



USO EFICIENTE DE SOFTWARE NO ENSINO DE GRADUAÇÃO EM ENGENHARIA QUÍMICA

Thiago Gonçalves das Neves¹ e Michel François Fossy²

RESUMO

O aperfeiçoamento dos computadores fez surgir uma gama de softwares com grande potencial para auxiliar na tarefa de ensinar engenharia, dentre os quais podemos citar: Mathcad[®], Mathematica[®], Matlab[®], Maple[™] e Polymath. Diversos artigos têm sido publicados com o objetivo de demonstrar as vantagens do uso destes softwares, bem como no sentido de comparar os seus desempenhos na tarefa de ensinar. Os livros mais recentes também já indicam claramente a importância desta ferramenta para a engenharia. Neste artigo, os autores são enfáticos: independente do software escolhido, o pré-requisito fundamental para uso do computador na resolução de problemas da engenharia é o estudante pensar de forma lógica e sistemática. Minimizar o problema através do uso de facilidades e/ou simplificações pode custar um preço muito alto para o futuro da engenharia. A sistemática usada para resolução de problemas da engenharia defendida neste trabalho apresenta elevada taxa de sucesso em nível de pós-graduação e a idéia é antecipar o seu uso para os estudantes de graduação.

Palavras-chave: Algoritmo; Programação; Lógica; Matlab.

EFFICIENT USE OF SOFTWARE IN UNDERGRADUATE TEACHING IN CHEMICAL ENGINEERING

ABSTRACT

The development of computers brought an amount of software with a great potential to assist in the task of teaching engineering, which can be mentioned: Mathcad[®], Mathematica[®], Matlab[®], Maple[™] e Polymath. Several articles have been published with the aim of demonstrating the advantages of using these softwares, and to compare their performance. The most recent books have clearly indicated the importance of this tool in engineering. In this article, the authors declare: independent of the software chosen, the basis for the use of computers to solving engineering problems is the logical and systematic thinking. The use of facilities and/or simplifications in order to minimize the problems can cost a heavy price for the future of engineering. The methodology used to solve engineering problems defended in this work has a high success rate in graduate level and the idea is to anticipate its use for undergraduate students.

Keywords: Algorithm; Programming; Logic; Matlab.

INTRODUÇÃO

Neste século XXI pode-se perceber a necessidade do engenheiro em incorporar o computador ao seu cotidiano, vendo que esta máquina é capaz de sistematicamente coletar, manipular e fornecer os resultados da manipulação de informações para um ou mais objetivos.

A automatização de procedimentos, o controle administrativo sobre pessoas e materiais diversos e, ainda, a possibilidade da execução de cálculos rotineiros de forma mais precisa, foram os grandes

¹ Aluno de Curso de Engenharia Química, Depto. de Engenharia Química, UFPG, Campina Grande, PB, E-mail: thiago.gdn@hotmail.com

² Engenheiro Químico, Prof. Doutor, Depto. de Engenharia Química, UFPG, Campina Grande, PB, E-mail: michel@deq.ufcg.edu.br

motivadores do aprendizado em programação de computadores. Na programação é essencial ter-se um raciocínio com uma seqüência de instruções capaz de conduzir a uma determinada conclusão. A ordem das instruções deve obedecer a uma particular lógica chamada de *algoritmo*. As etapas de um algoritmo são as instruções que deverão ser executadas por uma máquina. Esse conjunto de instruções é o que podemos chamar de programação, ou seja, a formalização de um algoritmo em linguagem inteligível pelo computador.

Todo dado coletado pelos computadores, as instruções por eles executadas, bem como os resultados de um processamento, são sempre constituídos de conjuntos ordenados de zeros e uns, linguagem que é chamada de *linguagem de máquina* que é tediosa de manipular, difícil de compreender e fácil de acarretar erros. Por essa razão, foram desenvolvidas outras linguagens, mais próximas do entendimento dos operadores, genericamente chamadas de *linguagem de programação*. Cada uma dessas linguagens possui regras fixas e rígidas de sintaxe (FORBELLONE, 1993).

Porém, mesmo com as diversas linguagens de programação, ainda existe uma barreira entre o computador e o engenheiro, pois ao aprender uma linguagem de programação, o engenheiro fica muito preso com a sintaxe da mesma. E sabemos que isto não basta, pois os problemas que nos deparamos são complexos e exigem senso criativo, desenvolvida capacidade para formular e equacionar matematicamente problemas de engenharia e, ainda, competência na programação destes métodos em modelos computacionais de solução (MARTINS FILHO & SANGLARD, 2000).

Segundo Lauermaann et al. (2003), as tecnologias computacionais aparecem como meios de possibilitar uma ação docente inovadora. Entretanto, essa inovação não deve estar somente restrita ao uso do computador, mas também à maneira como o professor vai aproveitar as formas de elaborar planos metodológicos que superam a simples reprodução do conhecimento. Por isso, deve-se ressaltar que recursos existem e estão disponíveis, basta-se dispor de projetos educativos adequados para desencadear o processo de aprendizagem.

Ainda segundo Longo (2006), o computador dá uma falsa sensação de segurança que nem sempre garante resultados satisfatórios. Na verdade, o homem, pela sua própria natureza, pode cometer enganos e o computador pode multiplicar estes erros e causar grandes desastres. Dentre as possíveis causas dos problemas computacionais, podem-se citar as seguintes:

- Escolha inadequada de um programa;
- Erros de programação;
- Usuário com pouco conhecimento do assunto;
- Interpretação errada do programa;
- Erros de digitação na entrada de dados;
- Modelagem do problema mal feita;
- Erros na interpretação de resultados.

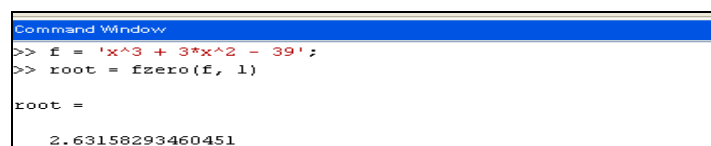
Por estes e outros motivos é de fundamental importância a análise dos resultados obtidos no computador. Alguns tipos de análises foram feitas nas aplicações a seguir, tais como:

- Visualização gráfica;
- Verificação das diversas soluções, verificando se estas estão condizentes com a literatura;
- Utilização da noção da natureza da grandeza da solução.

DESCRIÇÃO DO PROBLEMA

Tradicionalmente, diversos cursos de programação de computadores para engenharia adotam a linguagem Pascal como referência de programação estruturada e buscam desenvolver no aluno o uso correto da sintaxe utilizada em situações clássicas de programação. Todavia, com esta forma de ensino a capacidade do estudante fica limitada e só consegue resolver problemas triviais, não utilizando a programação de forma eficaz durante o seu curso, ou seja, não consolida nem reconhece o papel do computador na modelação matemática de soluções de engenharia e na idealização de respostas ao problema original (MARTINS FILHO & SANGLARD, 2000).

Além disso, geralmente o estudante pensa que é tempo despendido implementar um método numérico para em seguida utilizar este código na resolução de problemas. Com o objetivo de amenizar este problema, foram desenvolvidos os softwares comerciais, tais como Mathcad[®], Mathematica[®], Matlab[®], Maple[™] e Polymath. Porém o exemplo apresentado na *Figura 1* mostra que isso ainda não é o suficiente.



```
Command Window
>> f = 'x^3 + 3*x^2 - 39';
>> root = fzero(f, 1)

root =

    2.63158293460451
```

Figura 1. Uso do Matlab (Janela de comando) para encontrar uma raiz de um polinômio de 3º grau.

A *Figura 1* mostra a eficácia do software Matlab na determinação de uma das raízes de um polinômio de 3º grau, porém os problemas de engenharia não aparecem de forma tão simples. Estes aparecem de forma contextualizada, exigindo que o estudante defina uma seqüência de etapas até chegar ao equacionamento final. Este é o verdadeiro sentido de programar.

Este trabalho tem como objetivo principal desenvolver uma metodologia de uso de softwares para resolução de problemas da graduação em Engenharia Química. É uma metodologia que em nível de pós-graduação apresenta elevada taxa de sucesso e a idéia é antecipar o seu uso para os estudantes de graduação.

Os objetivos específicos são:

- Mostrar a importância da ferramenta computacional no cotidiano do engenheiro;
- A resolução sistemática de problemas envolvendo conteúdos básicos, profissionais e específicos da Engenharia Química.

Para facilitar o desenvolvimento de algoritmos e a implementação de códigos na resolução de problemas de engenharia, foi utilizada neste projeto uma metodologia que visa dividir o problema de forma modular, tal divisão ocorre da seguinte forma:

- Programa principal;
- Modelo do problema;
- Parâmetros;
- Solver.

Esta metodologia é baseada no que diz respeito à construção de programas complexos. De acordo com Passos (2007) poucos são os programas complexos que estão “livres de erros”. E quando um programa deste porte atinge certo tamanho a mente humana não consegue controlar toda a complexidade. Atualmente para resolver este problema, uma das opções é baseada na construção de pequenos programas “livres de erros” e reuni-los, construindo um grande programa “livre de erro”.

O uso dessa metodologia é focado no tópico: Resolução de problemas da engenharia química, sendo o Matlab escolhido por dispor de uma biblioteca bastante abrangente de funções matemáticas, geração de gráficos e manipulação de dados que auxiliam muito o trabalho do programador. E ainda possui uma vasta coleção de bibliotecas, denominadas toolboxes. A linguagem e o ambiente de programação Matlab permitem ainda que o usuário possa escrever suas próprias bibliotecas. Assim, o usuário pode enriquecer a linguagem, incorporando a ela novas funções.

RESOLUÇÕES DE PROBLEMAS DA ENGENHARIA QUÍMICA

Equações de Estado

Em sistemas termodinâmicos nos deparamos com relações matemáticas envolvendo variáveis de estado, ou seja, variáveis referentes ao estado momentâneo do sistema, tais como: Pressão, temperatura e volume. Relações deste tipo são chamadas EQUAÇÕES DE ESTADO.

Para descrever-se uma equação de estado, a princípio surgiu a equação dos gases ideais como mostra a *Equação (1)*, esta é bastante satisfatória apenas quando o fator de compressibilidade (Z) do gás tende a um. Ou seja, quando a pressão é baixa e a temperatura é alta, isto faz com que a distância entre as moléculas seja a maior possível. Nessas condições, os choques entre as moléculas se tornam praticamente elásticos, havendo pouca perda de energia cinética. Porém quando isto não ocorre, equações de estado devem apresentar maior complexidade, exigindo o uso de métodos numéricos para tal resolução.

$$PV = nRT \quad (1)$$

Existem várias equações para gases reais, entre elas podemos destacar a Equação de Redlich-Kwong, pois durante muitos anos foi a equação mais usada em cálculos reais de engenharia. Esta representou um enorme avanço sobre a Equação de Van der Waals, justamente por apresentar o termo \sqrt{T} no denominador. A Equação de Redlich-Kwong é representada a seguir:

$$P = \frac{RT}{(V-b)} - \frac{a}{V(V+b)\sqrt{T}} \quad (2)$$

As constantes a e b são expressão da seguinte forma:

$$a = 0,42747 \left(\frac{R^2 T_c^{2.5}}{P_c} \right) \quad (3)$$

$$b = 0,08664 \left(\frac{R T_c}{P_c} \right) \quad (4)$$

A Figura 2 mostra um exemplo de como desenvolver um código no Matlab para resolver problemas desta natureza, utilizando a metodologia proposta e alguns dados fornecidos por Van Ness et al. (2007).

```

1      % ===== Main =====
2
3      clear all; clc    % Limpa a tela e todas as variaveis existente
4
5      global T p
6
7
8      %-----entrada de dados-----
9
10     T= input('entre com o valor da temperatura absoluta: ');
11
12     Est_inicial= input('entre com uma estimativa inicial: ');
13
14
15     %-----calculos-----
16
17     for i=1:31
18         pres(i)=i;
19         p=i;
20         vol(i)=fzero('model', Est_inicial);
21     end
22
23     parameters
24
25
26     Z=pres.*vol/(R*T); % fator de compressibilidade
27
28
29     %-----graficos-----
30     plot(pres,Z, 'm*-')
31     ylabel('Fator de compressibilidade (Z)'), xlabel('Pressao (p)'), grid
32

```

(a)

```

1      % ===== parameters =====
2
3      % para o n-butano:
4
5      Tc=425.1; % temperatura critica em K
6
7      pc=37.96; % pressao critica em bar
8
9      R=81.144; %constante do gas em bar.cm³/mol.K
10
11     a=0.42748*R^2*Tc^2.5/pc;    % constante a
12
13     b=0.08664*(R*Tc/pc);      % constante b

```

(b)

```

1      % ===== model =====
2
3      function y = model(v)
4
5      global T p
6
7      parameters
8
9
10     |
11     y = R*T*v*(v+b) *T^(1/2) -a*(v-b) -p*v*(v+b) *T^(1/2) *(v-b);

```

(c)

Figura 2. M-files utilizados de forma modular, envolvendo equações de estado: (a) programa principal; (b) Parâmetros do n-butano e (c) modelo matemático.

O código segundo mostra a figura anterior nos permite plotar o gráfico do fator de compressibilidade versus a pressão para o n-butano a uma determinada temperatura, usando a Equação de Redlich-Kwong com a pressão variando de 1 até 31 atm.

No arquivo m-file chamado *parameters.m*, de acordo com a *Figura 2b*, estão todos os parâmetros associados à espécie química em questão (n-butano). Como estes dados estão em um local específico, apenas este arquivo será modificado caso ocorra à necessidade de analisar outra espécie química.

No programa principal (*Main.m*), ilustrado na *Figura 2a*, são utilizados antes de tudo, os comandos *clear all* e *clc*, o primeiro é utilizado para limpar todas as variáveis existentes e o segundo é utilizado para limpar a tela de comandos. Em seguida com o comando *global*, tornou-se possível um jogo de variáveis para ser globalmente acessível. Neste caso, acessam-se as variáveis *T* (temperatura) e *p* (pressão) sem a necessidade de declará-las na lista de entrada das funções quando tal comando é usado (UERJ, 1999). Ainda nesse m-file, é bastante freqüente a utilização do comando *input que nos permite* exibir uma mensagem na tela dando ao usuário um prompt para entrada de dados. Neste exemplo tal comando foi usado para armazenar uma estimava inicial fornecida para resolver a função modelo.

A expressão *i = 1:31* (vide *Figura 2a*) gera uma matriz linha contendo números de 1 a 31 com incremento unitário. Portanto, os comandos entre as instruções *for* e *end* são executados uma vez para cada coluna da expressão matricial, ou seja, trinta e uma vezes. Dentre estes comandos, está a função *fzero*, a qual é utilizada para localizar a raiz da Equação de Redlich-Kwong, definida em outro arquivo m-file (*model.m*) conforme mostra a *Figura 2c*. Para isto foi necessário utilizar o comando *function*, criando-se assim uma nova função chamada *model*. Esta nova função tem uma variável *V* (argumento de entrada) e o valor desta função para um determinado valor de *V* é armazenado na variável *y* (argumento de saída). Mas antes de definir a função é preciso deixar disponíveis, no mesmo m-file, os parâmetros (digitando o nome com o qual o arquivo que os contém foi salvo). Além dos parâmetros, é necessário ter-se disponível as variáveis *T* e *p* para o cálculo do volume específico do gás. Portanto, chamam-se também as variáveis globais (utilizando o comando *global*). A comunicação entre os arquivos é ilustrada abaixo:

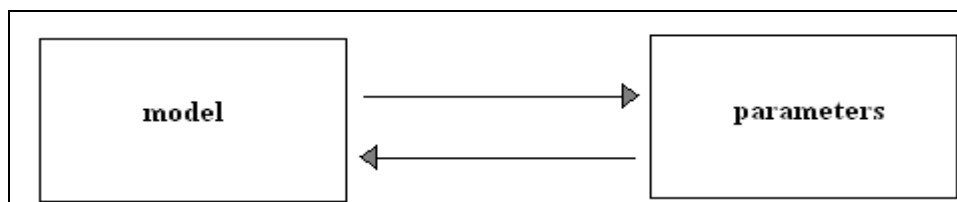


Figura 3. Diagrama de blocos da comunicação entre *model* e *parameters*.

Como a variável *V* da *Equação (2)* não pode ser isolada, coloca-se esta equação na forma padrão de modo que para um determinado valor de *V* a função seja zero.

$$\frac{RT}{(V - b)} - \frac{a}{V(V + b)\sqrt{T}} - P = 0 \quad (5a)$$

$$RTV(V + b)\sqrt{T} - a(V - b) - PV(V + b)\sqrt{T}(v - b) = 0 \quad (5b)$$

$$f(V) = RTV(V + b)\sqrt{T} - a(V - b) - PV(V + b)\sqrt{T}(v - b) \quad (5c)$$

Pode-se observar que temos o número de variáveis igual ao número de equações disponíveis. Então se pode utilizar algum método numérico para equações não-lineares. Isto se torna fácil no Matlab, pois é um software comercial que já tem disponível robustos algoritmos de métodos numérico que nos permitem resolver tais problemas. A função *fzero* (*fzero.m*), por exemplo, é utilizada para resolver funções não-lineares pelo método de Newton-Raphson.

O arquivo *parameters.m* é novamente chamado no arquivo principal (*Main.m*), para calcular o fator de compressibilidade o qual precisa do valor de *R*. Observe que foi utilizado *** pois como *press* e *vol* são vetores, tal símbolo significa que a multiplicação será feita elemento por elemento. Logo *Z* também será um vetor com o mesmo número de elementos de *press* e de *vol*.

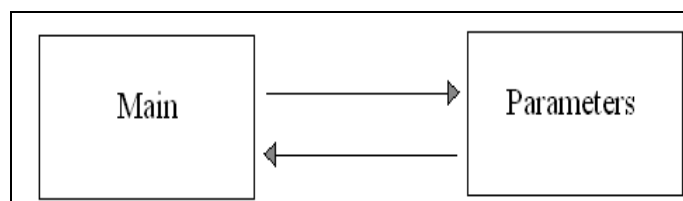


Figura 4. Diagrama de blocos da comunicação entre os *Main* e *parameters*

Finalmente, o comando *plot* é usado para imprimir o vetor Z na ordenada em relação ao vetor press na abscissa. A fim de melhorar a aparência do gráfico foram utilizados:

- A string 'm*-' que plota o gráfico na cor magenta, com símbolos na forma de asteriscos (*) para cada ponto e linha tracejada;
- O comando xlabel que permite nomear o eixo das abscissas, assim como ylabel que permite nomear o eixo das ordenas;
- O comando grid que permite traçar uma grade sob o plot atual.

O diagrama de blocos da *Figura 5*, torna fácil a visualização de como todos os arquivos m-file se comunicam, incluindo também o arquivo embutido *fzero.m*.

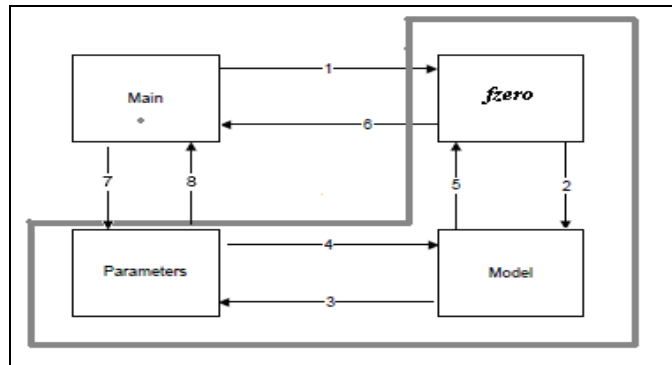


Figura 5. Diagrama de blocos da comunicação entre os arquivos da figura 2

O procedimento 2-3-4-5 é repetido até que a função possa convergir para um valor suficientemente próximo de zero, ou seja, até que seja encontrado através de um processo iterativo um valor para volume específico tal que satisfaça a Equação de *Redlich-Kwong* reescrita como mostra a *Equação (5c)*.

O gráfico gerado pelo Matlab nos possibilita visualizar como varia o fator de compressibilidade (Z) para diferentes pressões do n-butano a uma temperatura de 500 K.

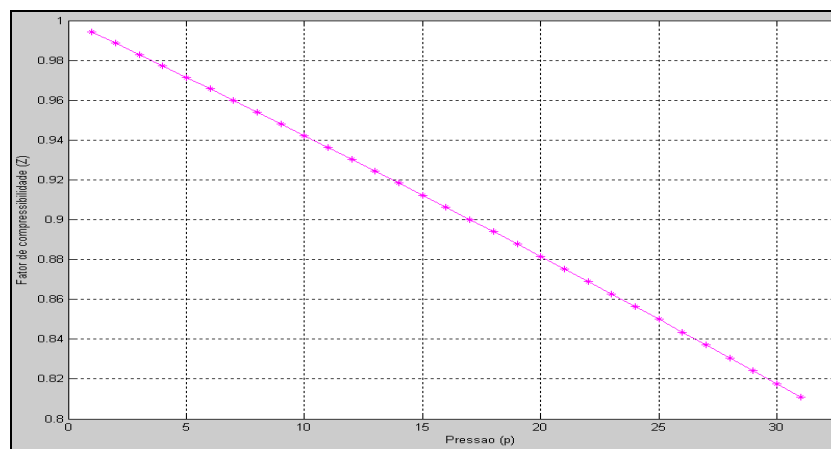


Figura 6. Gráfico do fator de compressibilidade (Z) em função da pressão (p) para o n-butano na temperatura de 500 K

Para um gás ideal o fator de compressibilidade é sempre igual a um, em todas as pressões e temperaturas, porém para um gás real isso não ocorre. Através da visualização gráfica, percebe-se que um gás real aproxima-se do gás ideal quando a pressão é bastante reduzida, ou seja, seu fator de compressibilidade tende para um.

Além disso, o efeito da temperatura também afeta no fator de compressibilidade de um gás não ideal. Pois o desvio do comportamento ideal torna-se maior quando submetido a baixas temperaturas. Com auxílio do Matlab, basta executarmos o programa principal (*main.m*) novamente, mudando a temperatura para 2000 K. Veja o novo gráfico a seguir:

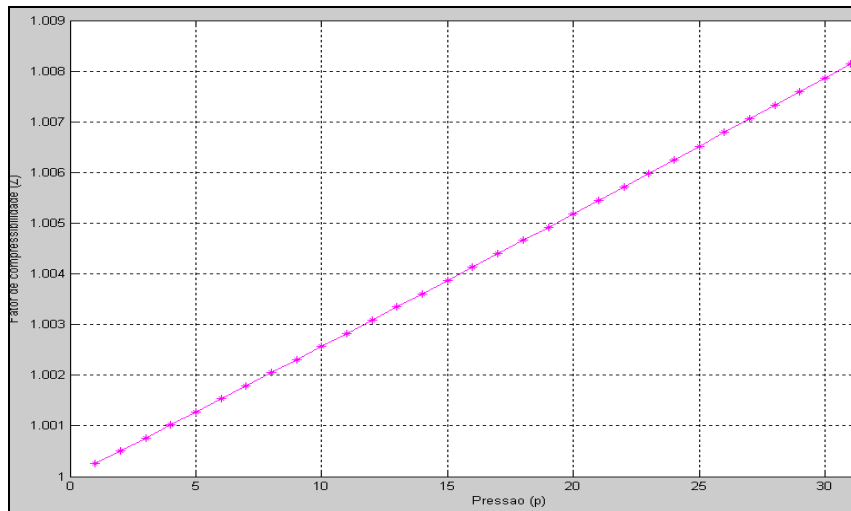


Figura 7. Gráfico do fator de compressibilidade (Z) em função da pressão (p) para o n-butano na temperatura de 2000 K.

Percebe-se que o comportamento do gás na temperatura de 500 K se desvia mais do comportamento de um gás ideal quando comparado ao mesmo na temperatura de 2000 K (veja a *Figura 8*).

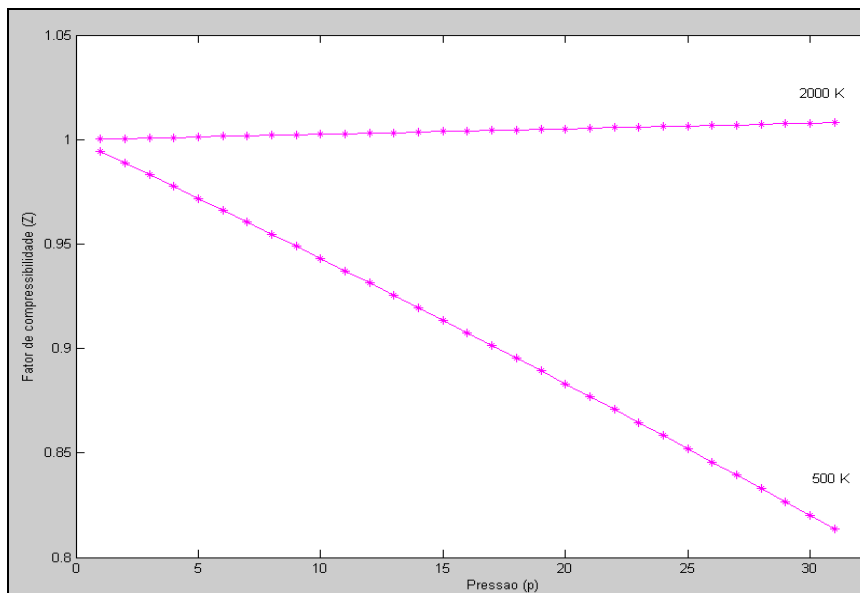


Figura 8. Comparação do comportamento do fator de compressibilidade (Z) em função da pressão (p) para o n-butano nas temperaturas de 2000 e 500 K.

Equilíbrio Líquido-Vapor

No caso de uma substância pura e líquida, se mantivermos a pressão constante, o aquecimento provoca aumento de temperatura até o ponto de ebulição, a partir do qual a substância passa a se transformar da fase líquida para a fase vapor, com estacionamento da temperatura. Isso caracteriza a ebulição do líquido. Entretanto, quando consideramos uma mistura binária, por exemplo, existem ali duas substâncias diferentes, com pontos de ebulição diferentes entre si. Quando se aquece essa mistura, mantida constante a pressão, há um certo momento em que se forma a primeira bolha de vapor. Este ponto é o chamado ponto de bolha da mistura. O aquecimento adicional causará aumento gradual na temperatura do líquido, diferentemente de uma substância pura, na qual a temperatura estaciona nessa etapa de mudança de fase.

De forma análoga, se em vez de líquido tivermos uma mistura na forma de vapor, a redução de temperatura em certo momento provoca a formação da primeira gota de líquido. Este ponto é chamado de ponto de orvalho da mistura. O resfriamento adicional do vapor resultará em diminuição gradual na temperatura e não ficará estacionada, como seria de se esperar em uma substância pura. Como é de se prever, o ponto de bolha e o ponto de orvalho dependem da proporção de cada um dos componentes presentes na mistura.

Considerando o fluxograma da *Figura 9*, supondo que se conheça a temperatura, a pressão e a composição global das correntes.

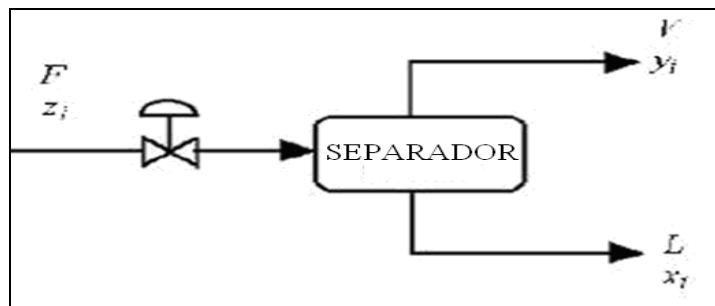


Figura 9. Vaso flash

As frações molares dos produtos químicos na entrada são chamadas (Z_i). No entanto, o líquido e o vapor são separados. As frações molares das substâncias químicas na fase de vapor são chamadas (y_i) e as frações molares das substâncias químicas na fase líquida são chamadas (x_i). Quando o vapor e o líquido estão em equilíbrio, pode-se relacionar as frações molares do líquido e do vapor de acordo com a seguinte equação:

$$y_i = K_i x_i \quad (6)$$

Onde K_i são constantes relacionadas a cada substância que podem ser previstas pela termodinâmica. Para expandir a equação que rege o fenômeno, devemos fazer as seguintes considerações:

$$\sum_{i=1}^{N_{\text{comp}}} y_i = 1 \quad \sum_{i=1}^{N_{\text{comp}}} x_i = 1 \quad (7)$$

Subtraindo as equações anteriores, obtemos:

$$\sum_{i=1}^{N_{\text{comp}}} y_i - \sum_{i=1}^{N_{\text{comp}}} x_i = 0 \quad (8)$$

Substituindo a *Equação (6)* na *Equação (8)*:

$$\sum_{i=1}^{N_{\text{comp}}} (K_i - 1) x_i = 0 \quad (9)$$

Fazendo o balanço de massa no separador, onde F é a vazão molar total de alimentação, V é a vazão molar da fase de vapor e L é a vazão molar da fase líquida, então

$$Z_i F = y_i V + x_i L, \quad F = V + L \quad (10)$$

Dividindo a *Equação (10)* por F e definindo v' como sendo a fração da vazão molar do vapor, então

$$z_i = y_i v' + x_i (1 - v'), \quad v' = \frac{V}{F} \quad (11)$$

Substituindo a *Equação (6)* na *Equação (10)* podemos escrevê-la como:

$$z_i = K_i x_i v' + x_i (1 - v') \quad (12)$$

Reorganizando a *Equação (12)*, obtemos

$$x_i = \frac{z_i}{1 + (K_i - 1)v'} \quad (13)$$

Substituindo a *Equação (13)* na *Equação (9)*, otem-se a equação final:

$$\sum_{i=1}^{N_{\text{comp}}} \frac{(K_i - 1) z_i}{1 + (K_i - 1)v'} = 0 \quad (14)$$

Esta é chamada de Equação de Rachford–Rice. Observe que se os valores de K e as composições de entrada (z_i) são conhecidos, então essa equação linear pode ser resolvida utilizando métodos numéricos (FINLAYSON, 2006).

A partir da tabela abaixo, por exemplo, podemos determinar a fração de fluxo de vapor numa mistura de hidrocarbonetos.

Tabela 1. Composições de entrada e constantes (K) na fase de equilíbrio

Substância química	Z_i	Constantes (K) na fase de equilíbrio
Propano	0,1	6,8
n-Butano	0,3	2,2
n-Pentano	0,4	0,8
n-Octano	0,2	0,052
Total	1	--

A figura a seguir mostra a utilização da metodologia proposta implementada no Matlab para resolver este problema.

```

1  % ===== MAIN =====
2
3  % vapor-liquid equilibrium
4
5  clc, clear all
6
7  est_ini = 0.2; % estimativa inicial
8
9  sol = fzero(@model, est_ini);
10
11  fprintf('A fração do fluxo de vapor, ou seja, a vazao de vapor pela vazao total e igual a: %f', sol)
12

```

(a)

```

1  % ===== MODEL =====
2
3  function f = model(v)
4
5  parameters
6
7  soma = 0;
8
9  for i = 1:4
10
11     num = (K(i)-1)*z(i); % numerador
12
13     denom = 1+(K(i)-1)*v; %denominador
14
15     soma = soma + num / denom;
16
17  end
18
19  f = soma; % Equação de Rachford-Rice

```

(b)

```

1  % ===== PARAMETERES =====
2
3  z = [0.1 0.3 0.4 0.2]; % frcoes do fluxo de entrada de cada especie quimica
4
5  K = [6.8 2.2 0.8 0.052]; % constantes termodinamicas de cada especie quimica

```

(c)

Figura 10. M-files utilizados de forma modular, envolvendo equilíbrio líquido-vapor: (a) programa principal; (b) Parâmetros da mistura e (c) modelo matemático.

Este é um programa simples que utiliza a função embutida *fzero* (*fzero.m*) do Matlab para resolver a Equação de *Rachford–Rice* (*Figura 10a*) partindo de uma estimativa inicial. Esta equação é definida em outro arquivo m-file (*Figura 10b*) de acordo com a metodologia utilizada neste projeto de pesquisa.

No m-file em que a Equação de *Rachford–Rice* foi definida utilizou-se a função *function*, criando-se assim uma nova função chamada *model*. Esta nova função tem uma variável *v* (argumento de entrada) e o valor desta função para um determinado valor de *v* é armazenado na variável *f* (argumento de saída). Mas antes de definir a função é preciso deixar disponíveis, neste m-file, as composições de entrada e as constantes termodinâmicas de equilíbrio líquido-vapor da mistura de hidrocarbonetos. A fim de que a resolução do problema ficasse com uma forma mais didática, tais dados foram definidos em um terceiro m-file (*parameters.m*) conforme mostra a *Figura 10c*. Uma vez criado, basta digitar o nome, com o qual esse arquivo foi salvo, no m-file que contém o modelo matemático (*model.m*).

O mais importante é saber como definir a equação do problema. Para que a expressão não ficasse muito extensa foi utilizado o comando *for*. Neste caso, este comando nos possibilita criar o somatório de acordo com a *Equação (14)*.

Criou-se uma variável chamada *soma*, e atribui-se a esta o valor zero. A variável *soma* é modificada cada vez que o comando *for* é utilizado, como são quatro espécies químicas diferentes, então o *i* utilizado no comando *for* terá quatro variações (*Figura 10b*), variando de um até quatro com incremento unitário. Ou melhor, para cada valor de *i* é acrescentado na soma um novo termo relacionado a cada substância da mistura.

A comunicação dos arquivos é mostrada na *Figura 11*.

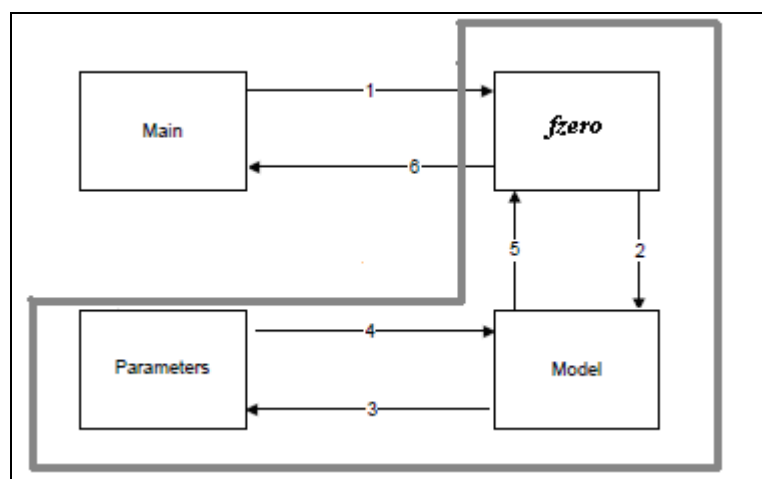


Figura 11. Comunicação entre os arquivos m-files da figura 10

Da mesma forma do exemplo anterior, o procedimento 2-3-4-5 é repetido até que a função possa convergir para um valor suficientemente próximo de zero.

Na janela de comando do Matlab, o resultado para o equilíbrio líquido-vapor ao executarmos o programa principal como mostra a *Figura 10*, será 0.425838, ou seja, aproximadamente 42,58 % da vazão de alimentação da mistura saem na fase de vapor, enquanto que o restante é obtido na fase líquida. Caso o engenheiro, deseje obter mais fase líquida, basta aumentar a composição de uma substância que tenha maior ponto de ebulição em relação à outra que tenha menor ponto de ebulição. Com isso aumentará a temperatura de bolha da mistura. Essa conclusão pode ser obtida se mudarmos as composições da alimentação de entrada no m-file dos parâmetros (*parameters.m*). Uma percentagem de 31,24% é obtida quando mudamos a composição do n-octano de 0.2 para 0.3 e a composição do n-pentano de 0.4 para 0.3, pois a primeira substância tem maior ponto de ebulição em relação à outra.

Balanço de massa com equilíbrio químico

Se você conhecer os fluxos de massa de um processo químico, incluindo os fluxos de entrada e os fluxos de saída, você pode estimar a viabilidade econômica desse processo, acrescentando-se o preço de venda dos produtos, e subtraindo o custo das matérias-primas e tratamento de resíduos. Se esse número não é positivo, o processo não é lucrativo. Então você pode tentar torná-lo economicamente viável, talvez usando outro caminho de reação química ou um novo esquema de separação para purificar o produto (FINLAYSON, 2006).

A presença de reação leva a necessidade de preocupação em relação ao termo de geração, mesmo com uma operação em regime estacionário. Se efetuarmos o balanço em termos mássicos a geração aparecerá somente nos balanços por componentes. Em termos molares, de acordo com os coeficientes

estequiométricos da reação, pode haver geração também em termos globais. Esse fato ocorre quando o somatório dos coeficientes estequiométricos dos reagentes é diferente do somatório dos coeficientes estequiométricos dos produtos. Se lembrarmos do conceito de grau de liberdade, constatamos que o aparecimento do termo de geração em nossas equações trás a necessidade de informações adicionais para podermos resolver o sistema de equações formado pelo balanço de massa. Estas informações adicionais (equações matemáticas) são fornecidas pelas equações estequiométricas e, quando a reação química não ocorre completamente ou há mais de uma reação presente, os parâmetros cinéticos também devem ser utilizados (REKLAITIS, 1983).

Os balanços de massa com a presença de reações químicas são normalmente efetuados na base molar, pois as informações vindas das reações químicas são fornecidas invariavelmente nesta base. Um problema proposto por Chapra & Canale (2008) é descrito na Figura 12:

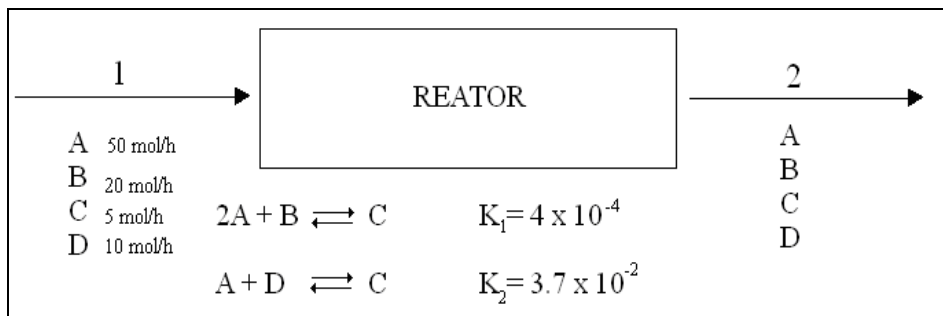


Figura 12. Reações múltiplas em um reator

O objetivo do problema é determinar as vazões de todas as espécies presentes na saída do reator, no qual acontecem duas reações.

Tabela 2. Análise do grau de liberdade da figura 12

	REATOR
VARIÁVEIS	4 + 2
Balanços de Massa	4
Constantes de equilíbrio	2
DF	0

Como a análise do grau de liberdade (DF) do problema é igual a zero, isto significa que o problema está bem posicionado.

Tabela 3. Modelo do problema da Figura 12

EQUAÇÕES	ORIGEM
$n_{a,1} - 2 \cdot \text{eps}_1 - \text{eps}_2 = n_{a,2}$	Balanço de massa no reator para a espécie A
$n_{b,1} - \text{eps}_2 = n_{b,2}$	Balanço de massa no reator para a espécie B
$n_{c,1} + \text{eps}_1 + \text{eps}_2 = n_{c,2}$	Balanço de massa no reator para a espécie C
$n_{d,1} - \text{eps}_2 = n_{d,2}$	Balanço de massa no reator para a espécie D
$K_1 = \frac{n_{c,1} + \text{eps}_1}{(n_{a,1} + 2 \cdot \text{eps}_1)^2 \cdot (n_{c,1} - \text{eps}_1)}$	Constante de equilíbrio da reação 1
$K_2 = \frac{n_{c,1} + \text{eps}_2}{(n_{a,1} + \text{eps}_1) \cdot (n_{c,1} - \text{eps}_2)}$	Constante de equilíbrio da reação 2

Porém, modelado de acordo com a Tabela 3 não se pode obter soluções para o sistema. Portanto devem-se manipular as equações de uma forma que se possa resolver no Matlab, passando os termos do lado direito que constituem as equações para o lado esquerdo. Feito isto, podemos obter seis funções na forma padrão.

Tabela 4. Modelo padrão do problema da Figura 12

EQUAÇÕES	ORIGEM
$f_1 = n_{a,1} - 2*eps_1 - eps_2 - n_{a,2}$	Balanco de massa no reator da espécie A
$F_2 = n_{b,1} - eps_2 - n_{b,2}$	Balanco de massa no reator da espécie B
$f_3 = n_{c,1} + eps_1 + eps_2 - n_{c,2}$	Balanco de massa no reator da espécie C
$F_4 = n_{d,1} - eps_2 - n_{d,2}$	Balanco de massa no reator da espécie D
$f_5 = K_1 - \frac{n_{c,1} + eps_1}{(n_{a,1} + 2*eps_1)^2 * (n_{c,1} - eps_1)}$	Constante de equilíbrio da reação 1
$f_6 = K_2 - \frac{n_{c,1} + eps_2}{(n_{a,1} + eps_1) * (n_{c,1} - eps_2)}$	Constante de equilíbrio da reação 2

O problema descrito pode ser resolvido no Matlab utilizando a metodologia proposta, conforme mostra a Figura 13.

```

1 % -----Programa principal-----
2
3 clear all; clc;
4
5 est=ones(1,6); % estimativas iniciais
6
7 sols=fsolve(@model,est);
(a)

1 % -----PARAMETROS-----
2
3
4 % Concentrações inicial (entrada do reator)
5
6 na1 = 50; nb1 = 20;
7
8 nc1 = 5; nd1 = 10;
9
10
11 % constantes de equilibrio
12
13 K1=4e-4;
14
15 K2=3.7e-2;
16
17
18 % Concentrações finais (saida do reator)
19
20 na2=x(1); nb2=x(2); nc2=x(3); nd2=x(4);
21
22
23 % taxas de reações
24
25 eps1=x(5); eps2=x(6);
26

1 % -----Modelo matematico -----
2
3 function f = model(x)
4
5 parameters
6
7 % balanco de massa no reator
8
9 f(1) = na1 - 2*eps1 - eps2 - na2;
10
11 f(2) = nb1 - eps2 - nb2;
12
13 f(3) = nc1 + eps1 + eps2 - nc2;
14
15 f(4) = nd1 - eps2 - nd2;
16
17 % Equilibrio quimico
18
19 f(5) = ((nc1+eps1)/((na1-2*eps1)^2*(nb1-eps1)))-K1;
20
21 f(6) = ((nc1+eps2)/((na1-eps2)*(nd1-eps2)))-K2;
22

```

Figura 13. M-files utilizados de forma modular, envolvendo balanço de massa com equilíbrio químico: (a) programa principal; (b) Parâmetros do processo e (c) modelo matemático.

Da mesma maneira que os exemplos anteriores, tais como: equações de estado e equilíbrio líquido-vapor, o programa principal (main.m) é utilizado como elo de ligação entre os outros m-files, fornecendo seis estimativas iniciais, pois se trata de um problema que envolve um sistema de equações algébricas. E pelo mesmo motivo foi usada a função fsolve (fsolve.m).

Com o auxílio da Tabela 4, podemos definir tais equações em outro arquivo m-file, o qual será o modelo matemático (model.m) do problema. Todos os dados fornecidos pelo problema se encontram no arquivo m-file dos parâmetros (parameters.com) inclusive a agrupação das variáveis, que pretendem ser obtidas, em um único vetor, pois dessa maneira fica mais fácil criar a nova função e uma estimativa inicial utilizando recursos do Matlab tais como a função ones, a qual neste exemplo cria um vetor com seis elementos, onde cada elemento recebe o valor um.

A comunicação entre os arquivos ocorre conforme apresentado na Figura 11, apenas mudando a função embutida (de fzero para fsolve).

A Figura 14 mostra o resultado obtido na janela de comandos do Matlab do balanço de massa com equilíbrio químico.

```

Command Window
>> sols

sols =

    27.27719029273977    15.58330212342875    18.56975374988341    5.58330212342364    9.15305589998159    4.41669787657552

>> |
    
```

Figura 14. Saída de dados convencional no Matlab para a figura 13

Os resultados mostrados na figura acima requerem um pouco de interpretação, veja as tabelas a seguir:

Tabela 5. Variáveis criadas no exemplo de balanço de massa com equilíbrio químico

M-FILE	PARÂMETRO	PROGRAMA PRINCIPAL
Variáveis	na2 = x(1)	sols(1) = x(1)
	nb2 = x(2)	sols(2) = x(2)
	nc2 = x(3)	sols(3) = x(3)
	nd2 = x(4)	sols(4) = x(4)
	eps1 = x(5)	sols(5) = x(5)
	eps2 = x(6)	sols(6) = x(6)

Como o objetivo do problema é determinar as vazões de saída do reator, então devemos interpretar os resultados em termos de vazões parciais.

Tabela 6. Interpretação dos resultados do exemplo de balanço de massa com equilíbrio químico

CONCLUSÃO
na2 = 27.2772
nb2 = 15.5833
nc2 = 18.5698
nd2 = 5.5833
eps1 = 9.1531
eps2 = 4.4167

A sugestão para este tipo de problema é a utilização de uma saída de dados formatada, pois a interpretação dos resultados é imediata. Então se sugere acrescentar no m-file principal (figura 13a) a função *fprintf*. Esta função permite imprimir textos e conteúdo de matrizes. Podemos aplicá-la da seguinte forma:

```

fprintf('\n na2 =%10.4f \n nb2 =%10.4f \n nc2 =%10.4f \n nd2 =%10.4f \n eps1 =%10.4f \n eps2 =%10.4f', sols(1), sols(2), sols(3), sols(4), sols(5), sols(6))
    
```

No comando acima `\n` insere uma nova linha, e a especificação `%10.4f` insere um campo fixo de largura 10 e o valor será impresso com 4 casas decimais. Os dados são impressos nos mesmos locais em que as especificações se encontram no texto. Vejamos como a saída de dados formatada é obtida na janela de comandos do Matlab.

```

na2 =    27.2772
nb2 =    15.5833
nc2 =    18.5698
nd2 =     5.5833
eps1 =     9.1531
eps2 =     4.4167

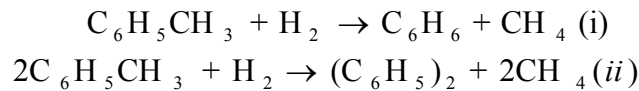
>>
    
```

Figura 15. Saída de dados formatada no Matlab para a figura 13

É importante perceber que para problemas desta natureza todas as soluções obrigatoriamente têm que ser positivas, pois não existem vazões e taxas de reações negativas.

Balanco de massa com reciclo

Um exemplo mais trabalhado, e semelhante ao anterior, é citado a seguir segundo Reklaitis (1983). A desalquilação de tolueno (T) para produzir benzeno é acompanhada por uma reação secundária que produz bifenilo (D).



A primeira etapa do processo de separação recupera benzeno (B) puro e produz dois reciclôs. A sobrecarga é constituída de hidrogênio (H) e metano (M). O produto final da primeira fase de separação é um fluxo de líquido que passa por uma separação em tolueno puro (fluxo de reciclagem), e um subproduto puro: Bifenilo. A alimentação do reator tem uma relação de 5:1 de hidrogênio para o tolueno, enquanto 75% conversão de tolueno é alcançada. Sabendo que o fluxo de saída do reator é de 5% de benzeno e 2% de tolueno, a *Figura 16* ilustra o esquema desse processo.

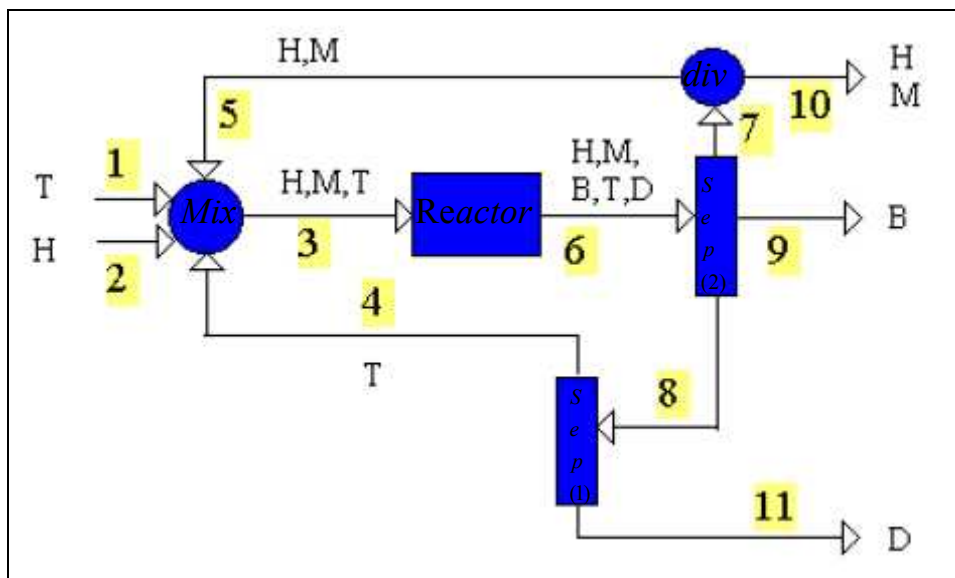


Figura 16. Esquema da desalquilação do Tolueno
Fonte: Adaptada de Reklaitis (1983)

A partir deste exemplo podemos determinar a vazão de todas as substâncias em todas as correntes deste processo de desalquilação do tolueno. A *Tabela 7* mostra a análise do grau de liberdade do esquema descrito na *Figura 16*.

Tabela 7. Análise do grau de liberdade da figura 16

	MIX	DIV	SEP_1	SEP_2	REACTOR	GLOBAL	PROCESSO
VAR	8	6	4	10	8+2	6+2	21+2
B. massa	3	2	2	5	5	5	17
Comp	--	--	--	2	2	--	2
Conv	--	--	--	--	1	--	1
razão	1	--	--	--	1	--	1
Rest. Div	1	--	--	--	--	--	1
Base	--	--	--	1	1	--	1
DF	3	4	2	2	0	3	0

Como a análise do grau de liberdade (DF) do problema é igual a zero para o processo, isto significa que o problema está bem posicionado. Então podemos modelar o problema levando em consideração todo o processo. Analogamente ao exemplo anterior, a tabela abaixo mostra vinte e três funções colocadas na sua forma padrão para que o Matlab possa resolver este problema.

Tabela 8. Modelo padrão do problema da figura 16

EQUAÇÕES	ORIGEM
$f_1 = n_{T,1} + n_{T,4} - n_{T,3}$	Balanço no misturar para o tolueno
$f_1 = n_{H,2} + n_{H,5} - n_{H,3}$	Balanço no misturar para hidrogênio
$F_3 = n_{M,5} - n_{M,3}$	Balanço no misturar para hidrogênio
$f_4 = n_{H,7} - n_{H,10} - n_{H,5}$	Balanço no divisor para hidrogênio
$f_5 = n_{M,7} - n_{M,10} - n_{M,5}$	Balanço no divisor para o metano
$f_6 = n_{T,8} - n_{T,4}$	Balanço no separador_1 para o tolueno
$f_7 = n_{D,8} - n_{D,11}$	Balanço no separador_1 para o difenil
$f_8 = n_{H,6} - n_{H,7}$	Balanço no separador_2 para o hidrogênio
$f_9 = n_{M,6} - n_{M,7}$	Balanço no separador_2 para o metano
$f_{10} = n_{B,6} - n_{B,9}$	Balanço no separador_2 para o benzeno
$f_{11} = n_{T,6} - n_{T,8}$	Balanço no separador_2 para o tolueno
$f_{12} = n_{D,6} - n_{D,8}$	Balanço no separador_2 para o difenil
$f_{13} = n_{H,3} - r_1 - r_2 - n_{H,6}$	Balanço no reator para o hidrogênio
$f_{14} = n_{M,3} + r_1 + 2*r_2 - n_{M,6}$	Balanço no reator para o metano
$f_{15} = n_{B,6} - r_1$	Balanço no reator para o benzeno
$f_{16} = n_{T,3} - r_1 - 2*r_2 - n_{T,6}$	Balanço no reator para o tolueno
$f_{17} = r_2 - n_{D,6}$	Balanço no reator para o difenil
$f_{18} = n_{B,6} - 0.05*(n_{H,6} + n_{M,6} + n_{T,6} + n_{B,6} + n_{D,6})$	Composição do benzeno na saída do reator
$f_{19} = n_{T,6} - 0.02*(n_{H,6} + n_{M,6} + n_{T,6} + n_{B,6} + n_{D,6})$	Composição do tolueno na saída do reator
$f_{20} = n_{T,3} - n_{T,6} - 0.75*n_{T,3}$	Conversão do tolueno
$f_{21} = 5*n_{T,3} - n_{H,3}$	Razão entre o hidrogênio e o tolueno na corrente 5
$f_{22} = n_{H,10}/(n_{H,10} + n_{M,10}) - n_{H,5}/(n_{H,5} + n_{M,5})$	Restrição do divisor
$f_{23} = n_{H,6} + n_{M,6} + n_{T,6} + n_{B,6} + n_{D,6} - 1000$	Base na saída do reator

A Figura 17 mostra a utilização do Matlab para resolver este tipo de problema.

(a)

```

1 % main
2
3 clc; clear all;
4
5 est_ini = 1000*ones(1, 23); % estimativas inicial;
6
7 sol = fsolve(@model, est_ini)
    
```

(b)

```

1 %----- parameters-----
2
3
4 % nomenclatura
5 nT1 = x(1);
6 nH2 = x(2);
7 nH3 = x(3); nM3 = x(4); nT3 = x(5);
8 nT4 = x(6);
9 nH5 = x(7); nM5 = x(8);
10 nH6 = x(9); nM6 = x(10); nT6 = x(11); nB6 = x(12);
11 nH7 = x(14); nM7 = x(15);
12 nT8 = x(16); nD8 = x(17);
13 nB9 = x(18);
14 nH10 = x(19); nM10 = x(20);
15 nD11 = x(21);
16 r1 = x(22); r2 = x(23);
    
```

(c)

```

1 % *****model*****
2
3 function f = model(x)
4
5 parameters
6
7 % balanço para o misturador
8 f(1) = nT1 + nT4 - nT3;
9 f(2) = nH2 + nH5 - nH3;
10 f(3) = nM5 - nM3;
11
12 % balanço para o divisor
13 f(4) = nH7 - nH10 - nH5;
14 f(5) = nM7 - nM10 - nM5;
15
16 % balanço para o sep_1
17 f(6) = nT8 - nT4;
18 f(7) = nD8 - nD11;
19
20 % balanço para o sep_2
21 f(8) = nH6 - nH7;
22 f(9) = nM6 - nM7;
23 f(10) = nB6 - nB9;
24 f(11) = nT6 - nT8;
25 f(12) = nD6 - nD8;
26
27 % balanço para o reator
28 f(13) = nH3 - r1 - r2 - nH6;
29 f(14) = nM3 + r1 + 2*r2 - nM6;
30 f(15) = nB6 - r1;
31 f(16) = nT3 - r1 - 2*r2 - nT6;
32 f(17) = r2 - nD6;
33
34 % composição
35
36 f(18) = nB6 - 0.05*(nH6 + nM6 + nT6 + nB6 + nD6);
37 f(19) = nT6 - 0.02*(nH6 + nM6 + nT6 + nB6 + nD6);
38
39 % conversao
40 f(20) = nT3 - nT6 - 0.75*nT3;
41
42 % razao
43 f(21) = 5*nT3 - nH3;
44
45 % restrições no divisor
46 f(22) = nH10/(nH10 + nM10) - nH5/(nH5 + nM5);
47
48 % base
49 f(23) = nH6 + nM6 + nT6 + nB6 + nD6 - 1000;
50
    
```

Figura 17. M-files utilizados de forma modular, envolvendo balanço de massa com reciclo: (a) programa principal; (b) Parâmetros do processo e (c) modelo matemático.

O programa principal (main.m) é utilizado como elo de ligação entre os outros m-files, fornecendo vinte e três estimativas iniciais, pois se trata de um problema que envolve um sistema de vinte e três equações algébricas. E pelo mesmo motivo foi usada a função fsolve (fsolve.m).

A análise do problema de balanço de massa com reciclo é similar ao exemplo de balanço de massa com equilíbrio químico. Porém este último exemplo nos mostra o quanto é importante a utilização da ferramenta computacional, pois um problema, que a primeira vista, parece ser bastante trabalhoso, com o auxílio do software Matlab e com a utilização da metodologia proposta neste trabalho, tal problema pode ser resolvido sem grandes esforços.

CONSIDERAÇÕES FINAIS

Foram realizados Mini-Cursos no Laboratório de Experimentação Numérica de Processos (LENP), da Unidade Acadêmica de Engenharia Química (UAEQ) na Universidade Federal de Campina grande. Nesses Mini-Cursos, cultivando-se alguns problemas do curso de engenharia química, tangível para o aluno, um trabalho central, então, passou a ser o de introduzir a metodologia proposta juntamente com a formalidade da matemática como referências de modelação dos problemas do curso. Com a utilização do Matlab, o estudante passou a resolver diferentes problemas, utilizando uma mesma metodologia de desenvolvimento.

Foi notado o aumento de alunos do curso de engenharia química que passaram a utilizar o computador ativamente durante o curso, pois tendo em mente a metodologia proposta neste projeto, o aluno percebeu certa facilidade na utilização do software Matlab, comprovando que os problemas mais realísticos não são viáveis resolvê-los analiticamente, mas sim com o auxílio do computador.

Diante disso, um mesmo problema, como já foi explicitado anteriormente, pode ser resolvido rapidamente de várias formas, fazendo com que o aluno, através das diversas soluções, obtenha um senso crítico e tire suas próprias conclusões. As quais podem ser novidades para as literaturas já existentes.

A solução dos trabalhos do curso deve passar obrigatoriamente pelo exercício de explicitação da lógica, desde a fase conceitual, de estruturação da solução do problema, até a processual, de codificação dos programas.

Com uso da metodologia proposta, tornou-se mais fácil a tradução do problema já conceitual e logicamente resolvido nas estruturas formais da linguagem de programação utilizada. E através desta, o estudante pode usar o computador como ferramenta conceitual e simuladora de processos reais.

Podemos dizer que este projeto obteve resultados promissores, pois foi visto o despertar do interesse de alguns alunos para a área das ferramentas computacionais de apoio à engenharia.

AGRADECIMENTOS

Ao CNPq pela concessão da bolsa PIBIC.

REFERÊNCIAS BIBLIOGRÁFICAS

CHAPRA, S. C.; CANALE, R. P, **Numerical Methods for Engineers**, McGraw-Hill, (2008).

FINLAYSON, B. A, **Introduction to Chemical Engineering Computing**, Wiley (2006).

FORBELLONE, A. **Lógica de Programação - A Construção de Algoritmos e Estruturas de Dados**. São Paulo: Ed. Makron Books, 1993.

LAUERMANN, R. A. C.; MÜLLER, F. M.; DE BASTOS, F. P. e FERNÁNDEZ, E. G. **Uma experiência de ensino em Engenharia de Produção com o apoio do AMEM**. In: XXIII Encontro Nacional de Engenharia de Produção, 2003. Ouro Preto.

LONGO, H. L. **Importância da verificação dos resultados do computador**. Anais do XXXIV COBENGE. Passo Fundo: Ed. Universidade de Passo Fundo, Setembro de 2006.

MARTINS FILHO, P. D; SANGLARD, J. H. **A importância da programação de computadores na formação dos engenheiros navais**. In: VI ENCONTRO DE ENSINO DE ENGEHARIA, 2000, ITAIPAVA. Anais do VI EEE. Rio de Janeiro: Escola de Engenharia da UFRJ, 2000. v. I. p. 1-7

PASSOS, J. **O futuro do software**. 2007. Disponível em: <www.facape.br/jocelio/es/material_em_pdf.../ESFuturoSW.pdf>. Acesso em: 01/07/210

REKLAITIS, G. V. **Introduction to material and energy**. John Wiley & Sons, New York, 1983
SMITH, J. M.; VAN NESS, H. C.; ABBOTT, M. M. **Introdução à Termodinâmica da Engenharia química**.
Sétima Edição. Rio de Janeiro: LTC, 2007.

UNIVERSIDADE DO ESTADO DO RIO DE JANEIRO (UERJ). **Curso de MATLAB 5.1: Introdução à
solução de problemas de engenharia**. 2ª edição, 1999. 181p.