

INTENSIFICAÇÃO DO USO DE FERRAMENTAS COMPUTACIONAIS NOS CURSOS DE GRADUAÇÃO EM ENGENHARIA QUÍMICA – A EXPERIÊNCIA DA UFES, CAMPUS ALEGRE

G. T. PELISSON¹, F. T. VIEIRA¹

¹ LAMCES – Laboratório de Métodos Computacionais, Controle e Estimação
Universidade Federal do Espírito Santo – Engenharia Química (campus Alegre)
e-mail para contato: gabriela_tosip@hotmail.com

RESUMO – Na Engenharia Química (EQ), os programas computacionais têm sido cada vez mais utilizados. Porém, muitas vezes os estudantes não têm muito interesse e/ou não são devidamente estimulados a aplicar linguagens computacionais na solução de problemas de engenharia, algo tão fundamental na atualidade. Assim, percebeu-se a necessidade da criação de um material didático para facilitar e incentivar a implementação de meios computacionais, desde o ciclo básico, em problemas vistos em sala de aula, criando uma familiaridade e continuidade do uso de softwares como o *Matlab*, amplamente utilizado no ensino e pesquisa no meio acadêmico, para efetuar desde contas simples até na aplicação de métodos numéricos na solução de problemas na área de EQ.

1 - INTRODUÇÃO

O modelo tradicional de ensino adotado na educação em Engenharia está apoiado na transmissão de conhecimentos, que normalmente focaliza os aspectos conceituais das diversas teorias, sem a sua necessária contextualização. A reprodução desses conhecimentos é valorizada por meio do estímulo à memorização, pela prática repetitiva dos mecanismos e da lógica de funcionamento dos modelos conceituais e pela aplicação de técnicas e métodos como forma única de solução de problemas (Bazzo, 1998).

É de conhecimento comum entre os profissionais do ensino de engenharia que este modelo precisa evoluir, pois não cabe mais em uma sociedade onde o conhecimento, principalmente em termos de conteúdo, pode ser adquirido por diferentes meios, além daquele transmitido pelo professor (Silveira, 2005). Atualmente, o desenvolvimento das ferramentas computacionais possibilita a expansão de uma fronteira até então pouco explorada, a da incorporação da simulação numérica como método de ensino (Belhot et al., 2001).

A modelagem matemática, em conjunto a simulação de processos são poderosas ferramentas para o estudo e entendimento do comportamento de processos complexos comuns na EQ. Estas ferramentas já encontram grande aplicação no meio industrial e no campo da pesquisa, porém ainda não são muito utilizadas no meio acadêmico como um método sistematizado para o ensino.

Nesse sentido, inicialmente realizou-se uma enquete com o corpo discente acerca das principais demandas e implicações do uso de ferramentas computacionais para o processo de ensino/aprendizagem. O desafio desse projeto é incentivar o uso e revelar a importância da aplicação de ferramentas computacionais na solução de problemas de EQ, compreender as dificuldades de implantação de ferramentas computacionais e incentivar todo o corpo docente para o uso como recurso didático nas aulas do curso de graduação em EQ da UFES, campus Alegre.

2 - APOSTILA DE MATLAB

O projeto consiste em elaborar um material didático composto de tópicos envolvendo operações básicas em matlab, a aplicação de ferramentas na solução de problemas de EQ e também exemplos de implementação de métodos numéricos, com códigos comentados linha a linha. Este é o grande diferencial desse material didático: facilitar essa tarefa, que muitas vezes é a grande barreira a ser vencida pelos alunos: como passar o método do papel para a linguagem computacional.

Os principais tópicos explorados no projeto até o momento, serão apresentados a seguir.

2.1 - Operações com Vetores e Matrizes

Vetores Simples: Baseando-se em Costa (2003), a criação de um vetor é simples, basta abrir um colchete e introduzir os valores separados por espaço ou vírgula e depois fechar o colchete:

```
>> x = [0 0.25*pi 0.5*pi 0.75*pi pi]
x =
    0    0.7854    1.5708    2.3562    3.1416
```

Construção de Vetores: Para a construção de vetores com muitos termos, existem comandos que facilitam essa escrita. São eles:

- Criar um vetor que começa em zero e vai até o valor π , aumentando $0,25*\pi$ por vez:

```
>> x = (0:0.25:1)*pi
x =
    0    0.7854    1.5708    2.3562    3.1416
```

- Criar um vetor que começa em zero e vai até o valor π com 5 elementos com o uso da função *linspace*:

```
linspace(primeiro_valor, ultimo_valor, numero_de_valores).
>> x = linspace(0,pi,5)
x =
    0    0.7854    1.5708    2.3562    3.1416
```

Construção de Matrizes: Para a construção de matrizes no MATLAB utilizam-se espaço para separar elementos da mesma linha e ponto e vírgula para separar os elementos de uma linha da outra:

```
>> g = [1 2 3 4; 5 6 7 8]
g =
     1     2     3     4
     5     6     7     8
```

Formando, no exemplo, uma matriz 2 por 4. Se todas as linhas inseridas não tiverem o mesmo número de colunas, o MATLAB exibirá uma mensagem de erro.

Dimensão de Vetores e Matrizes: Algumas vezes é necessário saber o tamanho de uma matriz ou vetor, para isto existe as funções *size* e *length* no MATLAB:

```
>> [r,c] = size(g)      % do exemplo anterior
r =
     2
c =
     4
```

Onde, 'r' é o número de linhas e 'c' o de colunas.

length retorna o número total de elementos ao longo da maior dimensão da matriz e para vetores, o tamanho do mesmo:

```
>> a = [1 3 7 2 9 4];
>> length(a)
ans =
     6
```

2.2 - Controle de Fluxo

Como mostrado por Hanselman e Littlefield (2005) temos:

Loops FOR: O loop *for* no MATLAB tem a mesma função estudada na programação, eles repetem os comandos descritos um determinado número de vezes.

Loops WHILE: Ao contrário de comando *for*, o comando *while* executa os passos infinitas vezes até a especificação inicial ser alcançada. Para vetores e matrizes, o loop só termina quando todos os elementos são verdadeiros aos comandos.

Estrutura IF-ELSE-END: Quando é necessário aplicar um comando condicional, usa-se *if* e *else* onde se a o comando for verdadeiro ao programa a estrutura *if* é executado, caso contrário, o *else* será executado. O uso do *else* é opcional.

Estrutura SWITCH-CASE: Quando a variável deve se adequar a apenas um de vários comandos, ou seja, quando houver uma variável condicionada em relação a um mesmo argumento, aplica-se a estrutura *switch-case*.

2.3 - Arquivos M de Funções

Segundo Vicente (2003), é um programa elaborado como arquivo M e que calcula algo desejado pelo programador, devolvendo somente o valor da resposta. Os comandos executados por esse programa ficam ocultos. Você só visualiza o que entra e o que sai, ou seja, uma função é uma caixa preta.

Como escrever uma Função:

- Linha 1→ indica que o arquivo M contém uma função e especifica seu nome (que é o nome do arquivo sem a extensão.m). Essa linha também define as variáveis de entrada e saída.

- Próximas linhas → são comentários (textos) explicando o que essa função calcula.

Exemplo, calcular o fatorial de um número usando função:

```
function fat = fatorial(x)      % definição da função segundo a lógica exigida pelo
matlab
fat = 1; % inicializa o valor do fatorial
for n = 1:x % inicialização do loop
fat = fat*n; % permite a multiplicação que a operação fatorial exige
end % fim do loop
```

Na workspace do matlab o procedimento para o cálculo do fatorial seria o seguinte:

```
g = input('digite um numero: '); % informa o valor que se deseja o cálculo do fatorial
f = fatorial(g); % lógica para o cálculo do fatorial de g
disp(['Este eh o resultado: ' num2str(f)]) % estrutura para criar um texto informando o
valor do fatorial
```

2.4 - Ajuste Polinomial

Comando polyfit: Este comando acha os coeficientes do polinômio que estamos procurando. Mas, para isto devemos especificar o grau do polinômio. Possui três argumentos: primeiro as coordenadas x e y, e depois o grau do polinômio.

```
>> x = [0,1,2,3,4,5]; %vetor coluna de valores de x
>> y = [0,20,60,68,77,110]; %vetor coluna de valores de y
>> coef = polyfit(x,y,1); %o valor 1 significa que o polinômio será de grau 1
>> a = coef(1) %mostra o valor do coeficiente angular, já que será uma reta
>> b = coef(2) %mostra o valor do coeficiente linear

a =
    20.8286
b =
     3.7619
```

Dessa forma, teremos um polinômio da seguinte forma: $y = 20,8286 \cdot x + 3,7619$.

2.5 - Sistema Linear

Considerando o sistema de equações: $\{3x + 11y - 2z = 7; x + y - 2z = 4; x - y + z = 19\}$.

Para resolver, escreva o sistema de forma matricial $A \cdot x = b$, onde A é a matriz dos coeficientes, x é a matriz coluna das incógnitas e b é a matriz coluna dos termos independentes:

```
>> A = [ 3 11 -2; 1 1 -2; 1 -1 1];
>> b = [ 7; 4; 19];      %também pode ser escrito como b = [7 4 19]'
>> x = A\b

>> x =
    13.2188
    -2.3438
     3.4375
```

2.6 - Um Método para Mínimos Quadrados

É o processo que determina a função linear mais aproximada do comportamento dos pontos, calculada através do somatório dos mínimos quadrados das distâncias entre a reta e os pontos.

Um método simples que pode ser aplicado segundo Beers (2007), é a partir dos n dados das diferentes p incógnitas e os n resultados obtidos em cada 'experimento', fazer o seguinte procedimento a partir da forma genérica: $y^k = \beta_0 + \beta_1 \cdot x_1^k + \beta_2 \cdot x_2^k + \dots + \beta_p \cdot x_p^k$.

Com $k = 1, 2, 3, \dots, n$. Dessa forma, teremos n equações, cada uma com os valores de um 'experimento'. Para a solução do problema, devemos descobrir os valores dos β , para chegarmos à equação linear que satisfaz os pontos dados. Teremos:

$$Y = \begin{bmatrix} y^1 \\ y^2 \\ \vdots \\ y^n \end{bmatrix} \quad X = \begin{bmatrix} 1 & x_1^1 & x_2^1 & \dots & x_p^1 \\ 1 & x_1^2 & x_2^2 & \dots & x_p^2 \\ 1 & \vdots & \vdots & \dots & \vdots \\ 1 & x_1^n & x_2^n & \dots & x_p^n \end{bmatrix} \quad \theta = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \vdots \\ \beta_p \end{bmatrix}$$

Como $Y = X \cdot \theta$, para sabermos os valores de β , definida pela matriz coluna θ , basta multiplicar a transposta de X por Y, como no programa abaixo:

```
>> X = [1 -1 -1; 1 -0.699 -1; 1 -1 -0.699; 1 -0.699 -0.699];
>> Y = [-4.6096 -4.3157 -4.2999 -3.9988]';
>> teta = X\Y

teta =
    -2.5822
     0.9884
```

1.0409

Contudo, a equação obtida com estes resultados é $y = - 2,5822 + 0,9884.x_1 + 1,0409.x_2$.

3 - CONSIDERAÇÕES FINAIS

Nesse trabalho é abordado apenas a parte inicial desse material. Seu desenvolvimento visa ao incentivo do uso de programas computacionais nos cursos de EQ, já que é notável a falta de traquejo da maioria dos alunos com esse tipo de ferramenta, tal essencial para o profissional na atualidade. Espera-se que, ao término da apostila, esta seja um suporte eficiente e de fácil acesso e manejo para que os alunos de EQ e outros cursos se familiarizem, desenvolvam suas habilidades e pratiquem o que foi aprendido em sala no *Matlab* e outros programas.

4 - REFERÊNCIAS

BAZZO, W. A. *Ciência, Tecnologia e Sociedade e o Contexto da Educação Tecnológica*. ed. Florianópolis: Editora da UFSC, p. 287, 1998.

BEERS, K. J. *Numerical Methods for Chemical Engineering*. Cambridge: Cambridge University Press, p. 373-380, 2007.

BELHOT, R. V.; FIGUEIREDO, R. S.; MALAVÉ, C. O. *O Uso da Simulação no Ensino de Engenharia*. Anais COBENGE 2001– CD-ROM - XXIX Congresso Brasileiro de Ensino de Engenharia. Porto Alegre: UFRGS, 2001.

COSTA, A. O. S. *Matlab, Discas iniciais de utilização*. Rio de Janeiro: COPPE/UFRJ, p. 6-9, 2003.

DIEGUEZ, J. P. P. *Métodos e Cálculo Numérico*. Rio de Janeiro: IME, p. 431, 2005.

HANSELMAN, D; LITTLEFIELD, B. *Mastering Matlab 7*. New Delhi: Pearson Education, p. 200-210, 2005.

SILVEIRA, M. A. *A Formação do Engenheiro Inovador: Uma Visão Internacional*. Rio de Janeiro: PUC-Rio, Sistema Maxwell, p. 135, 2005.

VICENTE, S. A. S. *Curso introdutório de Matlab 6.5*. São Paulo: Universidade Presbiteriana Mackenzie, p. 42-43, 2003.