Capítulo 2

Solução Numérica de Sistemas Lineares e Não Lineares de Equações Algébricas

Como já discutido anteriormente, modelos estacionários a parâmetros concentrados dão origem a sistemas de equações algébricas, que precisam ser resolvidas para fins de projeto e simulação de equipamentos. Quase sempre, tais modelos não admitem soluções analíticas e, por isto, precisam ser resolvidos numericamente.

Diz-se que uma solução de uma certa equação é numérica se ela é obtida de forma APROXIMADA através da manipulação NUMÉRICA da equação. Neste conceito estão implícitos dois dados importantes:

1- A solução é obtida NUMERICAMENTE; ou seja, através de uma série de testes e procedimentos numéricos e não da manipulação analítica da equação;

2- A solução é sempre APROXIMADA, como resultado da série de testes realizados. Por isto, 2 fatos devem ser salientados:

2a- A solução numérica e a solução real são diferentes, coincidindo apenas dentro de uma certa TOLERÂNCIA;

2b- Várias soluções numéricas diferentes podem ser obtidas, coincidindo apenas dentro de uma certa TOLERÂNCIA.

Neste momento é necessário introduzir a noção de tolerância. Chama-se de tolerância a um certo critério numérico utilizado para permitir a obtenção das raízes da equação em um número finito de testes. Por exemplo, se a equação $f(x)=0$ precisa ser resolvida através de testes numéricos, é conveniente considerar que o número x^* tal que $| f(x^*) | < \varepsilon$, onde ε é um número pequeno (tipicamente da ordem de 10^{-4}, 10^{-5}) é uma raiz aproximada da equação.

É conveniente apresentar o conjunto de técnicas numéricas adequadas para resolver sistemas de equações algébricas em dois grupos: técnicas para solução de sistemas lineares e de sistemas não lineares de equações algébricas. Estas técnicas são apresentadas nas Seções 2.1 e 2.2 que seguem.

2.1- Solução de Sistemas Lineares de Equações Algébricas

A solução de sistemas de equações algébricas lineares é um tópico extremamente importante dentro da área de métodos numéricos, uma vez que quase todos os procedimentos de solução numérica de modelos envolvendo sistemas de equações não lineares, sistemas de equações diferenciais ordinárias ou equações diferenciais parciais envolvem, no seu cerne, a solução de sistemas lineares.

Além disso, os sistemas lineares também ocorrem diretamente na solução de diversos problemas de engenharia. Considere, por exemplo, a combustão parcial de propeno com ar em uma câmara de combustão termicamente isolada e mantida a 1 atm de pressão, esquematizada na Figura 2.1.

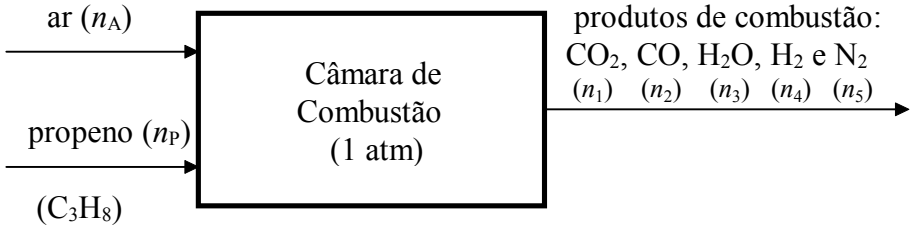

Figura 2.1- Combustão de propeno com ar atmosférico.

Ar (21% molar de O_2) e propeno (C_3H_8) são alimentados em uma certa proporção mássica que permite calcular o número de moles de ar, n_A, e de propeno, n_P, que entram na câmara para uma certa base mássica. Sabendo que os produtos de combustão são apenas o dióxido e o monóxido de carbono, o vapor d'água e o hidrogênio, além do nitrogênio existente no ar de alimentação, queremos calcular a composição do gás de combustão. Precisamos, então, calcular o número de moles, n_i, dos seus 5 componentes. Para tanto, são necessárias 5 equações independentes que relacionem os seus valores. Quatro destas equações podem ser obtidas através do balanço de massa na câmara de combustão, uma para cada espécie química:

$$\text{C:} \quad n_1 + n_2 = 3n_P \tag{2.1}$$

$$\text{H:} \quad 2n_3 + 2n_4 = 8n_P \tag{2.2}$$

$$\text{O:} \quad 2n_1 + n_2 + n_3 = 0{,}21n_A \tag{2.3}$$

$$\text{N:} \quad n_5 = 0{,}79n_A \tag{2.4}$$

enquanto que uma última equação é obtida pelo dado adicional que, na temperatura de combustão da câmara (2030 K), a razão entre o número de moles do monóxido de carbono e do dióxido de carbono é de 0,342:

$$0{,}324n_1 - n_2 = 0 \tag{2.5}$$

Assim, a composição dos gases de combustão pode ser obtida da solução simultânea das Equações (2.1) a (2.5), que formam um sistema linear de equações algébricas. Embora este sistema seja simples o suficiente para ser resolvido por substituição direta, vamos escrevê-lo na forma matricial:

$$\mathbf{A}\,\mathbf{x} = \mathbf{b} \tag{2.6}$$

onde **A** é a matriz de coeficientes, **x** é o vetor de incógnitas e **b** o vetor dos termos não-homogêneos, dados por

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 2 & 2 & 0 \\ 2 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0{,}324 & -1 & 0 & 0 & 0 \end{bmatrix} \qquad \mathbf{x} = \begin{bmatrix} n_1 \\ n_2 \\ n_3 \\ n_4 \\ n_5 \end{bmatrix} \qquad \mathbf{b} = \begin{bmatrix} 3n_{\mathrm{P}} \\ 8n_{\mathrm{P}} \\ 0{,}21n_{\mathrm{A}} \\ 0{,}79n_{\mathrm{A}} \\ 0 \end{bmatrix} \tag{2.7}$$

Nosso objetivo é calcular a solução de sistemas lineares na forma da Equação (2.6), para um número *n* de equações a *n* incógnitas, o que será visto nas próximas seções. Convém ressaltar que, no exemplo acima, o sistema final de equações poderia ser não-linear se a condição adicional fosse dada da forma mais usual, ou seja, como sendo a condição de equilíbrio da reação reversível de formação do gás d'água:

$$\mathrm{CO} + \mathrm{H_2O} \leftrightarrow \mathrm{H_2} + \mathrm{CO_2} \qquad \Rightarrow \qquad n_1 n_4 - K(T)\, n_2 n_3 = 0 \tag{2.8}$$

onde $K(T)$ é a constante de equilíbrio da reação, avaliada na temperatura da câmara de combustão. A solução de equações e sistemas de equações não-lineares será vista mais adiante neste capítulo.

2.1.1- Métodos Diretos e Iterativos de Solução

A Equação (2.6) descreve um sistema linear genérico, na forma matricial, onde **A** é a matriz dos coeficientes, com elementos a_{ij} na linha *i* e coluna *j*, **x** é o vetor de incógnitas e **b** é o vetor dos termos não-homogêneos. Sistemas lineares podem ter quatro tipos de solução: (i) uma única solução, (ii) nenhuma solução, (iii) um número infinito de soluções e (iv) a solução trivial (**x** = **0**). Os sistemas de uma única solução são os mais usuais nas soluções das equações matemáticas de um dado modelo físico.

Os sistemas lineares podem ser resolvidos através de dois tipos de métodos: **os métodos diretos e os métodos iterativos**.

Os métodos diretos consistem em procedimentos, baseados na eliminação algébrica, que obtêm a solução exata em um número fixo de operações. Entre estes métodos, encontram-se: (i) a eliminação Gaussiana, (ii) a eliminação de Gauss-Jordan, (iii) a inversão matricial, (iv) a decomposição LU e outros métodos derivados para matrizes de coeficientes de formas especiais, como o algoritmo de Thomas.

Os métodos iterativos obtêm uma solução aproximada para o sistema de equações utilizando um procedimento iterativo a partir de uma solução aproximada inicial ("chute"). O grau de aproximação requerido na solução controlará o número de operações necessárias para obtê-la. Exemplos de métodos iterativos são: (i) iteração de Jacobi, (ii) iteração de Gauss-Seidel, (iii) sobre-relaxação sucessiva ("sucessive overrelaxation", SOR), (iv) relaxação por linhas (LSOR), (v) método "Alternating-Direction Implicit" (ADI), além de outros métodos específicos para matrizes de coeficientes de formas especiais, como o "Modified Strong Implicit Procedure" (MSIP).

Neste curso, veremos apenas os métodos de eliminação Gaussiana, fatorização LU, algoritmo de Thomas, iteração de Gauss-Seidel e sobre-relaxação sucessiva (SOR), que serão suficientes para a solução dos problemas a serem apresentados.

2.1.2- Eliminação Gaussiana

O processo de eliminação Gaussiana consiste na aplicação sucessiva das operações com linhas para transformar o sistema linear em um outro equivalente, mas cuja matriz dos coeficientes seja triangular superior ou inferior.

Uma matriz triangular superior, **U**, tem todos os elementos abaixo da diagonal principal nulos, enquanto que em uma matriz triangular inferior, **L**, todos os elementos acima da diagonal principal são nulos. Em ambos os casos, o processo de solução do sistema linear é bem simples.

Vejamos o caso de um sistema linear na forma triangular superior

$$\mathbf{U}\,\mathbf{x} = \mathbf{b} \tag{2.9}$$

onde **U** tem a forma

$$\mathbf{U} = \begin{bmatrix} u_{11} & u_{12} & u_{13} & \cdots & u_{1,n} \\ & u_{22} & u_{23} & \cdots & u_{2,n} \\ & & \ddots & & \vdots \\ & \mathbf{0} & & u_{n-1,n-1} & u_{n-1,n} \\ & & & & u_{n,n} \end{bmatrix} \tag{2.10}$$

A solução do sistema (2.9) é facilmente obtida pela chamada *substituição reversa* ("back-substitution"). A última incógnita é facilmente determinada por

$$x_n = \frac{b_n}{u_{n,n}} \tag{2.11}$$

Pode-se, então, obter sucessivamente todos os outros x_i da penúltima para a primeira equação

$$x_i = \frac{b_i - \sum_{j=i+1}^{n} u_{i,j} x_j}{u_{i,i}}, \quad i = n\text{-}1, n\text{-}2, ..., 1 \tag{2.12}$$

As Equações (2.11) e (2.12) constituem o processo de substituição reversa, resolvendo o sistema.

Considere agora um sistema na forma triangular inferior

$$\mathbf{Lx} = \mathbf{b} \tag{2.13}$$

onde **L** tem a forma

$$\mathbf{L} = \begin{bmatrix} l_{11} & & & & \\ l_{21} & l_{22} & & \mathbf{0} & \\ l_{31} & l_{32} & l_{33} & & \\ \vdots & \vdots & \ddots & \ddots & \\ l_{n,1} & l_{n,2} & \cdots & l_{n,n-1} & l_{n,n} \end{bmatrix} \tag{2.14}$$

A solução do sistema (2.13) também é facilmente obtida, utilizando um processo similar ao anterior, mas que começa da primeira para a última linha do sistema, sendo denominado de *substituição progressiva* ("forward substitution")

$$x_1 = \frac{b_1}{l_{11}} \tag{2.15}$$

$$x_i = \frac{b_i - \sum_{j=1}^{i-1} l_{i,j} x_j}{l_{i,i}}, \quad i = 2, 3, ..., n \tag{2.16}$$

Voltemos agora ao processo de eliminação Gaussiana, que transforma um sistema linear com uma matriz de coeficientes de forma genericamente cheia (Equação 2.6), na qual todos os elementos podem ser não-nulos, em um sistema equivalente, de mesma solução, mas cuja matriz de coeficientes é triangular superior (Equação 2.9).

As operações com linhas não modificam a solução do sistema linear e derivam-se das seguintes propriedades de suas equações:

(1) uma equação pode ser multiplicada por uma constante diferente de zero;
(2) a ordem das equações pode ser trocada;
(3) qualquer equação pode ser substituída por uma combinação linear dela mesma com outras equações do sistema.

As operações (1) e (3) afetam apenas o valor dos coeficientes da matriz **A** e dos termos independentes de **b**, não alterando o vetor **x**, enquanto que a operação (2) troca a ordem das incógnitas dentro de **x**, mas não os seus valores. A operação (1) é por vezes utilizada para

alterar a ordem de grandeza dos coeficientes das equações. A operação (2) é utilizada para evitar divisões por zero e diminuir o erro de truncamento na solução final. A operação (3) é a que efetivamente realiza o processo de eliminação.

Considere um sistema linear qualquer com *n* equações, que pode ser escrito na forma

$$\sum_{j=1}^{n} a_{ij} x_j = b_i, \qquad i = 1, ..., n \tag{2.17}$$

Utilizando a operação (2) podemos fazer com que a_{11} seja diferente de zero. Mais do que isto, escolhe-se para ser a primeira equação aquela na qual a_{11} é o maior possível para reduzir os erros de truncamento, operação que é chamada de *pivotamento*. Com a operação (3), podemos substituir cada uma das equações com índices $i = 2, ..., n$ pela combinação linear dela mesma com a primeira equação multiplicada por $-a_{i1}/a_{11}$, o que transforma em zeros todos os elementos da primeira coluna, exceto a_{11}. Toma-se agora o novo sistema, utilizando a operação (2) para colocar na segunda linha aquela equação com o maior elemento existente na segunda coluna para $i = 2, ..., n$ (maior a_{22}). Utilizando de novo a operação (3), substitui-se cada uma das linhas com $i = 3, ..., n$ por uma combinação linear dela própria com a segunda linha multiplicada por $-a_{i2}/a_{22}$, o que elimina os elementos da segunda coluna abaixo de a_{22}. Prosseguindo neste processo de eliminação até a equação *n-1* obtém-se um sistema equivalente ao original, mas cuja matriz dos coeficientes é triangular superior.

Pode-se descrever matematicamente o procedimento acima utilizando a notação a_{ij}^{k}, b_i^{k}, onde o sobrescrito *k* denota o número de eliminações menos um necessário para se obter aquele elemento. O processo de eliminação Gaussiana consiste, então, nas seguintes operações

$$a_{i,j}^{k} = a_{i,j}^{k-1} - \frac{a_{i,k-1}^{k-1}}{a_{k-1,k-1}^{k-1}} a_{k-1,j}^{k-1} \qquad k = 1, 2, ..., n\text{-}1 \tag{2.18}$$

$$b_i^{k} = b_i^{k-1} - \frac{a_{i,k-1}^{k-1}}{a_{k-1,k-1}^{k-1}} b_{k-1}^{k-1} \qquad k = 1, 2, ..., n\text{-}1 \tag{2.19}$$

onde $a_{ij}^{1} = a_{ij}$, $b_i^{1} = b_i$ e o termo $a_{k-1,k-1}^{k-1}$ sempre é obtido por pivotamento antes da eliminação de índice *k*. A matriz dos coeficientes finalmente obtida, $\mathbf{A}^n$, é triangular superior, sendo designada por **U**, e o vetor final de termos independentes é o $\mathbf{b}^n$, sendo dados por

$$\mathbf{A}^n = \mathbf{U} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1,n} \\ & a_{22}^{2} & a_{23}^{2} & \cdots & a_{2,n}^{2} \\ & & \ddots & & \vdots \\ & \mathbf{0} & & a_{n-1,n-1}^{n-1} & a_{n-1,n}^{n-1} \\ & & & & a_{n,n}^{n} \end{bmatrix} \qquad \mathbf{b}^n = \begin{bmatrix} b_1 \\ b_2^{2} \\ \vdots \\ b_{n-1}^{n-1} \\ b_n^{n} \end{bmatrix} \tag{2.20}$$

O sistema linear finalmente obtido é

$$\mathbf{U}\,\mathbf{x} = \mathbf{b}^n \tag{2.21}$$

onde se deve atentar para o fato de que a ordem das incógnitas no vetor **x** pode ter sido trocada pelo processo de pivotamento. A solução do sistema (2.21) é, então, obtida por *substituição reversa*, Equações (2.11) e (2.12), que, neste caso, tornam-se

$$x_n = \frac{b_n^n}{a_{n,n}^n} \tag{2.22}$$

$$x_i = \frac{b_i^i - \sum_{j=i+1}^{n} a_{i,j}^i x_j}{a_{i,i}^i}, \qquad i = n\text{-}1,\ n\text{-}2,\ \ldots,\ 1 \tag{2.23}$$

Nada impede que a eliminação Gaussiana seja usada para resolver, ao mesmo tempo, mais de um sistema linear com a mesma matriz de coeficientes. Para isto, basta fazer a eliminação da seguinte matriz aumentada

$$\left[\mathbf{A} \;\vdots\; \mathbf{b}_1 \;\vdots\; \cdots \;\vdots\; \mathbf{b}_m\right] \tag{2.24}$$

onde $\mathbf{b}_1$, $\mathbf{b}_2$, ..., $\mathbf{b}_m$ são os diversos vetores independentes para os quais se quer a solução do sistema. É claro que a substituição reversa (Equações 2.22 e 2.23) deverá ser executada uma vez para cada vetor de termos independentes.

Exemplo 2.1: A solução do sistema representado pela Equação (2.6) com

$$\left[\mathbf{A} \;\vdots\; \mathbf{b}\right] = \left[\begin{array}{cccc:c} 2 & 1 & 1 & \vdots & 7 \\ 1 & 5 & 2 & \vdots & 17 \\ 1 & 2 & 3 & \vdots & 14 \end{array}\right]$$

é feita em duas etapas:

$$\left[\begin{array}{cccc:c} 2 & 1 & 1 & \vdots & 7 \\ 0 & 4{,}5 & 1{,}5 & \vdots & 13{,}5 \\ 0 & 1{,}5 & 2{,}5 & \vdots & 10{,}5 \end{array}\right] \begin{array}{l} \\ (\text{linha } 2) - 1/2\ (\text{linha}1) \\ (\text{linha } 2) - 1/2\ (\text{linha}1) \end{array}$$

e

$$\mathbf{U} = \left[\begin{array}{cccc:c} 2 & 1 & 1 & \vdots & 7 \\ 0 & 4{,}5 & 1{,}5 & \vdots & 13{,}5 \\ 0 & 0 & 2 & \vdots & 6 \end{array}\right] (\text{linha } 2) - 1/3\ (\text{linha}1)$$

de onde, pelo processo de substituição reversa, obtém-se

$$\mathbf{x} = \left[1 \quad 2 \quad 3\right]$$

Exercício 2.1: Resolva o sistema linear abaixo por eliminação Gaussiana

$$\begin{bmatrix} 2 & -2 & 2 & 1 \\ 2 & -4 & 1 & 3 \\ -1 & 3 & -4 & 2 \\ 2 & 4 & 3 & -2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 7 \\ 10 \\ -14 \\ 1 \end{bmatrix}$$

2.1.3- Fatoração LU

Pode-se provar que a matriz triangular inferior construída com os fatores utilizados no processo de eliminação, cuja forma é

$$\mathbf{L} = \begin{bmatrix} 1 & & & \\ \frac{a_{21}}{a_{11}} & 1 & & \mathbf{0} \\ \frac{a_{31}}{a_{11}} & \frac{a_{32}^2}{a_{22}^2} & 1 & \\ \vdots & \vdots & \ddots & \ddots \\ \vdots & \vdots & \cdots & \frac{a_{n,n-1}^{n-1}}{a_{n-1,n-1}^{n-1}} \quad 1 \end{bmatrix} \tag{2.25}$$

é tal que

$$\mathbf{LU} = \mathbf{A} \tag{2.26}$$

Isto é, obtém-se a fatoração da matriz **A** em matrizes triangulares superior, **U**, e inferior, **L**, sendo que esta última tem uma diagonal principal unitária (método de Doolittle).

A vantagem da fatoração LU é que ela pode ser calculada uma única vez, independentemente do vetor de termos não-homogêneos. Uma vez dado um vetor independente, as matrizes obtidas na fatoração permitem resolver facilmente o sistema dado pela Equação (2.6). Isto pode ser visto substituindo a fatoração no sistema original

$$\mathbf{LUx} = \mathbf{b} \Rightarrow \mathbf{Ux} = \mathbf{L}^{-1}\mathbf{b} = \mathbf{b}^n = \mathbf{c} \tag{2.27}$$

ou

$$\mathbf{Ux} = \mathbf{c} \tag{2.28}$$

$$\mathbf{Lc} = \mathbf{b} \tag{2.29}$$

Assim, uma vez fatorada a matriz dos coeficientes, podemos considerar posteriormente o vetor de termos independentes, **b**, para obter **c**, resolvendo a Equação (2.29) pelo processo de *substituição progressiva*, Equações (2.15) e (2.16), com c_i no lugar de x_i. Após a determinação do vetor **c**, podemos resolver o sistema da Equação (2.28) por *substituição reversa*, Equações (2.11) e (2.12), com c_i no lugar de b_i. Deve-se apenas tomar cuidado em guardar a informação sobre as trocas de posição das linhas durante a fatoração, pois as mesmas trocas estão presentes na ordem das variáveis de **x**.

Exemplo 2.2: Considere o sistema dado no Exemplo 2.1, onde a matriz triangular superior da fatoração da matriz **A** = **LU** foi obtida. De acordo com a Equação (2.25), a matriz triangular inferior é dada por (veja, no Exemplo 2.1, os fatores multiplicativos ao lado da matriz estendida durante a eliminação):

$$\mathbf{L} = \begin{bmatrix} 1 & 0 & 0 \\ 1/2 & 1 & 0 \\ 1/2 & 1/3 & 1 \end{bmatrix}$$

A verificação da fatoração pode ser feita diretamente por multiplicação de **L** e **U**. A solução da Equação (2.29) com o vetor **b** dado no Exemplo 2.1 leva a

$$\mathbf{c} = \begin{bmatrix} 7 & 13{,}5 & 6 \end{bmatrix}$$

gerando, quando substituído na Equação (2.28), o mesmo sistema que já foi resolvido no Exemplo 2.1.

Exercício 2.2: Obtenha a fatoração LU da matriz dos coeficientes do sistema linear dado no Exercício 2.1, calculando também a sua solução.

2.1.4- Algoritmo de Thomas

Diversos métodos de solução de problemas de valor de contorno envolvendo equações diferenciais unidimensionais geram sistemas lineares onde a matriz dos coeficientes tem a chamada forma *tridiagonal*

$$\mathbf{A} = \begin{bmatrix} d_1 & u_1 & & & \\ l_2 & d_2 & u_2 & \mathbf{0} & \\ & \ddots & \ddots & \ddots & \\ & \mathbf{0} & l_{n-1} & d_{n-1} & u_{n-1} \\ & & & l_n & d_n \end{bmatrix} \tag{2.30}$$

Neste caso, podemos fazer a eliminação Gaussiana apenas dos elementos que sabemos que não são nulos (os l_i), sem nenhum pivotamento. Evidentemente, é necessário que $d_i \neq 0$, para todo i. Este procedimento é chamado de algoritmo de Thomas (TDMA).

As equações básicas do processo de eliminação são

$$\bar{d}_1 = d_1, \qquad \bar{d}_i = d_i - \frac{\bar{l}_i}{\bar{d}_{i-1}} u_{i-1}, \qquad i = 2, 3, ..., n \tag{2.31}$$

$$\bar{b}_1 = b_1, \qquad \bar{b}_i = b_i - \frac{\bar{l}_i}{\bar{d}_{i-1}} \bar{b}_{i-1}, \qquad i = 2, 3, ..., n \tag{2.32}$$

A solução do sistema linear é obtida pela substituição reversa que, neste caso, consiste nas equações

$$x_n = \frac{\bar{b}_n}{\bar{d}_n}, \qquad x_i = \frac{\bar{b}_i - u_i x_{i+1}}{\bar{d}_i}, \qquad i = n\text{-}1, n\text{-}2, ...,1 \tag{2.33}$$

Exercício 2.3: Resolva o sistema linear abaixo pelo algoritmo de Thomas.

$$\begin{bmatrix} -2 & 1 & 0 & 0 \\ 1 & -2 & 1 & 0 \\ 0 & 1 & -2 & 1 \\ 0 & 0 & 1 & -2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ -7 \\ -1 \end{bmatrix}$$

2.1.5- Iteração de Gauss-Seidel

O método de Gauss-Seidel é uma variante da iteração de Jacobi. Esta última consiste em rearranjar a Equação (2.17) da seguinte forma

$$x_i^{(k+1)} = \frac{1}{a_{ii}} \left(b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(k)} \right), \qquad i = 1, 2, ..., n \tag{2.34}$$

onde o índice k indica a iteração. Inicia-se o processo iterativo com um valor inicial para o vetor solução, $\mathbf{x}^{(0)}$, utilizando-se sucessivamente a Equação (2.34) para todo i, até que a solução seja obtida dentro da tolerância que foi prescrita. A convergência é garantida se o sistema linear tiver a *dominância diagonal*, isto é

$$|a_{ii}| \geq \sum_{\substack{j=1 \\ j \neq i}}^{n} |a_{ij}|, \qquad i = 1, 2, ..., n \tag{2.35}$$

sendo estritamente maior para, pelo menos, uma linha. Algumas vezes, é possível modificar um sistema linear que não é diagonalmente dominante através de trocas de linhas, para transformá-lo em um com dominância diagonal. Sistemas lineares originários da discretização de equações diferenciais usualmente são diagonalmente dominantes.

O método de Gauss-Seidel consiste em se usar sempre o valor mais recente para cada variável do processo iterativo. Assim, supondo que as linhas são processadas em ordem crescente de seu índice, a iteração é dada por

$$x_i^{(k+1)} = \frac{1}{a_{ii}}\left(b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(k)} \right), \quad i = 1, 2, ..., n \tag{2.36}$$

A convergência também é garantida para sistemas com dominância diagonal.

Exemplo 2.3: Vamos utilizar o método de Gauss-Seidel para resolver o sistema linear já resolvido por eliminação no Exemplo 2.1. Escrevendo as equações do sistema de acordo com a Equação (2.36), temos

$$x_1^{(k+1)} = \frac{1}{2}\left[7 - x_2^{(k)} - x_3^{(k)} \right]$$

$$x_2^{(k+1)} = \frac{1}{5}\left[17 - x_1^{(k+1)} - 2x_3^{(k)} \right]$$

$$x_3^{(k+1)} = \frac{1}{3}\left[14 - x_1^{(k+1)} - 2x_2^{(k+1)} \right]$$

Escolhendo um conjunto inicial de valores para k = 0 ("chute inicial"), as 5 primeiras iterações do método estão na Tabela 2.1. Devido à dominância diagonal, os valores iterados tendem rapidamente à solução exata [1 , 2 , 3].

Tabela 2.1 - Iterações do método de Gauss-Seidel

k	0	1	2	3	4	5
$x_1^{(k)}$	3,5	1,300	0,973	0,968	0,987	0,996
$x_2^{(k)}$	2,7	2,460	2,168	2,048	2,011	2,002
$x_3^{(k)}$	1,7	2,593	2,897	2,979	2,997	3,000

Exercício 2.4: Resolva o sistema linear dado no Exercício 2.3, usando a iteração de Gauss-Seidel, até que três dígitos significativos sejam invariantes em todos os valores do vetor solução.

2.1.6- Sobre-Relaxação Sucessiva (SOR)

Considere um método iterativo qualquer que está sendo usado para determinar a solução de um problema na forma

$$\mathbf{x}^{(k+1)} \leftarrow \mathbf{x}^{(k)} \tag{2.37}$$

como, por exemplo, a iteração de Gauss-Seidel. Podemos tentar aumentar a taxa de convergência do processo, isto é, diminuir o número de iterações necessárias para obter a solução dentro da tolerância prescrita, utilizando a própria "tendência" do processo de convergência. Considere a correção dada por

$$\mathbf{x}_C^{(k+1)} = \mathbf{x}^{(k)} + \omega\left(\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\right) \tag{2.38}$$

onde ω é chamado de fator de relaxação. Note que a diferença entre duas aproximações sucessivas, (*k*) e (*k+1*), fornece a "tendência" do processo de convergência. Assim, ao se fazer ω maior que 1, tenta-se aumentar a velocidade de um processo com boa convergência, enquanto que, ao se fazer ω menor que 1, pode-se tornar convergente um processo originalmente divergente. Desta forma, podemos classificar

- $1 < \omega < 2$: sobre-relaxação
- $0 < \omega < 1$: sub-relaxação

O valor de ω não pode superar 2, ou o processo iterativo se tornará divergente em algum ponto próximo a sua solução.

Exemplo 2.4: Vamos utilizar a sobre-relaxação na solução pelo método de Gauss-Seidel do sistema linear resolvido no Exemplo 2.3. As equações do sistema linear são escritas tal como no Exemplo 2.3, determinando o valor do vetor solução *antes* da correção. Após *todo* o vetor ser obtido, a relaxação é então efetuada.

As 4 primeiras iterações estão na Tabela 2.2, onde fica clara a tendência de convergência para os valores exatos da solução. É patente, também, uma melhoria na velocidade de convergência do processo iterativo, pois as 4 iterações levaram, praticamente, ao mesmo resultado obtido com as 5 iterações do método de Gauss-Seidel. O valor do parâmetro de relaxação foi arbitrariamente escolhido, podendo não ser o melhor possível.

Note que a relaxação também poderia ser feita elemento a elemento de **x**, logo após o seu valor iterado ser obtido. Aplicar-se-ia, desta forma, a Equação (2.38) na forma escalar a cada um dos componentes do vetor **x**.

Tabela 2.2 - Iterações do método de relaxação usando Gauss-Seidel (ω = 1,3)

k	0	1	2	3	4
$x_1^{(k)}$	3,5	0,640	0,946	1,007	1,001
$x_2^{(k)}$	2,7	2,388	1,988	1,992	2,000
$x_3^{(k)}$	1,7	2,861	3,026	3,003	3,000

Exercício 2.5: Resolva o sistema linear dado no Exercício 2.3 usando SOR, variando ω no intervalo [1,2, 1,4], com variações de 0,01, até que três dígitos significativos sejam invariantes em todos os valores do vetor solução. Compare e comente os resultados.

2.1.7- Teste de Convergência

Na descrição dos processos iterativos, falou-se várias vezes em se obter a solução "dentro da tolerância prescrita". Mais adiante, na solução de equações não-lineares, teremos a mesma necessidade de definir quando o processo numérico iterativo já obteve a solução "dentro da tolerância prescrita". O significado desta expressão será aqui analisado.

Primeiramente, é necessário estabelecer uma medida do grau de afastamento (distância) entre a solução exata e a sua aproximação. Como, usualmente, a solução exata não é conhecida, utiliza-se o artifício de comparar aproximações sucessivas da mesma. A distância entre dois escalares é facilmente medida pelo valor absoluto da diferença entre eles:

$$\left| x^{(k+1)} - x^{(k)} \right| \tag{2.39}$$

Para medir a distância entre dois vetores, devemos definir uma norma no espaço vetorial, como por exemplo

$$\|\mathbf{x}\| \equiv \left[\sum_{i=1}^{n} x_i^2 \right]^{\frac{1}{2}} \qquad \text{ou} \qquad \|\mathbf{x}\| \equiv \max_i \left| x_i \right| \tag{2.40}$$

Com uma norma, podemos definir a distância entre vetores e exigir que a mesma satisfaça determinados critérios de convergência que envolvem as chamadas *tolerâncias*. Por exemplo, o critério de *tolerância absoluta* é

$$\left\| \mathbf{x}^{(k+1)} - \mathbf{x}^{(k)} \right\| < \varepsilon_a \tag{2.41}$$

enquanto que o de *tolerância relativa* é dado por

$$\left\| \mathbf{x}^{(k+1)} - \mathbf{x}^{(k)} \right\| < \varepsilon_r \left\| \mathbf{x}^{(k+1)} \right\| \tag{2.42}$$

O critério dado pela Equação (2.42) apresenta problemas para ser satisfeito se a solução é próxima de zero, isto é, $\left\| \mathbf{x}^{(k)} \right\| \to 0$. Um critério misto que corrige esta deficiência é dado por

$$\left\| \mathbf{x}^{(k+1)} - \mathbf{x}^{(k)} \right\| < \varepsilon_a + \varepsilon_r \left\| \mathbf{x}^{(k+1)} \right\| \tag{2.43}$$

onde ε_a é a tolerância absoluta e ε_r é a tolerância relativa.

Às vezes, deseja-se um certo grau de convergência em todos os elementos do vetor **x**, o que leva ao uso dos critérios expressos pelas Equações (2.41), (2.42) ou (2.43), trocando-se a norma do vetor pelo valor absoluto do componente *i* e testando se o critério é válido para todo *i*.

Exemplo 2.5: Utilizando a definição Euclidiana de norma de um vetor (a definição que está à esquerda, na Equação 2.40), vamos comparar agora a convergência dos métodos iterativos usados nos Exemplos 2.3 e 2.4. A Tabela 2.3 contém as normas do vetor **x**

para cada iteração *k,* a distância entre o valor atual deste vetor e o seu valor anterior na iteração *k-1*, e a variação relativa do vetor **x**.

De acordo com a Tabela 2.3, podemos agora comparar, quantitativamente, a taxa de convergência dos dois métodos iterativos. Aquele que utiliza a relaxação atingiu melhores valores tanto na medida do erro absoluto, $\left\|\mathbf{x}^{(k)}-\mathbf{x}^{(k-1)}\right\|$, quanto na medida do erro relativo, $\left\|\mathbf{x}^{(k)}-\mathbf{x}^{(k-1)}\right\|/\left\|\mathbf{x}^{(k)}\right\|$, em 4 iterações, do que os valores obtidos pelo método que não usa a relaxação em 5 iterações.

Caso ε_a tivesse sido escolhido como 0,01, nenhum dos dois métodos teria satisfeito o critério dado pela Equação (2.41), porém, o que usa a relaxação está bem perto disto. Por outro lado, para ε_r = 0,01, o critério dado pela Equação (2.42) é satisfeito por ambos os métodos nas sua últimas iterações. Já o critério dado pela Equação (2.43), com $\varepsilon_a = \varepsilon_r$ = 0,01, é satisfeito por ambos os métodos para k = 4. Entretanto, os valores do erro absoluto, $\left\|\mathbf{x}-\mathbf{x}^{(4)}\right\|$, é de 0,001 para o método com relaxação, mas de apenas 0,017 para o método de Gauss-Seidel. Isto é resultado da menor taxa de convergência do método sem relaxação, que diminui o grau de confiança da aproximação do erro absoluto por $\left\|\mathbf{x}^{(k)}-\mathbf{x}^{(k-1)}\right\|$. Deve-se, portanto, tomar cuidado na especificação de valores muito altos para tolerâncias quando se utilizam métodos com baixa taxa de convergência.

Tabela 2.3 - Convergência dos métodos iterativos.

k	1	2	3	4	5
Gauss-Seidel sem relaxação					
$\left\|\mathbf{x}^{(k)}\right\|$	3,803	3,747	3,742	3,742	3,742
$\left\|\mathbf{x}^{(k)}-\mathbf{x}^{(k-1)}\right\|$	2,386	0,533	0,145	0,0453	0,0131
$\left\|\mathbf{x}^{(k)}-\mathbf{x}^{(k-1)}\right\|/\left\|\mathbf{x}^{(k)}\right\|$	0,627	0,142	0,0389	0,0121	0,00349
Gauss-Seidel com relaxação (ω = 1,3)					
$\left\|\mathbf{x}^{(k)}\right\|$	3,781	3,742	3,742	3,742	—
$\left\|\mathbf{x}^{(k)}-\mathbf{x}^{(k-1)}\right\|$	3,102	0,530	0,0653	0,0104	—
$\left\|\mathbf{x}^{(k)}-\mathbf{x}^{(k-1)}\right\|/\left\|\mathbf{x}^{(k)}\right\|$	0,821	0,142	0,0175	0,00278	—

Exercício 2.5: Resolva o sistema linear dado no Exercício 2.3 usando SOR, utilizando ω = 1,27, e um critério de tolerância relativa com $\varepsilon_r = 10^{-4}$.

2.1.8- Rotinas Disponíveis

Existem inúmeras bibliotecas matemáticas que têm diversas rotinas de solução de sistemas lineares, as quais são altamente otimizadas. Para a comodidade dos alunos deste curso, estão incluídas abaixo para *download* rotinas que implementam o método da eliminação Gaussiana e o algoritmo de Thomas, tanto em linguagem C como em FORTRAN.

Instruções sobre a utilização das rotinas são encontradas nelas próprias na forma de comentários.

2.2- Solução de Sistemas Não-Lineares de Equações Algébricas

De forma pragmática, equações algébricas não-lineares são aquelas que não podem ser colocadas na forma da Equação (2.6). Como regra geral, estas equações não possuem solução analítica e algum tipo de procedimento numérico tem que ser usado para que uma solução seja obtida. É conveniente classificar estes métodos numéricos em dois grupos:

1- Métodos Diretos - não fazem uso de cálculo de derivadas de $f(x)$ nem de aproximações destas derivadas;

2- Métodos Indiretos - fazem uso das derivadas de $f(x)$ ou de aproximações destas derivadas para acelerar a convergência do método numérico.

Apresentam-se a seguir alguns métodos diretos e indiretos comumente utilizados para a solução de problemas de engenharia. A maior parte dos exemplos será aplicada sobre o modelo desenvolvido na Seção 1.3 para o tanque agitado exotérmico, em estado estacionário. Este modelo encontra-se representado pelas Equações (1.58-61), que podem ainda ser representadas na forma:

$$x_A = \frac{1}{1 + \Theta_R \; exp\left(\frac{-\Delta E_R}{\gamma}\right)} \tag{2.44}$$

$$x_B = \frac{\Theta_R \; exp\left(\frac{-\Delta E_R}{\gamma}\right)}{1 + \Theta_R \; exp\left(\frac{-\Delta E_R}{\gamma}\right)} \tag{2.45}$$

$$x_{Se} - \mathrm{x_S} = \frac{\left(1 - x_A\right) W_A - x_B \, W_B}{W_S} \tag{2.46}$$

$$0 = \left(1 - \gamma\right) + \beta \left(\gamma_c - \gamma\right) + \Delta h_R \; \frac{\Theta_R \; exp\left(\frac{-\Delta E_R}{\gamma}\right)}{1 + \Theta_R \; exp\left(\frac{-\Delta E_R}{\gamma}\right)} \tag{2.47}$$

de forma que a resolução da Equação (2.47) permite obter a solução do problema. Nos exemplos numéricos analisados, considera-se que

$$\beta = 1, \; \gamma_c = 1, \; \Theta_R = 50, \; \Delta E_R = 10, \; \Delta h_R = 10$$

que representa um conjunto típico de valores em problemas de reação.

2.2.1- Método de Monte Carlo

Seja $f(x)$ uma função, cuja raiz é procurada num certo intervalo [a,b], onde $f(x)$ troca de sinal (ver Figura 2.2 abaixo).

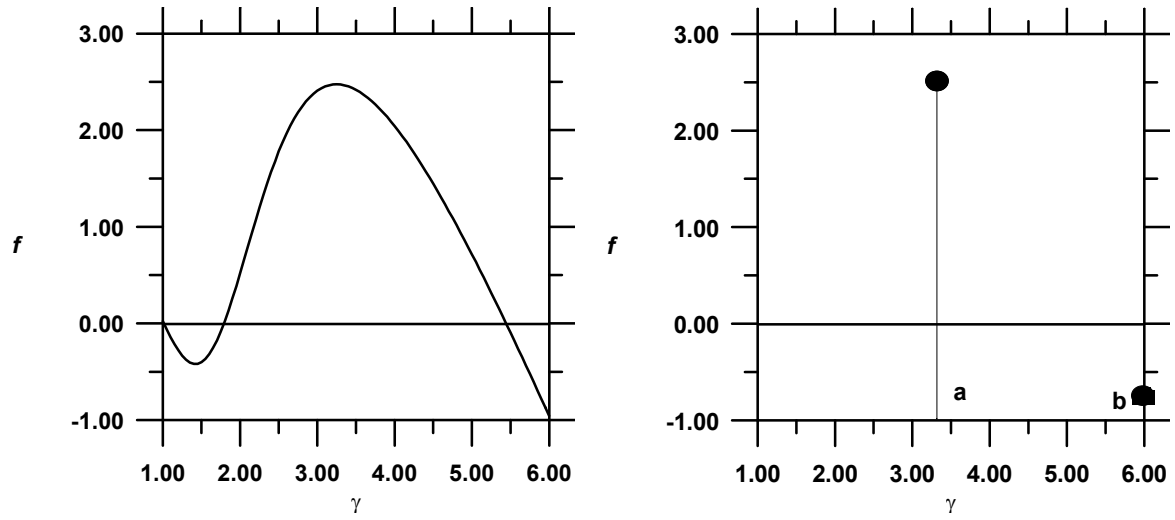

Figura 2.2- Função $f(\gamma)$ (Equação 2.47) e intervalo conhecido onde se encontra a raiz.

O método consiste em gerar pontos aleatoriamente no intervalo [a,b], onde testes numéricos devem ser efetuados; ou seja, onde a função deve ser computada. Seleciona-se então o novo intervalo [a',b'] onde a função troca de sinal e onde novos testes serão realizados. O procedimento deve ser repetido até que o tamanho do intervalo [a',b'] e/ou f(a') e f(b') sejam menores que uma tolerância especificada.

Vantagens:

1- O método sempre funciona e sempre converge para a solução procurada, qualquer que seja a tolerância especificada.

2- O método é facilmente implementável, já que tabelas de números aleatórios estão disponíveis em qualquer livro de Estatística e rotinas de geração de números aleatórios estão disponíveis na maior parte das bibliotecas de rotinas científicas.

Desvantagens:

1- O método é lento e requer o cálculo de $f(x)$ um número elevado de vezes.

2- O método exige o conhecimento prévio de uma região onde a raiz se encontra, o que nem sempre é possível.

3- A extensão do método para problemas multi-variáveis é complexa.

2.2.2- Método da Bissecção

Dadas as mesmas condições da Figura 2.2, o método consiste em verificar o valor da função no ponto médio do intervalo; ou seja, c=(a+b)/2. Se o valor de *f*(c) tem o mesmo sinal de *f*(a), o novo intervalo de busca será [c,b]; caso contrário, o novo intervalo de busca será [a,c] (ver Figura 2.3). O procedimento é repetido até que o tamanho do intervalo de busca e/ou *f*(a) e *f*(b) sejam menores que uma tolerância especificada.

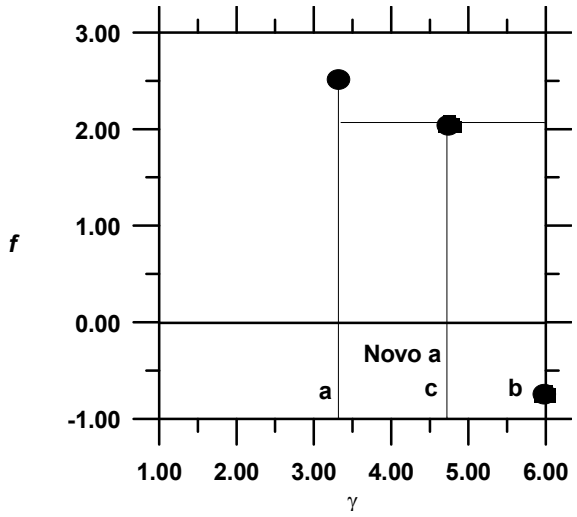

Figura 2.3- Técnica da bissecção

Vantagens:

1- O método sempre funciona e sempre converge para a solução procurada, qualquer que seja a tolerância especificada.

2- O método é facilmente implementável.

3- O método tem desempenho regular e previsível. Após *n* testes, a incerteza ou tamanho do intervalo remanescente é

$$\varepsilon = \left(\frac{1}{2}\right)^n \left[b_o - a_o\right] \quad \Rightarrow \quad n = \frac{ln\left(\frac{\left[b_o - a_o\right]}{\varepsilon}\right)}{ln\,2} \tag{2.48}$$

Sabendo, portanto, que a raiz do problema analisado se encontra no intervalo [3,6], para atingir a precisão de 10^{-4} são necessários 15 passos. A Tabela 2.4 apresenta os resultados obtidos.

Desvantagens:

1- O método é lento e requer o cálculo de *f*(*x*) um número elevado de vezes.

2- O método exige o conhecimento prévio de uma região onde a raiz se encontra, o que nem sempre é possível.

3- A extensão do método para problemas multi-variáveis é complexa.

A técnica da bissecção pode ter seu desempenho bastante melhorado se os pontos de avaliação interna (c) são escolhidos de forma apropriada. Algumas destas técnicas melhoradas são conhecidas pelos nomes de Seção Áurea, Técnicas de Fibonacci, etc., mas não serão aqui analisadas por não introduzirem aspectos significativamente novos na análise do problema.

Tabela 2.4- Resultados obtidos com o método da bissecção

a	b	f(a)	f(b)
.300000E+01	.600000E+01	.240766E+01	-.957508E+00
.450000E+01	.600000E+01	.144198E+01	-.957508E+00
.525000E+01	.600000E+01	.315573E+00	-.957508E+00
.525000E+01	.562500E+01	.315573E+00	-.308127E+00
.543750E+01	.562500E+01	.745389E-02	-.308127E+00
.543750E+01	.553125E+01	.745389E-02	-.149477E+00
.543750E+01	.548438E+01	.745389E-02	-.707879E-01
.543750E+01	.546094E+01	.745389E-02	-.316100E-01
.543750E+01	.544922E+01	.745389E-02	-.120637E-01
.543750E+01	.544336E+01	.745389E-02	-.230128E-02
.544043E+01	.544336E+01	.257721E-02	-.230128E-02
.544189E+01	.544336E+01	.138192E-03	-.230128E-02
.544189E+01	.544263E+01	.138192E-03	-.108149E-02
.544189E+01	.544226E+01	.138192E-03	-.471634E-03
.544189E+01	.544208E+01	.138192E-03	-.166718E-03

2.2.3- Método “Regula-Falsi”

Dadas as mesmas condições da Figura 2.2, o método consiste em verificar o valor da função num ponto intermediário do intervalo, obtido por interpolação linear dos dados conhecidos previamente.

$$c = \frac{a f(b) - b f(a)}{f(b) - f(a)} \tag{2.49}$$

Se o valor de $f(c)$ tem o mesmo sinal de $f(a)$, o novo intervalo de busca será [c,b]; caso contrário, o novo intervalo de busca será [a,c] (ver Figura 2.4). Note que, neste caso, não é possível garantir que o intervalo [a,b] vai sempre convergir para zero. Por isto, é conveniente modificar o critério de convergência para

$$|x_{k+1} - x_k| \le \varepsilon_x \quad , \quad |f(x_{k+1})| \le \varepsilon_f \tag{2.50}$$

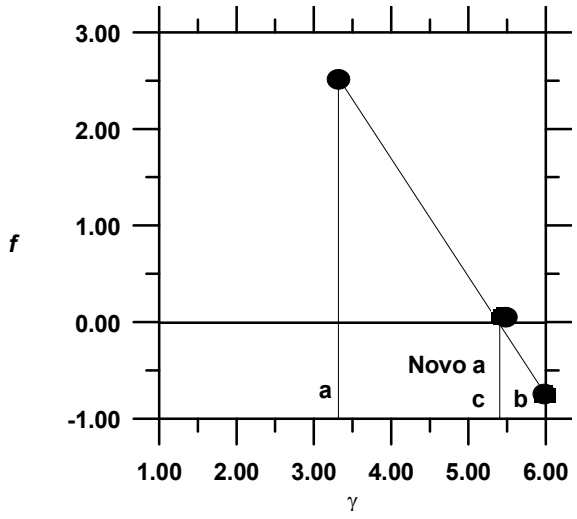

Figura 2.4- Método "Regula-Falsi"

Vantagens:

1- O método sempre funciona e sempre converge para a solução procurada, qualquer que seja a tolerância especificada.

2- O método é facilmente implementável.

3- O método tem desempenho melhor que o da bissecção, muitas vezes comparável ao desempenho dos métodos indiretos. Para fins de comparação, resultados obtidos com o método "Regula-Falsi" são apresentados na Tabela 2.5.

Desvantagens:

1- O método exige o conhecimento prévio de uma região onde a raiz se encontra, o que nem sempre é possível.

2- A extensão do método para problemas multi-variáveis é complexa.

Tabela 2.5- Resultados obtidos com o método "Regula-Falsi"

a	b	c	f(a)	f(b)	f(c)
.300000E+01	.600000E+01	.514639E+01	.240766E+01	-.957508E+00	.482156E+00
.514639E+01	.600000E+01	.543227E+01	.482156E+00	-.957508E+00	.161474E-01
.543227E+01	.600000E+01	.544169E+01	.161474E-01	-.957508E+00	.478850E-03
.544169E+01	.600000E+01	.544197E+01	.478850E-03	-.957508E+00	.141465E-04
.544197E+01	.600000E+01	.544198E+01	.141465E-04	-.957508E+00	.417877E-06
.544198E+01	.600000E+01	.544198E+01	.417877E-06	-.957508E+00	.123437E-07

2.2.4- Método da Substituição Sucessiva

Seja $f(x)$ uma função cuja raiz é procurada. Imaginemos que é possível colocar a função $f(x)=0$ na forma $x= g(x)$. A idéia fundamental é dar uma interpretação iterativa a esta forma alternativa de representar a função $f(x)=0$, para a busca da raiz. Assim, a partir de uma aproximação inicial, podem ser obtidas aproximações melhoradas da raiz de forma iterativa.

Exemplo 2.6: Seja a Equação (2.47), cuja raiz é procurada

$$0 = 2\left(1 - \gamma\right) + 500\,\frac{exp\left(\frac{-10}{\gamma}\right)}{1 + 50\,exp\left(\frac{-10}{\gamma}\right)} \tag{2.51}$$

Esta mesma função pode ser escrita na forma

$$\gamma = 1 + 250\,\frac{exp\left(\frac{-10}{\gamma}\right)}{1 + 50\,exp\left(\frac{-10}{\gamma}\right)} \tag{2.52}$$

que ganha a forma iterativa

$$\gamma_{k+1} = 1 + 250\,\frac{exp\left(\frac{-10}{\gamma_k}\right)}{1 + 50\,exp\left(\frac{-10}{\gamma_k}\right)} \tag{2.53}$$

Partindo da estimativa inicial $\gamma_0=6$, obtêm-se os resultados apresentados na Tabela 2.6.

Tabela 2.6- Resultados obtidos com substituição sucessiva - $\gamma_0=6$.

γ_k	γ_{k+1}
.600000E+01	.552125E+01
.552125E+01	.545492E+01
.545492E+01	.544414E+01
.544414E+01	.544234E+01
.544234E+01	.544204E+01
.544204E+01	.544199E+01
.544199E+01	.544198E+01

É importante observar que a definição da forma recursiva não é única. Por exemplo, a Equação (2.51) poderia também ser colocada na forma

$$\gamma = 2 - \gamma + 500 \frac{exp\left(\frac{-10}{\gamma}\right)}{1 + 50\ exp\left(\frac{-10}{\gamma}\right)} \tag{2.54}$$

$$\gamma_{k+1} = 2 - \gamma_k + 500 \frac{exp\left(\frac{-10}{\gamma_k}\right)}{1 + 50\ exp\left(\frac{-10}{\gamma_k}\right)} \tag{2.55}$$

cujo desempenho está mostrado na Tabela 2.7 abaixo.

Tabela 2.7- Resultados obtidos com substituição sucessiva - γ_0=6.

γ_k	γ_{k+1}
.600000E+01	.504249E+01
.504249E+01	.568876E+01
.568876E+01	.527182E+01
.527182E+01	.555196E+01
.555196E+01	.536758E+01
.536758E+01	.549087E+01
.549087E+01	.540920E+01
.540920E+01	.546366E+01
.546366E+01	.542750E+01
.542750E+01	.545158E+01
.545158E+01	.543558E+01
.543558E+01	.544623E+01
.544623E+01	.543915E+01
.543915E+01	.544386E+01
.544386E+01	.544072E+01
.544072E+01	.544281E+01
.544281E+01	.544142E+01
.544142E+01	.544235E+01
.544235E+01	.544173E+01
.544173E+01	.544214E+01
.544214E+01	.544187E+01
.544187E+01	.544205E+01
.544205E+01	.544193E+01
.544193E+01	.544201E+01
.544201E+01	.544196E+01
.544196E+01	.544199E+01
.544199E+01	.544197E+01
.544197E+01	.544198E+01
.544198E+01	.544197E+01

.544197E+01	.544198E+01

Portanto, é óbvio que a forma escolhida para a recursão influencia o desempenho da técnica. A depender da forma recursiva, a técnica pode levar a convergência rápida, lenta ou nem mesmo convergir. Para que possamos compreender estes resultados, é conveniente perceber que a técnica da substituição sucessiva procura o ponto no qual a curva de iteração $y=g(x)$ se iguala à reta $y=x$, rebatendo os valores obtidos na curva $g(x)$ continuadamente sobre a reta de 45 graus. O procedimento está ilustrado nas Figuras 2.5-8 abaixo.

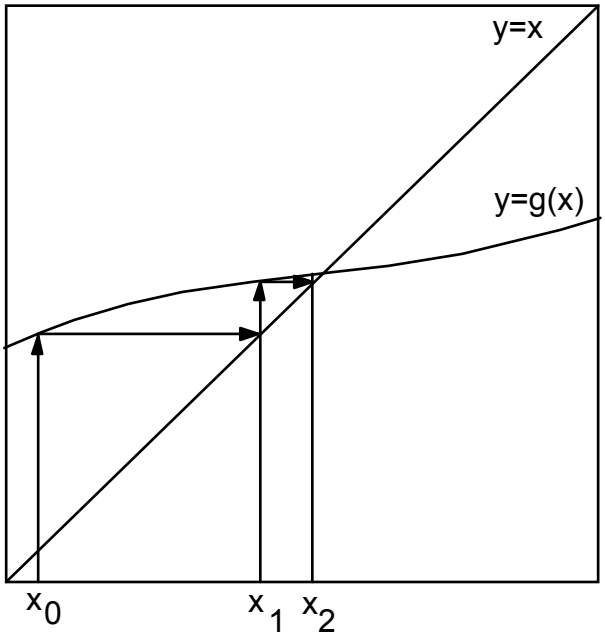

Figura 2.5 - Convergência uniforme do método da substituição sucessiva.

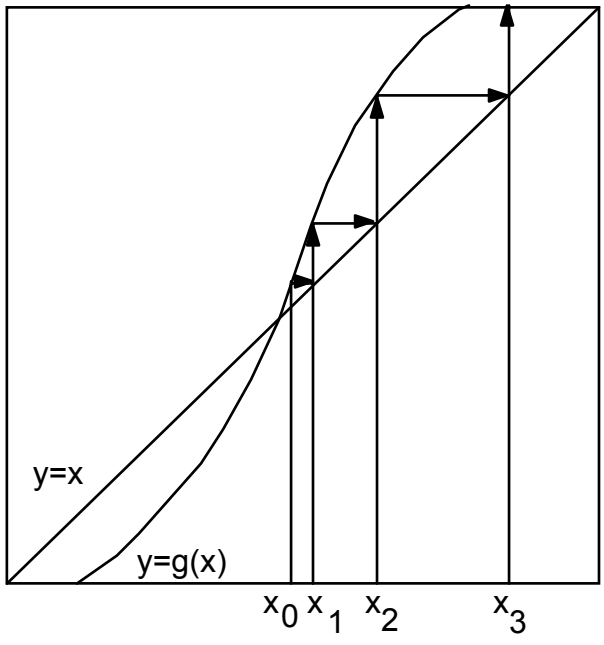

Figura 2.6 - Divergência uniforme do método da substituição sucessiva.

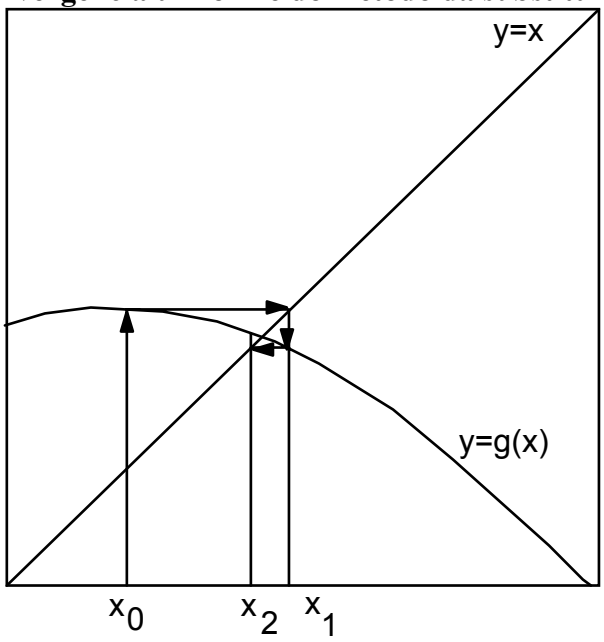

Figura 2.7 - Convergência oscilatória do método da substituição sucessiva.

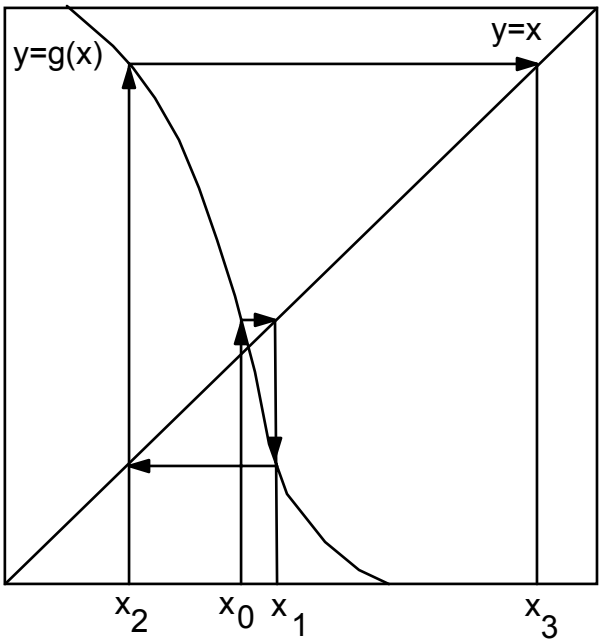

Figura 2.8 - Divergência oscilatória do método da substituição sucessiva.

As questões que se colocam, então, são: Quando o método da substituição sucessiva converge ? Como podemos compreender se uma formulação iterativa do problema é melhor do que uma outra ?

Para respondermos estas perguntas, é conveniente admitir que conhecemos a raiz do problema (x^*) e que começamos o procedimento iterativo nas proximidades desta raiz. Neste caso, pode-se utilizar a expansão em série de Taylor nas proximidades de x^* para escrever:

$$x_{k+1} = g(x_k) \tag{2.56}$$

$$x_{k+1} \approx g(x^*) + \left.\frac{dg}{dx}\right|_{x^*} (x_k - x^*) + \ldots \tag{2.57}$$

$$x_{k+1} \approx x^* + \left.\frac{dg}{dx}\right|_{x^*} (x_k - x^*) + \ldots \tag{2.58}$$

$$\varepsilon_{k+1} \approx \left.\frac{dg}{dx}\right|_{x^*} \varepsilon_k \tag{2.59}$$

onde ε_k é o desvio em relação à raiz x^* na iteração k ($\varepsilon_k = x_k - x^*$). Como o desvio na iteração posterior é uma mera transformação linear do desvio da iteração anterior, diz-se que o método da substituição sucessiva converge linearmente ou que é de ordem 1.

Olhando a Equação (2.59), observa-se facilmente que o desvio na iteração seguinte diminui se

$$\left| \left.\frac{dg}{dx}\right|_{x^*} \right| < 1 \tag{2.60}$$

que é o critério de convergência do método da substituição sucessiva. Quanto mais próximo de zero for a derivada de g em relação a x, calculada na raiz do problema, mais rapidamente converge a técnica proposta. Além disto, se a derivada for negativa, os desvios absolutos obtidos em iterações sucessivas ficam trocando de sinal, o que significa que o procedimento iterativo apresenta comportamento oscilatório.

Exemplo 2.7: No Exemplo 2.6, a fórmula de recursão utilizada tem como derivada

$$\frac{dg}{d\gamma} = 250 \frac{exp\left(\frac{-10}{\gamma}\right)\left(\frac{10}{\gamma^2}\right)}{\left(1 + 50\ exp\left(\frac{-10}{\gamma}\right)\right)^2} \tag{2.61}$$

que na raiz $\gamma = 5.44197$ assume o valor 0.167. Logo, como já visto, a técnica converge. Já a função de iteração mostrada na Equação (2.55) tem como derivada

$$\frac{\mathrm{dg}}{\mathrm{d}\gamma} = -1 + 500\,\frac{exp\left(\frac{-10}{\gamma}\right)\left(\frac{10}{\gamma^2}\right)}{\left(1 + 50\,exp\left(\frac{-10}{\gamma}\right)\right)^2} \tag{2.62}$$

que na raiz assume o valor -0.666. Por isto, o procedimento iterativo oscila e converge mais lentamente.

Como nunca sabemos de ante-mão qual é a raiz procurada, em problemas práticos torna-se necessário calcular a derivada da função de iteração em todo o intervalo de busca da raiz, para que se garanta que o procedimento converge.

Para estimarmos o erro da aproximação, podemos utilizar as relações já desenvolvidas da seguinte forma:

$$x_{k-1} - x^* = x_{k-1} - x_k + x_k - x^* \tag{2.63}$$

$$\left|x_{k-1} - x^*\right| \le \left|x_{k-1} - x_k\right| + \left|x_k - x^*\right| \tag{2.64}$$

$$\left|x_{k-1} - x^*\right| \le \left|x_{k-1} - x_k\right| + \left|\left.\frac{dg}{dx}\right|_{x^*}\right| \left|x_{k-1} - x^*\right| \tag{2.65}$$

$$\left|x_{k-1} - x^*\right| \le \frac{1}{1 - \left|\left.\frac{dg}{dx}\right|_{x^*}\right|} \left|x_{k-1} - x_k\right| \tag{2.66}$$

$$\left|x_k - x^*\right| \le \frac{\left|\left.\frac{dg}{dx}\right|_{x^*}\right|}{1 - \left|\left.\frac{dg}{dx}\right|_{x^*}\right|} \left|x_{k-1} - x_k\right| \tag{2.67}$$

ou seja, é possível estimar o erro máximo cometido a partir das sucessivas aproximações obtidas durante o esquema iterativo.

Finalmente, não é difícil compreender que o método pode ser facilmente estendido a problemas multi-variáveis. Neste caso, as transformações ficam na forma:

$$\boldsymbol{f}(\boldsymbol{x}) = \begin{bmatrix} f_1(x_1, x_2, \ldots, x_n) \\ f_2(x_1, x_2, \ldots, x_n) \\ \ldots \\ f_n(x_1, x_2, \ldots, x_n) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \ldots \\ 0 \end{bmatrix} \quad \Rightarrow \quad \begin{bmatrix} x_1 \\ x_2 \\ \ldots \\ x_n \end{bmatrix} = \begin{bmatrix} g_1(x_1, x_2, \ldots, x_n) \\ g_2(x_1, x_2, \ldots, x_n) \\ \ldots \\ g_n(x_1, x_2, \ldots, x_n) \end{bmatrix} \tag{2.68}$$

de forma que a Equação (2.59) passa a ter a seguinte forma vetorial:

$$\boldsymbol{\varepsilon}_{k+1} \approx \boldsymbol{J}_{\boldsymbol{g}}^{*}\, \boldsymbol{\varepsilon}_{k} \tag{2.69}$$

onde $\boldsymbol{J}_{\boldsymbol{g}}^{*}$ é a matriz de derivadas de $\boldsymbol{g}(\boldsymbol{x})$, ou matriz Jacobiana de $\boldsymbol{g}(\boldsymbol{x})$, que contém todas as derivadas de g em relação a todas as variáveis x_i, definida como:

$$\boldsymbol{J}_{\boldsymbol{g}}^{*} = \begin{bmatrix} \dfrac{\partial g_1}{\partial x_1} & \cdots & \dfrac{\partial g_1}{\partial x_n} \\ \cdots & \cdots & \cdots \\ \dfrac{\partial g_n}{\partial x_1} & \cdots & \dfrac{\partial g_n}{\partial x_n} \end{bmatrix} \tag{2.70}$$

Para garantir a convergência do esquema iterativo, agora, condições mais restritivas resultam da Equação (2.69). Para tanto, a matriz Jacobiana precisa ser diagonalizada, como feito algumas vezes na Seção anterior sobre resolução de sistemas lineares. Neste caso, prova-se que, para que haja convergência, é necessário que todos os valores característicos de $\boldsymbol{J}_{\boldsymbol{g}}^{*}$ tenham módulo menor do que 1. Apenas para lembrar, valores característicos são as raízes λ_i da Equação (2.71) mostrada abaixo

$$det\left(\boldsymbol{J}_{\boldsymbol{g}}^{*} - \lambda \boldsymbol{I}\right) = \boldsymbol{0} \tag{2.71}$$

Além disto, a Equação (2.67) pode ser reescrita na forma:

$$\left|\boldsymbol{x}_k - \boldsymbol{x}^*\right| \leq \left(\boldsymbol{I} - \boldsymbol{J}_{\boldsymbol{g}}^{*}\right)^{-1} \boldsymbol{J}_{\boldsymbol{g}}^{*} \left|\boldsymbol{x}_{k-1} - \boldsymbol{x}_k\right| \tag{2.72}$$

Vantagens:

1- O método é facilmente implementável em problemas mono e multi-variáveis;
2- O método não necessita da definição prévia de onde a raiz se encontra;
3- O erro pode ser facilmente estimado.

Desvantagens:

1- Não há garantia de convergência do método sem a análise prévia da função $g(x)$ e a convergência é tanto mais difícil quanto maior a dimensão do problema.

2.2.5- Método de Newton-Raphson

Seja $f(x)$ uma função cuja raiz é procurada. Admitamos, como na Seção anterior, que $f(x)$ pode ser expandida em série de Taylor nas proximidades de um certo ponto x_k. Neste caso,

$$f(x^*) \approx f(x_k) + \left.\frac{df}{dx}\right|_{x_k} (x - x_k) + \ldots \tag{2.73}$$

Admitindo-se que estamos suficientemente próximo da raiz e que a aproximação linear é boa, a Equação (2.73) pode ser utilizada para se estimar o valor da raiz

$$f(x^*) = 0 \approx f(x_k) + \left.\frac{df}{dx}\right|_{x_k} (x^* - x_k) + \ldots \tag{2.74}$$

$$x^* = x_k - \frac{f(x_k)}{\left.\frac{df}{dx}\right|_{x_k}} \tag{2.75}$$

que pode ser obviamente colocada numa forma recursiva como

$$x_{k+1} = x_k - \frac{f(x_k)}{\left.\frac{df}{dx}\right|_{x_k}} \tag{2.76}$$

A Equação (2.76) é a essência do método de Newton-Raphson. Esta técnica iterativa recebe comumente o nome de Newton-Raphson clássico porque aproximações de $f(x)$ com mais termos podem também ser utilizadas, embora o sejam raramente, em virtude da necessidade de se computar derivadas de ordem superior e de não apresentarem desempenho significativamente melhor do que aquele obtido com a Equação (2.76).

O método de Newton-Raphson pode ser ilustrado como na Figura 2.9. Vê-se que a aproximação da raiz é obtida recursivamente como extrapolações lineares da função, a partir de pontos prévios estimados.

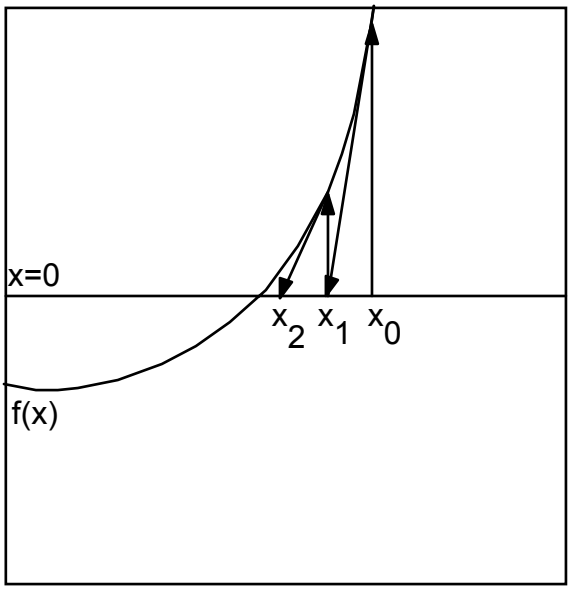

Figura 2.9 - Interpretação geométrica do método de Newton-Raphson.

Exemplo 2.7: Seja a Equação (2.47), cuja raiz é procurada

$$f(\gamma) = 0 = 2(1 - \gamma) + 500 \frac{exp\left(\frac{-10}{\gamma}\right)}{1 + 50\ exp\left(\frac{-10}{\gamma}\right)} \tag{2.77}$$

$$\frac{df}{d\gamma} = 0 = -2\gamma + 500 \frac{exp\left(\frac{-10}{\gamma}\right)\left(\frac{10}{\gamma^2}\right)}{\left(1 + 50\ exp\left(\frac{-10}{\gamma}\right)\right)^2} \tag{2.79}$$

Partindo-se de γ_0=6 e aplicando o algoritmo de Newton-Raphson, obtêm-se os resultados apresentados na Tabela 2.8.

Tabela 2.8 - Resultados obtidos com Newton-Raphson - γ_0=6

γ_k	f_k	$(df/d\gamma)_k$	γ_{k+1}
.600000E+01	-.957508E+00	-.175949E+01	.545580E+01
.545580E+01	-.230453E-01	-.166811E+01	.544199E+01
.544199E+01	-.199642E-04	-.166521E+01	.544198E+01
.544198E+01	-.151540E-10	-.166521E+01	.544198E+01

Do exemplo anterior, vê-se que a técnica de Newton-Raphson converge usualmente de forma muito mais rápida que no caso anterior. Para que iso possa ser colocado num contexto mais rigoroso, é conveniente fazer uma análise de erros mais detalhada. Vejamos primeiramente que a técnica obedece a fórmula geral:

$$(x_{k+1} - x_x) \left.\frac{df}{dx}\right|_{x_k} + f(x_k) = 0 \tag{2.80}$$

Fazendo a expansão em série de Taylor nas proximidades da raiz até os termos de segunda ordem (em virtude da derivada):

$$\begin{aligned}(x_{k+1} - x_k)\left(\left.\frac{df}{dx}\right|_{x^*} + \left.\frac{d^2f}{dx^2}\right|_{x^*} (x_k - x^*) \right) + \\ \left(f(x^*) + \left.\frac{df}{dx}\right|_{x^*} (x_k - x^*) + \frac{1}{2} \left.\frac{d^2f}{dx^2}\right|_{x^*} (x_k - x^*)^2 \right) = 0\end{aligned} \tag{2.81}$$

Lembrando que $f(x^*)=0$ e utilizando a variável desvio $\varepsilon = x - x^*$, chega-se a:

$$\varepsilon_{k+1} = \frac{\left(\left. \frac{d^2 f}{dx^2} \right|_{x^*} \right)}{2 \left(\left. \frac{df}{dx} \right|_{x^*} \right)} \varepsilon_k^2 \tag{2.82}$$

o que significa dizer que o método de Newton-Raphson SEMPRE converge (a não ser que a derivada de $f(x)$ na raiz seja identicamente igual a zero) se a aproximação está SUFICIENTEMENTE próxima da raiz. Isto porque, quando ε_k é pequeno, ε_{k+1} é muito menor que ε_k . (Diz a Lei de Murphy do chute inicial que a sua estimativa inicial não estará suficientemente próxima da solução, de forma que não há garantia na prática de que o método convergirá !!!!) Devido à forma da Equação (2.82), diz-se que o método de Newton-Raphson apresenta convergência quadrática.

O método de Newton-Raphson pode ser facilmente estendido para problemas multi-dimensionais na forma:

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \boldsymbol{J}_f^{-1} \boldsymbol{f}_k \tag{2.83}$$

onde $\boldsymbol{J}_f$ é a matriz Jacobiana da função $\boldsymbol{f(x)}$.

Exemplo 2.8: Seja o modelo do tanque com reação, descrito originalmente na forma:

$$f_1(x_A, \gamma) = 1 - x_A \left(1 + \Theta_R \, exp\left(\frac{-\Delta E_R}{\gamma} \right) \right) \tag{2.84}$$

$$f_2(x_A, \gamma) = (1 - \gamma) + \beta (\gamma_c - \gamma) + \Delta h_R \, x_A \, \Theta_R \, exp\left(\frac{-\Delta E_R}{\gamma} \right) \tag{2.85}$$

cuja matriz Jacobiana é dada por:

$$\boldsymbol{J}_f = \begin{bmatrix} -\left(1 + \Theta_R \, exp\left(\frac{-\Delta E_R}{\gamma} \right) \right) & -x_A \, \Theta_R \, exp\left(\frac{-\Delta E_R}{\gamma} \right) \left(\frac{\Delta E_R}{\gamma^2} \right) \\ \Delta h_R \, \Theta_R \, exp\left(\frac{-\Delta E_R}{\gamma} \right) & -1 - \beta + \Delta h_R \, x_A \, \Theta_R \, exp\left(\frac{-\Delta E_R}{\gamma} \right) \left(\frac{\Delta E_R}{\gamma^2} \right) \end{bmatrix} \tag{2.86}$$

Partindo-se de x_A=0 e γ_0=6 e aplicando o algoritmo de Newton-Raphson, obtêm-se os resultados apresentados na Tabela 2.9.

Vantagens:

1- O método não necessita da definição prévia de onde a raiz se encontra;
2- O método sempre converge, se a estimativa inicial da raiz for boa.

Tabela 2.9 - Resultados obtidos com o método de Newton-Raphson - x_{A0}=0 e γ_0=6

x_{Ak}	γ_k	f_{1k}	f_{2k}	x_{Ak+1}	γ_{k+1}
.000000E+00	.600000E+01	.100000E+01	-.100000E+02	.957508E-01	.552125E+01
.957508E-01	.552125E+01	.121677E+00	-.121677E+01	.111174E+00	.544413E+01
.111174E+00	.544413E+01	.321771E-02	-.321771E-01	.111604E+00	.544198E+01
.111604E+00	.544198E+01	.251297E-05	-.251297E-04	.111604E+00	.544198E+01

Desvantagens:

1- O método requer o cálculo das derivadas da função, o que pode inviabilizar sua implementação em muitos casos;

2- O método requer a inversão da matriz Jacobiana em cada iteração, o que consome bastante tempo computacional e pode tornar o método lento, a despeito da redução sensível do número total de iterações.

Visando manter as características do método de Newton-Raphson e eliminar a sua principal desvantagem, que é o de exigir o cálculo e inversão da matriz de derivadas, uma ampla família de técnicas numéricas foi desenvolvida, mas não será aqui discutida. Textos de métodos numéricos devem ser consultados para estudos adicionais. Fica como exemplo apenas o conhecido método da secante, para problemas uni-variáveis, onde a derivada é substituída por uma aproximação numérica na forma:

$$x_{k+1} = x_k - \frac{x_k - x_{k-1}}{f(x_k) - f(x_{k-1})} f(x_k) \tag{2.87}$$

Repare, no entanto, que a extensão do método da secante para problemas multi-variáveis não é trivial.

2.2.6- Multiplicidades

Ao contrário de sistemas lineares de equações, que admitem apenas uma única solução, sistemas não lineares de equações podem apresentar soluções diferentes para um mesmo conjunto de equações. Vejamos, por exemplo, os resultados obtidos para o tanque de reação quando utilizamos as estimativas iniciais x_{A0}=1, γ_0=1 e x_{A0}=0.5, γ_0=2.

É impressionante descobrir que o tanque de reação possa operar de até três formas distintas, mantidas as mesmas condições de operação e o mesmo conjunto de parâmetros!!!!! Este fenômeno é extremamente relevante para o engenheiro, haja visto que certamente se deseja que o tanque real atinja uma das três possíveis soluções do problema (a que leva ao maior lucro da operação). Para que isto seja feito, é necessário desenvolver uma estratégia apropriada de partida do equipamento. O estudo de fenômenos relacionados à existência de multiplicidades e às consequências disto no projeto e operação de equipamentos é uma das áreas de pesquisa de maior interesse do Laboratório de Modelagem, Simulação e Controle de Processos do Programa de Engenharia Química da COPPE.

Tabela 2.10 - Resultados obtidos com o método de Newton-Raphson - x_{A0}=1 e γ_0=1

x_{Ak}	γ_k	f_{1k}	f_{2k}	x_{Ak+1}	γ_{k+1}
.100000E+01	.100000E+01	-.227000E-02	.227000E-01	.997446E+00	.101277E+01
.997446E+00	.101277E+01	-.143947E-04	.143947E-03	.997430E+00	.101285E+01
.997430E+00	.101285E+01	-.621365E-09	.621365E-08	.997430E+00	.101285E+01

Tabela 2.11 - Resultados obtidos com o método de Newton-Raphson - x_{A0}=0.5 e γ_0=2

x_{Ak}	γ_k	f_{1k}	f_{2k}	x_{Ak+1}	γ_{k+1}
.500000E+00	.200000E+01	.331551E+00	-.315513E+00	.890434E+00	.154783E+01
.890434E+00	.154783E+01	.399446E-01	-.399446E+00	.783862E+00	.208069E+01
.783862E+00	.208069E+01	-.104451E+00	.104451E+01	.829402E+00	.185299E+01
.829402E+00	.185299E+01	-.173270E-01	.173270E+00	.840877E+00	.179561E+01
.840877E+00	.179561E+01	-.122486E-02	.122486E-01	.841822E+00	.179089E+01
.841822E+00	.179089E+01	-.839945E-05	.839945E-04	.841829E+00	.179086E+01
.841829E+00	.179086E+01	-.406291E-09	.406291E-08	.841829E+00	.179086E+01

Exercício 2.6: Seja a função não linear descrita abaixo

$$2\,x\,-\,1\,-\,2\,sen(x)\,=\,0 \tag{2.88}$$

Calcule a raiz do problema usando cada uma das técnicas apresentadas anteriormente.

Exercício 2.7: Reconstrua o modelo do reator supondo que a reação é de segunda ordem, como na equação abaixo:

$$R_A\,=\,-\,K_0\,\exp\left(\frac{-\Delta E}{RT}\right)\,C_A^2 \tag{2.89}$$

Para o mesmo conjunto de parâmetros ustilizado nos exemplos anteriores, calcule as condições de operação no estado estacionário.