionamento; 4) Avaliar a solução quanto à precisão e quanto ao seu potencial como ferramenta para solucionar outros problemas. Tem-se como objetivo colaborar o para que o acadêmico se transforme em um sujeito mais ativo, capaz de contribuir para o desenvolvimento de novas habilidades.

No contexto da resolução do problema envolvendo o fluxo de potência com o uso da programação do *Mathcad Prime* e dos métodos numéricos estudados, no intuito de desenvolver habilidades relacionadas a resolução de um problema que envolve Sistemas de Equações Lineares Algébricas (SELA), permitindo criar com as tecnologias digitais, desenvolveu-se as seguintes etapas: 1) Compreensão do problema através de conceitos relacionados ao tópico de SELA – métodos numéricos envolvidos; 2) Paralelamente foi feito o estudo de fluxos de potência no componente curricular de Sistemas de Energia; 3) Utilização da Planilha Excel, para a organização de dados e uso do método numérico - Método iterativo de Gauss-Seidel, a fim de relacionar elementos para posterior construção de algoritmos no *Mathcad Prime*. 4) Estudo do procedimento algoritmo para resolver o problema, tendo como base o estudo de métodos numéricos; 5) Formulação do algoritmo e representação na forma de um programa – a partir da apresentação da ferramenta, suas funcionalidades, potencialidades, características e posteriormente a aplicação na modelação e resolução de problemas que se apresentam diferentes tópicos envolvidos. 6) Validação de modelos e avaliação quanto a precisão e potencial para resolver problemas de maior ordem. 7) Análise e conclusão a respeito dos resultados.

A partir dessas etapas, um dos acadêmicos do curso de Engenharia Elétrica desenvolveu o problema do fluxo de potência, em oito passos, sob a forma de algoritmos elaborados no *software Mathcad Prime*. Uma das grandes vantagens deste *software* é a capacidade que a sua interface gráfica traz para o usuário, tendo o fácil acesso à entrada de dados, manipulação e algoritmos.

6. DESENVOLVIMENTO DA ATIVIDADE E RESULTADO OBTIDO

Passo 1: Identificação das barras, Linhas de Transmissão e variáveis. Na Figura 5 (a) temos a definição das linhas das impedâncias nas linhas de transmissão, apresentada na matriz *''Input''*, já na matriz "*Inj*", Figura 5 (b), as potências demandadas e geradas pelo sistema são organizadas por ordem das barras, respectivamente.

Passo 2: Encontrar a Y_{bus} (Figura 5 (c)) constitui uma tarefa repetitiva e que induz ao erro, conforme o aluno resolve atividades. Com o uso do *software* o aluno pode tanto entender como funciona a constituição da matriz Y_{bus}, quanto conferir o resultado desenvolvido manualmente.

Passo 3: Calcular a injeção líquida de potências em (pu) é um dos passos para definir as características do sistema, através da equação (4) – algoritmo apresentado na Figura 5 (d).

Passo 4: Este passo,utiliza o método de Gauss-Seidel a partir da equação (5) e (6), motor da resolução do problema. A Figura 5 (e) representa o algoritmo que calcula as tensões de barra.

Passo 5: Com o resultado das iterações do Passo 4, o restante dos cálculos definem as características específicas do sistema, como a injeção de potência na barra referência, na Figura 5 (f), usando a equação (7).

Passo 6: Para encontrar as correntes nas linhas de transmissão o algoritmo apresentado na Figura 5 (g), utiliza a equação (8). A apresentação em forma de matriz permite uma melhor visualização das correntes.

Passo 7: O fluxo de potência nas linhas de transmissão é apresentado em uma matriz na Figura 5 (h), obtido através da equação (9).

Passo 8: Perdas nas LTs (Figura 5 (i) e Figura 5 (j)), usando a equação (10).

Os resultados demonstram que o *software Mathcad Prime*, contribui na solução do problema. Pode ser observado que em todo o processo não foi negligenciada nenhuma etapa da resolução do problema, garantindo a sua total compreensão pelo acadêmico. Como as equações e códigos ficam explícitos na interface do *software*, torna-se possível analisar o caminho percorrido, diminuindo a possibilidade de erros. Logo, a estratégia do *software* integrar os métodos números na solução do problema em um número finito de passos, possibilitando a verificação de cada um deles. O desenvolvimento do método de Gauss-Seidel através do *software* permite que se tenha acesso dinâmico ao problema, podendo alterar e conferir os resultados apresentados. É importante considerar que o uso de *softwares* matemáticos cria condições para utilização de métodos numéricos em situações futuras mais complexas, e que a linguagem própria do *Mathcad Prime* torna possível que acadêmicos resolvam o mesmo problema de diferentes maneiras, pois vai depender da organização do pensamento lógico ao qual se dispõe, e também do seu nível de desenvolvimento cognitivo e conhecimento na área.

Figura 5 - (a) Impedância das linhas, cargas e demandas de potência; (b) Valores em matriz; (c) Matriz de admitância; (d) Injeção líquida de potências em PU; (e) Algoritmo para obtenção das tensões de barra; (f) Equação que permite encontrar a injeção de potência; (g) Algoritmo para obtenção das correntes de linhas (h) Algoritmo para obtenção dos fluxos de potências nas LTs; (i) Algoritmo para determinar as perdas nas LTs; (j) Resultado.

$$Input := \begin{bmatrix} 0 & (0.02+0.04\cdot 1i)^{-1} & (0.01+0.03\cdot 1i)^{-1} \\ (0.02+0.04\cdot 1i)^{-1} & 0 & (0.0125+0.025\cdot 1i)^{-1} \\ (0.01+0.03\cdot 1i)^{-1} & (0.0125+0.025\cdot 1i)^{-1} & 0 \end{bmatrix}$$

(a)

$$Inj := \begin{array}{c} \begin{array}{cccc} Pg & Pd & Qg & Qd \end{array} \\ \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 400\cdot 10^6 & 0 & 250\cdot 10^6 \\ 0 & 0 & 200\cdot 10^6 & 0 \end{bmatrix} \end{array} \qquad Sb := 100000000$$

(b)

$$\begin{array}{l} Ybus := \\ \quad X \leftarrow Input \\ \quad \text{for } i \in 1..n \\ \quad\quad \text{for } j \in 1..n \\ \quad\quad\quad \text{if } i \neq j \\ \quad\quad\quad\quad X_{i,i} \leftarrow X_{i,i} + Input_{i,j} \\ \quad\quad\quad\quad X_{i,j} \leftarrow X_{i,j} \cdot -1 \\ \quad X \end{array}$$

$$Ybus = \begin{bmatrix} 20-50i & -10+20i & -10+30i \\ -10+20i & 26-52i & -16+32i \\ -10+30i & -16+32i & 26-62i \end{bmatrix}$$

(c)

$$\begin{array}{l} S := \\ \quad \text{for } i \in 1..n \\ \quad\quad X_i \leftarrow \dfrac{\left(Inj_{i,1} - Inj_{i,2}\right) + \left(\left(Inj_{i,3}\right) - \left(Inj_{i,4}\right)\right)\cdot 1i}{Sb} \\ \quad X \end{array}$$

$$S = \begin{bmatrix} 0 \\ -4-2.5i \\ 2i \end{bmatrix}$$

(d)

```
Gauss := X ← V
         fim ← 0
         erro ← 0.00001
         while fim = 0
             X ← V
             for i ∈ 2..n
                 M ← 0
                 for j ∈ 1..n
                     if j ≠ i
                         M ← M + Ybus[i,j] · V[j]
                 K ← V
                 V[i] ← 1/Ybus[i,i] · (conj(S[i])/conj(V[i]) − M)
                 if ((Inj[i,3]) ∨ (Inj[i,1])) ≠ 0
                     Q ← −1i · (conj(V[i]) · (Ybus[i,i] · V[i] + M))
                     V[i] ← abs(K[i]) · cos(arg(K[i])) + 1i · abs(K[i]) · sin(arg(V[i]))
             fim ← 1
             for i ∈ 1..n
                 if |V[i] − X[i]| ≥ erro
                     fim ← 0
                     break
         V
```

(e)

$$Y := Ybus \cdot -1$$

$$P := \overline{Gauss_{1,1}} \cdot \left(\left(Ybus_{1,1} \cdot Gauss_1 \right) + \sum_{i=2}^{n} \left(Ybus_{1,i} \cdot Gauss_i \right) \right) = 4.11 - 0.7i$$

(f)

```
I := for i ∈ 1..n
         for j ∈ 1..n
             I[i,j] ← Y[i,j] · (Gauss[i] − Gauss[j])
     I
```

$$I = \begin{bmatrix} 0 & 2.221-0.898i & 1.692+0.231i \\ -2.221+0.898i & 0 & -1.695+1.966i \\ -1.692-0.231i & 1.695-1.966i & 0 \end{bmatrix}$$

(g)

```
S := for i ∈ 1..n
         for j ∈ 1..n
             S[i,j] ← Gauss[i] · I[i,j]
     S
```

$$S = \begin{bmatrix} 0 & 2.332-0.943i & 1.777+0.242i \\ -2.09+1.028i & 0 & -1.504+2.027i \\ -1.772-0.15i & 1.658-2.135i & 0 \end{bmatrix}$$

(h)

```
Sp := for i ∈ 1..n
          for j ∈ 1..n
              Sp[i,j] ← S[i,j] + S[j,i]
      Sp
```

(i)

$$Sp = \begin{bmatrix} 0 & 0.242+0.085i & 0.005+0.092i \\ 0.242+0.085i & 0 & 0.154-0.108i \\ 0.005+0.092i & 0.154-0.108i & 0 \end{bmatrix}$$

(j)

Fonte: Autores

CONSIDERAÇÕES FINAIS

A estruturação do pensamento lógico permitiu integrar o computador, a matemática e a área de sistemas de energia. Avaliou-se que o software *Mathcad Prime* foi uma estratégia apropriada para reduzir a dimensão do problema de fluxo de potência, através da sua divisão em passos, combinação de dados e modelação para obter sua solução. A organização e a análise de dados permitiram automatizar a solução pelo pensamento algorítmico, generalizando esse processo para resolução de outros problemas, auferindo assim aspectos relacionados ao conceito de pensamento computacional.

A interatividade do software utilizado torna possível o desenvolvimento de habilidades citadas por ISTE e CSTA, tais como: a confiança em lidar com a complexidade; persistência em trabalhar com problemas difíceis; tolerância para ambiguidades; capacidade de lidar com problemas abertos; capacidade de comunicação; e trabalho com outros para atingir objetivos.

Pode-se afirmar que em todo o processo de resolução do problema houve a interação entre o objeto de estudo e o acadêmico, que construiu seu próprio conhecimento. Ao fim desse ciclo, é possível constatar uma mudança de postura por parte dos envolvidos, potencializada pela interdisciplinaridade e pelo uso do *software*, destacando a integração diferentes áreas do conhecimento e o estímulo a resolução, onde o erro pode ser visto como uma oportunidade para a aprendizagem. Por fim, constatou-se que a elaboração de soluções usando os elementos do pensamento computacional promove desafios interdisciplinares, levando o acadêmico a entrar em contato com esse conteúdo de forma dinâmica, observando que a máquina estende a capacidade humana de resolver problemas, e assim potencializando o trabalho e favorecendo o ensino aprendizagem com autonomia intelectual e pensamento crítico.

REFERÊNCIAS

[1] CHERRI, Adriana; VIANNA, Andréa; BALBO, Antonio; BAPTISTA, Edméa. Métodos Numéricos Computacionais. **Citação de referências e documentos eletrônicos**. Disponível em: http://wwwp.fc.unesp.br/~adriana/Numerico/Introducao.pdf. Acesso em: 28 de setembro de 2017.

[2] Operational Definition of Computational Thinking. ISTE. **Citação de referências e documentos eletrônicos.** Disponível em: http://www.iste.org/docs/ct-documents/computational-thinking-operational-definition-flyer.pdf?sfvrsn=2. Acesso em: 27 set. 2017.

[3] POLYA, G. **A arte de resolver problemas: um novo aspecto do método matemático**. 2.ed. Rio de Janeiro: Interciência, 1995.

[4] STEVENSON, W. **Elementos de análise de sistemas de potência**. 2° Edição São Paulo. McGraw-Hill, 1986.

[5] SICA, Carlos. (2011). Ciência da Computação no Ensino Básico e Médio. **Citação de referências e documentos eletrônicos.** Disponível em: http://www.odiario.com/blogs/carlossica/2011/10/07/ciencia-da-computacao-no-ensino-medio. Acesso em: 28 de set. de 2017.

[6] VALENTE, José Armando. **O computador na sociedade do conhecimento**. Campinas: UNICAMP/NIED, 1999.

[7] WING, Jeannette M. **Computational Thinking**. Communications of the ACM, v.49, n. 3, p. 33-35, 2006.

PROBLEM SOLVING SKILLS WITH MATHCAD PRIME PRIME UNDER STIMULUS OF COMPUTATIONAL THINKING

Abstract: *The computational capacity offered by mathematical software allows the simulation of problems in computational form, a process that involves mathematical modeling, implementation of equations and analysis of produced results. This applicability and automation of calculations refer to important elements of computational thinking. For an experienced experience in the development of a study topic of the Computational Numerical Computation curricular component of Electrical Engineering, through the resolution of a power flow problem, we intend to emphasize the potential of the Mathcad Prime software for the development of thought computational. The planning of classes is based on Problem Solving, whose experience is transformed into learning through a learning cycle involving the following stages: understanding the problem, followed by the elaboration of a plan and execution, and finally, the verification of results. Among the activities proposed in this curricular component, it is highlighted that problem solving happens in a transversal way, establishing the relationship between the theoretical study and practical application, where the mathematics involved was favored by the area of computation.*

Keywords: *Numeric Methods. Computational Thinking. Problem Solving*