

Capítulo 4.

VERIFICAÇÃO DE ERROS NUMÉRICOS 1D EM MALHAS UNIFORMES

Conforme visto no capítulo 1, três tipos de métodos podem ser empregados na solução de um problema: experimentais, analíticos e numéricos. Os erros envolvidos nestes métodos, mostrados na Fig. 4.1, são:

- nos resultados experimentais: erros experimentais;
- nas soluções analíticas: erros de modelagem; e
- nas soluções numéricas: erros de modelagem e erros numéricos.

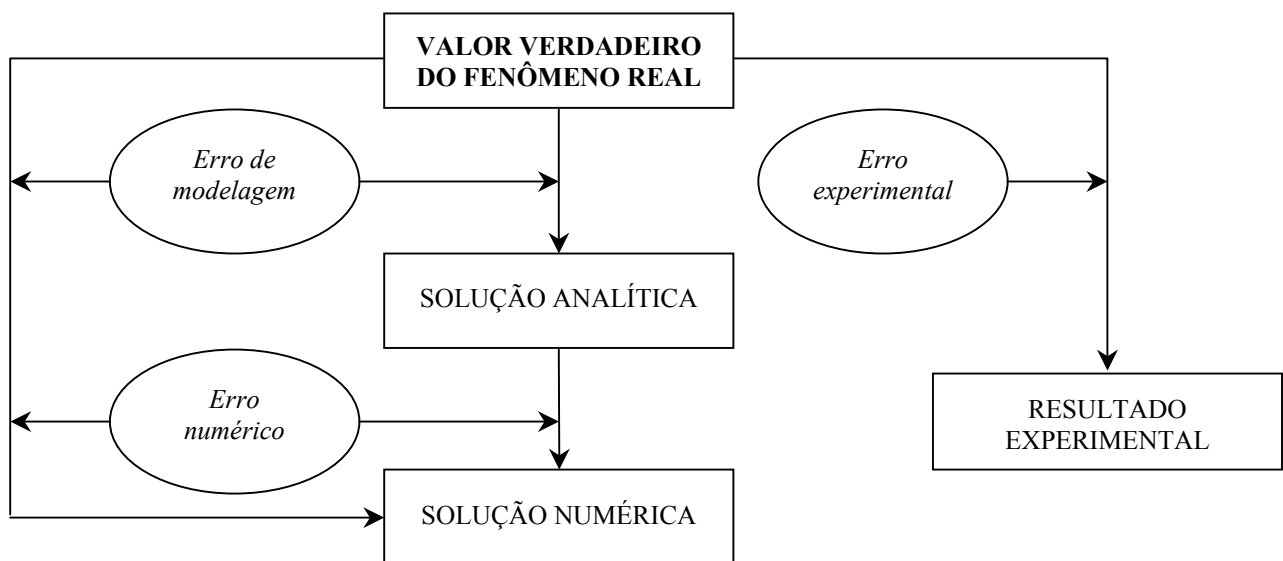


Figura 4.1 Erros envolvidos nos métodos da engenharia.

Erro experimental é a diferença entre o valor verdadeiro de uma variável de interesse e o seu resultado experimental (ABNT, 1997).

Erro de modelagem é a diferença entre o valor verdadeiro de uma variável de interesse e a sua solução analítica exata (Ferziger & Peric, 1999). Ele é causado pelas simplificações feitas sobre

o fenômeno real na concepção do modelo matemático. Os erros de modelagem afetam tanto as soluções analíticas quanto as numéricas porque ambas se baseiam em modelos matemáticos.

Erro numérico (E) é a diferença entre a solução analítica exata (Φ) de uma variável de interesse e a sua solução numérica (ϕ), isto é, (Ferziger & Peric, 1999)

$$E(\phi) = \Phi - \phi \quad (4.1)$$

Portanto, a solução numérica ideal é igual à solução analítica exata do problema, ou seja, é aquela em que o erro numérico é nulo.

Na prática, o valor verdadeiro é desconhecido e, portanto, consegue-se apenas estimar o valor do erro experimental ou do erro de modelagem. Este valor estimado é denominado de incerteza (U). A mesma situação ocorre com relação ao erro numérico. Normalmente, em situações práticas uma solução numérica é obtida porque a solução analítica é desconhecida. Assim, o erro numérico também tem que ser estimado. Os processos usados para estimar cada tipo de erro da Fig. 4.1 são:

- análise de incerteza (ABNT, 1997), para erros experimentais;
- validação (AIAA, 1998), para erros de modelagem; e
- verificação (AIAA, 1998), para erros numéricos.

Este capítulo é dedicado ao processo de verificação de soluções numéricas. Em outras palavras, como estimar o erro numérico ou calcular a incerteza numérica, que é definida por

$$U(\phi) = \phi_{\infty} - \phi \quad (4.2)$$

onde

ϕ = solução numérica

ϕ_{∞} = solução analítica estimada

$U(\phi)$ = incerteza numérica de ϕ

A incerteza de uma solução numérica é calculada com os chamados estimadores de erro.

A magnitude aceitável para o erro numérico depende: da finalidade da solução numérica (exercício acadêmico, pré-projeto, projeto); dos recursos financeiros disponíveis; do tempo disponível para realizar as simulações; e dos recursos computacionais (memória e velocidade de processamento) disponíveis.

Importância de se estimar o erro numérico:

- 1) $E >$ aceitável: compromete-se a confiabilidade do uso da solução numérica;

- 2) $E <$ necessário: há desperdício de recursos;
- 3) para validar e desenvolver modelos matemáticos; por exemplo, escoamentos turbulentos; neste caso o erro numérico deve ser muito menor do que o erro de modelagem; e
- 4) para otimizar o uso da malha através da homogeneização do erro, diminuindo as necessidades computacionais.

A qualidade de uma estimativa de erro pode ser avaliada através da razão entre o erro estimado (U) e o erro verdadeiro (E) (Marchi, 2001). Quanto mais próximo da unidade estiver esta razão, mais acurada é a estimativa do erro. Quando esta razão é maior ou igual à unidade, a estimativa de erro é considerada confiável. O que se busca é uma estimativa de erro ideal, isto é, quando o erro estimado (U) é igual ao erro verdadeiro (E).

O erro numérico é causado por quatro fontes (Ferziger & Peric, 1999):

- 1) erros de truncamento;
- 2) erros de iteração;
- 3) erros de arredondamento; e
- 4) erros de programação.

4.1 ERROS DE TRUNCAMENTO

Os erros de truncamento resultam das aproximações numéricas feitas na discretização de um modelo matemático (Ferziger & Peric, 1999; Tannehill *et al.*, 1997; Roache, 1998). Quando o erro numérico é causado apenas por erros de truncamento, admite-se que (Ferziger & Peric, 1999; Roache, 1998):

$$E(\phi) = C_1 h^{p_L} + C_2 h^{p_2} + C_3 h^{p_3} + \dots \quad (4.3)$$

onde

C_1, C_2, C_3, \dots = coeficientes que independem de h

p_L, p_2, p_3, \dots = ordens verdadeiras de $E(\phi)$; números inteiros positivos

p_L = ordem assintótica de $E(\phi)$; $p_L \geq 1$; inclinação da curva do erro

num gráfico $\log(|E|)$ versus $\log(h)$ para $h \rightarrow 0$

ϕ = variável de interesse

h = tamanho dos elementos da malha

Neste caso, o erro numérico também é denominado de erro de discretização. Existe uma nítida correspondência entre as Eqs. (4.3) e (2.47). Há dois tipos de estimativas do erro de discretização: *a priori* e *a posteriori* das soluções numéricas.

4.1.1 Estimativas *a Priori*

O objetivo de uma estimativa *a priori* é obter a ordem assintótica da equação diferencial discretizada. No caso de $h \rightarrow 0$, a Eq. (4.3) se reduz a

$$E(\phi) = C_1 h^{p_L} \quad (4.4)$$

Assim, antes de se obter qualquer solução numérica, é possível prever o comportamento assintótico do erro de discretização com relação a h e p_L , conforme esquematizado na Fig. 4.2. Não é possível obter o valor de $E(\phi)$ porque C_1 é desconhecido. Tanto C_1 quanto p_L dependem das aproximações numéricas empregadas. Da Eq. (4.4), $E(\phi) \rightarrow 0$ para $h \rightarrow 0$.

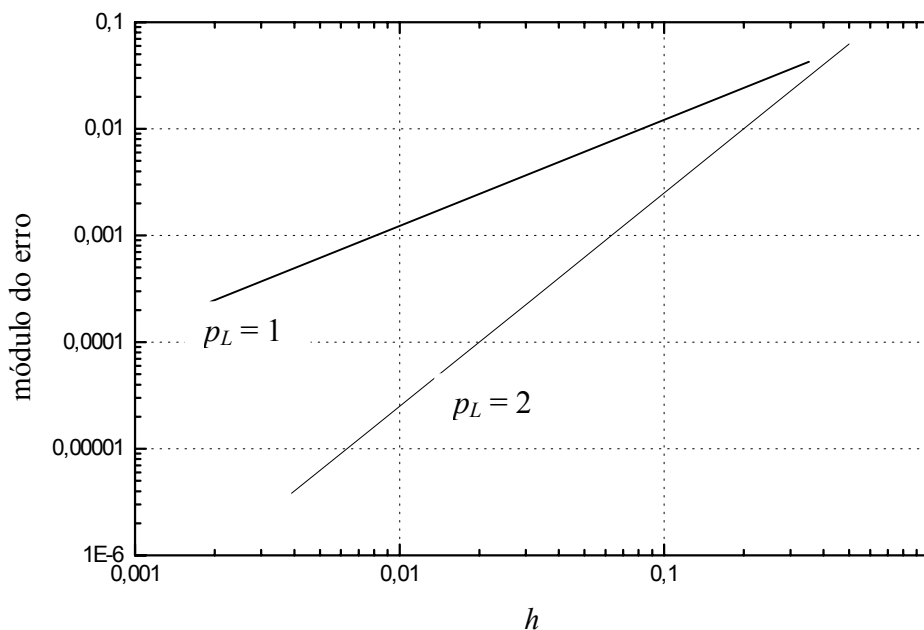


Figura 4.2 Efeito da ordem assintótica (p_L) sobre o erro (E).

Exemplo: $E(\phi_1) = C_1 h_1^{p_L}$ (malha fina)
 $E(\phi_2) = C_1 h_2^{p_L}$ (malha grossa)

Portanto,

$$\frac{E(\phi_1)}{E(\phi_2)} = \frac{C_1 h_1^{p_L}}{C_1 h_2^{p_L}} = \left(\frac{h_1}{h_2} \right)^{p_L}$$

Isto é, quanto maior é p_L , maior é a redução do erro com h . Para o caso de $h_1 = \frac{h_2}{2}$ e $p_L = 2$, temos

$$\frac{E(\phi_1)}{E(\phi_2)} = \left(\frac{\frac{h_2}{2}}{h_2} \right)^2 = \left(\frac{1}{2} \right)^2 = \frac{1}{4}$$

Ou seja, a redução à metade do tamanho dos elementos da malha reduziu a $\frac{1}{4}$ o erro. Quanto maior o p_L , mais rapidamente o erro cai com a redução de h .

4.1.2 Estimativas *a Posteriori*

As estimativas de erro *a posteriori* são usadas para estimar efetivamente a magnitude do erro de discretização. Existem vários métodos que podem ser empregados. Eles podem ser divididos em dois grandes conjuntos:

1. $E(\phi)$ é baseado na solução numérica obtida numa única malha. Em geral o método de elementos finitos (Zhu & Zienkiewicz, 1990) se enquadra neste conjunto.
2. $E(\phi)$ é baseado nas soluções numéricas obtidas em múltiplas malhas. Em geral os métodos de diferenças finitas e de volumes finitos se enquadram neste conjunto. Alguns estimadores deste tipo são (Marchi, 2001): delta, de Richardson, *GCI*, multicoeficientes e convergente. Praticamente todos os estimadores existentes são alguma variação do estimador de Richardson.

4.1.3 Estimador de Richardson baseado na ordem assintótica

A incerteza (U) de uma solução numérica (ϕ) segundo o estimador de erro de Richardson (Roache, 1994) é obtida admitindo-se que

$$U_{Ri}(\phi) = K_U h^{p_L} \tag{4.5}$$

onde

K_U = coeficiente que é admitido independe de h

h = tamanho dos elementos da malha

p_L = ordem assintótica dos erros de truncamento

Com a Eq. (4.2) em (4.5),

$$\phi_{\infty} - \phi = K_U h^{p_L} \quad (4.6)$$

Aplicando-se esta equação a duas malhas diferentes ($h_1 =$ fina e $h_2 =$ grossa) cujas soluções numéricas são ϕ_1 e ϕ_2 , obtém-se

$$\phi_{\infty} - \phi_1 = K_U h_1^{p_L} \quad (4.7)$$

$$\phi_{\infty} - \phi_2 = K_U h_2^{p_L} \quad (4.8)$$

Nestas duas equações são conhecidos os valores de ϕ_1, ϕ_2, h_1, h_2 e p_L , e as incógnitas são ϕ_{∞} e K_U . A solução para ϕ_{∞} é

$$\phi_{\infty}(p_L) = \phi_1 + \frac{(\phi_1 - \phi_2)}{(q^{p_L} - 1)} \quad (4.9)$$

onde a razão de refino de malha (q) é dada por

$$q = \frac{h_2}{h_1} \quad (4.10)$$

A Eq. (4.9) representa a extrapolação de Richardson (Roache, 1994). Com sua substituição na Eq. (4.2), a incerteza ou a estimativa do erro de discretização, segundo o estimador de Richardson, para a solução numérica ϕ_1 na malha fina (h_1) é

$$U_{Ri}(p_L) = \frac{(\phi_1 - \phi_2)}{(q^{p_L} - 1)} \quad (4.11)$$

Deve-se perceber que a estimativa do erro dada pela Eq. (4.11) tem sinal, que pode ser positivo ou negativo.

4.1.4 Estimador de Richardson baseado na ordem aparente

A Eq.(4.6) pode ser reescrita da seguinte forma:

$$\phi_{\infty} - \phi = K_U h^{p_U} \quad (4.12)$$

onde p_U é a ordem aparente (Marchi, 2001) da incerteza. Para malhas grossas, ou práticas, $p_U \neq p_L$, mas para $h \rightarrow 0$, $p_U \rightarrow p_L$.

Aplicando-se a Eq. (4.12) a três malhas diferentes ($h_1 =$ fina, $h_2 =$ grossa e $h_3 =$ supergrossa) cujas soluções numéricas são ϕ_1 , ϕ_2 e ϕ_3 , obtém-se

$$\phi_{\infty} - \phi_1 = K_U h_1^{p_U} \quad (4.13)$$

$$\phi_{\infty} - \phi_2 = K_U h_2^{p_U} \quad (4.14)$$

$$\phi_{\infty} - \phi_3 = K_U h_3^{p_U} \quad (4.15)$$

Nestas três equações são conhecidos os valores de ϕ_1 , ϕ_2 , ϕ_3 , h_1 , h_2 e h_3 , e as incógnitas são ϕ_{∞} , K_U e p_U . Para razão de refino (q) constante entre as malhas h_1 , h_2 e h_3 , isto é,

$$q = \frac{h_3}{h_2} = \frac{h_2}{h_1} \quad (4.16)$$

a solução das Eqs. (4.13) a (4.15) resulta em

$$\phi_{\infty}(p_U) = \phi_1 + \frac{(\phi_1 - \phi_2)}{(q^{p_U} - 1)} \quad (4.17)$$

onde

$$p_U = \frac{\log\left(\frac{\phi_2 - \phi_3}{\phi_1 - \phi_2}\right)}{\log(q)} \quad (4.18)$$

A Eq. (4.17) representa a extrapolação de Richardson com base na ordem aparente (p_U). Com a Eq. (4.18) é possível verificar se para $h \rightarrow 0$, $p_U \rightarrow p_L$, sem qualquer solução analítica, empregando apenas as soluções numéricas.

Com a substituição da Eq. (4.17) na Eq. (4.2), tem-se

$$U_{Ri}(p_U) = \frac{(\phi_1 - \phi_2)}{(q^{p_U} - 1)} \quad (4.19)$$

que é a incerteza da solução numérica (ϕ_1) na malha fina (h_1).

O estimador de Richardson só fornece o valor correto do erro, isto é,

$$U_{Ri}(p_L) = U_{Ri}(p_U) = E(\phi_1) \quad (4.20)$$

em duas situações:

- 1) no caso-limite de $h \rightarrow 0$; e
- 2) quando a equação do erro de discretização, Eq. (4.3), é composta apenas pelo primeiro termo.

Em geral, nenhuma destas duas situações ocorre na prática, e assim,

$$U_{Ri}(p_L) \neq U_{Ri}(p_U) \neq E(\phi_1) \quad (4.21)$$

Apesar disto, pode-se demonstrar (Marchi e Silva, 2002) que

$$\frac{U_{Ri}(p_L)}{E} < 1 < \frac{U_{Ri}(p_U)}{E} \quad (4.22)$$

quando $p_U \rightarrow p_L$ monotonicamente com valores inferiores a p_L , Fig. 4.3. No caso inverso, isto é, quando $p_U \rightarrow p_L$ monotonicamente com valores superiores a p_L , Fig. 4.4, tem-se

$$\frac{U_{Ri}(p_U)}{E} < 1 < \frac{U_{Ri}(p_L)}{E} \quad (4.23)$$

Além disso, só faz sentido empregar U_{Ri} , Eq. (4.19), quando $p_U > 0$.

4.1.5 Estimador *GCI*

De acordo com o estimador *GCI* (*Grid Convergence Index*), a incerteza de uma solução numérica (ϕ_1) é calculada através de (Roache, 1994)

$$U_{GCI}(p_L) = F_s \frac{|\phi_1 - \phi_2|}{(q^{p_L} - 1)} \tag{4.24}$$

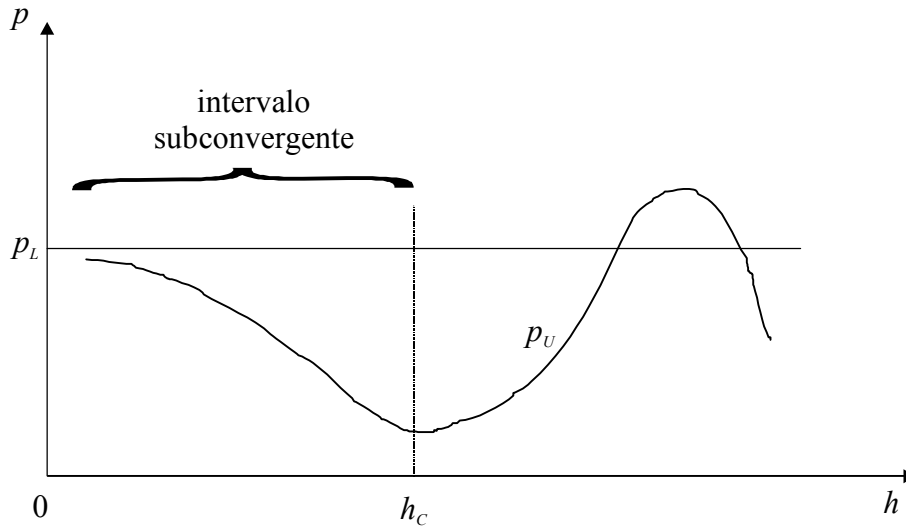


Figura 4.3 Definição do intervalo subconvergente da ordem aparente.

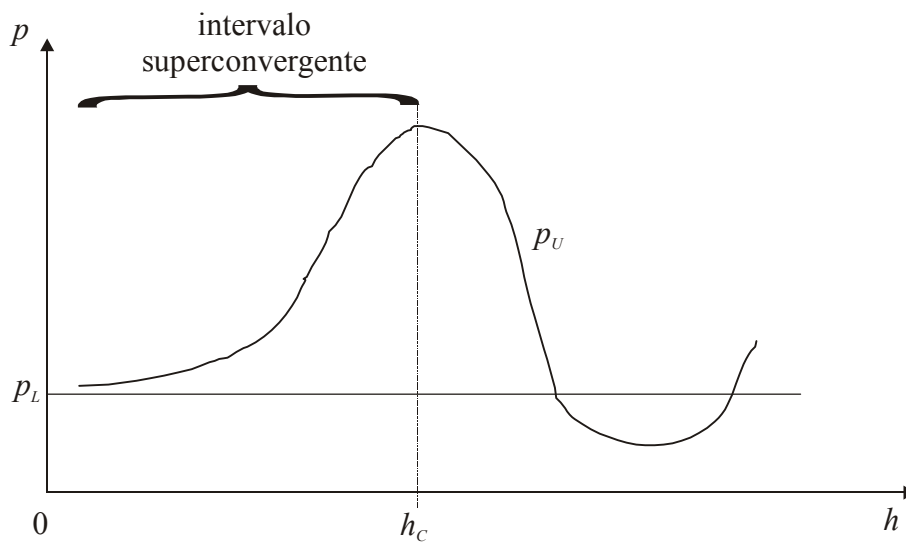


Figura 4.4 Definição do intervalo superconvergente da ordem aparente.

onde F_s é um fator de segurança com valor igual a três, para aplicações em geral. Este estimador também pode ser empregado com a ordem aparente (p_U). Neste caso,

$$U_{GCI}(p_U) = F_s \frac{|\phi_1 - \phi_2|}{(q^{p_U} - 1)} \quad (4.25)$$

Pode-se demonstrar (Marchi, 2001) que

$$\frac{U_{GCI}}{|E(\phi_1)|} \rightarrow F_s \quad \text{para } h \rightarrow 0 \quad (4.26)$$

A representação correta da solução numérica (ϕ_1) e sua respectiva incerteza (U_{GCI}) obtida com o estimador GCI é

$$\phi = \phi_1 \pm U_{GCI} \quad (4.27)$$

4.1.6 Estimador delta

De acordo com o estimador delta (U_Δ) usado por Demirdzic *et al.* (1992), a incerteza de uma solução numérica (ϕ_1) é calculada com

$$U_\Delta(\phi_1) = |\phi_1 - \phi_2| \quad (4.28)$$

A incerteza calculada com a Eq. (4.28) representa uma banda ou um intervalo em torno da solução numérica (ϕ_1).

O cálculo da incerteza segundo o estimador delta também usa duas soluções numéricas (ϕ_1 e ϕ_2) obtidas em duas malhas diferentes (h_1 e h_2), da mesma forma que o estimador de Richardson. Mas ao contrário deste, U_Δ não leva em conta a razão de refino de malha (q), ou os valores de h_1 e h_2 , nem a ordem assintótica (p_L) do erro de discretização. Conseqüentemente, U_Δ pode ser feito arbitrariamente muito pequeno, quando $q \rightarrow 1$, ou muito grande, quando $q \gg 1$. Uma relação facilmente verificável é que a magnitude do estimador delta (U_Δ) coincide com o resultado do estimador de Richardson (U_{Ri}) para o caso em que $q = 2$ e $p_L = 1$, conforme pode ser visto comparando-se as Eqs. (4.11) e (4.28).

Pode-se demonstrar (Marchi, 2001) que

$$\frac{U_{\Delta}}{|E|} \rightarrow q^{p_L} - 1 \quad \text{para } h \rightarrow 0 \quad (4.29)$$

Para $p_L = 1$ e $q = 2$, por exemplo, da Eq. (4.29), $U_{\Delta}/|E| \rightarrow 1$. No caso de $p_L = 2$ e $q = 2$, $U_{\Delta}/|E| \rightarrow 3$. Nestes dois casos, U_{Δ} é confiável. Mas, para $p_L = 2$ e $q = 1,1$, da Eq. (4.29), $U_{\Delta}/|E| \rightarrow 0,21$, isto é, U_{Δ} não é confiável porque subestima a magnitude do erro, representando apenas cerca de $1/5$ do seu valor. A subestimativa pode ser ainda pior quando $p_U < 1$.

4.1.7 Ordem efetiva

A ordem efetiva do erro verdadeiro é definida por (Marchi, 2001)

$$p_E = \frac{\log\left[\frac{E(\phi_2)}{E(\phi_1)}\right]}{\log(q)} \quad (4.30)$$

Conforme a Eq. (4.30), a ordem efetiva (p_E) é função do erro verdadeiro da variável de interesse. Assim, para os problema cuja solução analítica é conhecida, ela pode ser usada para verificar *a posteriori* se, à medida que $h \rightarrow 0$, obtém-se a ordem assintótica (p_L) do erro de discretização, que é um resultado teórico obtido *a priori*.

4.1.8 Procedimento recomendado

Para malhas unidimensionais uniformes e razão de refino constante entre as malhas, recomenda-se o seguinte procedimento para estimar o erro de discretização de cada variável de interesse:

- 1) Para as aproximações numéricas usadas na discretização do modelo matemático do problema, determinar a ordem assintótica (p_L).
- 2) Obter três soluções numéricas (ϕ_1, ϕ_2, ϕ_3) em três malhas diferentes (h_1, h_2 e h_3).
- 3) Com a Eq. (4.16), calcular a razão de refino (q) constante entre as malhas h_1, h_2 e h_3 .
- 4) Com a Eq. (4.18), calcular a ordem aparente (p_U).
- 5) Se $p_U > p_L$, calcular a incerteza (U_{GCI}) da solução numérica (ϕ_1), na malha fina (h_1), com a Eq. (4.24) e $F_s = 3$.
- 6) Se $p_U < p_L$, calcular a incerteza (U_{GCI}) da solução numérica (ϕ_1), na malha fina (h_1), com a Eq. (4.25) e $F_s = 3$.
- 7) Expressar a solução numérica e sua incerteza através da Eq. (4.27).

4.2 ERROS DE ITERAÇÃO

A discretização de uma equação diferencial através de um método numérico resulta em

$$[A][\lambda] = [B] \quad (4.31)$$

onde $[A]$ é a matriz dos coeficientes, $[\lambda]$ é o vetor incógnita e $[B]$ é o vetor do termo fonte. No caso do problema abordado na seção 3.1,

$$A_{k,j-1} \lambda_{j-1} + A_{k,j} \lambda_j + A_{k,j+1} \lambda_{j+1} = B_k \quad (\text{para } j = k) \quad (4.32)$$

onde j e k representam cada nó da malha.

O erro de iteração (E_n) da solução numérica (ϕ_n) de uma variável de interesse, na iteração n , é definido por (Fergiger & Peric, 1999)

$$E(\phi_n) = \phi_{n \rightarrow \infty} - \phi_n \quad (4.33)$$

onde $\phi_{n \rightarrow \infty}$ representa a solução exata do sistema (4.31), isto é, sem erros de arredondamento e de programação. A variável ϕ pode representar λ no sistema (4.31) ou qualquer outra variável obtida a partir de λ . Quando o sistema (4.31) tem coeficientes $[A]$ constantes e se usa um método direto para resolvê-lo, como por exemplo eliminação de Gauss, a solução de λ não contém erros de iteração. Algumas causas de erros de iteração são:

- emprego de métodos iterativos para resolver o sistema (4.31);
- problemas não-lineares, isto é, quando os coeficientes $[A]$ dependem de λ ;
- modelos matemáticos constituídos por mais de uma equação, sendo cada uma resolvida separadamente; e
- utilização de métodos *multigrid* para resolver o sistema (4.31).

Três características fundamentais dos erros de iteração:

- 1) em geral eles diminuem com o aumento do número de iterações;
- 2) para $n \rightarrow \infty$, $E(\phi_n) \rightarrow 0$; e
- 3) para convergência monotônica e $n \rightarrow \infty$,

$$E(\phi_n) = \frac{C}{10^{n p_L}} \quad (4.34)$$

onde

C = coeficiente que independe de n

p_L = ordem assintótica do erro de ϕ

O valor da ordem assintótica (p_L) é sempre um número real maior do que zero e representa a inclinação da curva do erro de iteração num gráfico $\log(|E_n|)$ versus n para $n \rightarrow \infty$.

Essas características podem ser ilustradas pelo exemplo a seguir. Considere a seguinte equação:

$$(\lambda - 2)(\lambda - 3) = 0 \quad \text{ou} \quad \lambda^2 - 5\lambda + 6 = 0 \quad (4.35)$$

A solução numérica de uma das raízes desta equação pode ser obtida pelo método da iteração linear (Barroso *et al.*, 1987); por exemplo, através de

$$\lambda_n = \frac{\lambda_{n-1}^2 + 6}{5} \quad (4.36)$$

Para $\lambda_0 = 0$, a curva de erro obtida é mostrada na Fig. 4.5. A solução exata neste caso é 2 e $p_L = \log(5/4) \cong 0,0969$.

Também há dois tipos de estimativas do erro de iteração: *a priori* e *a posteriori* das soluções numéricas.

4.2.1 Estimativas *a Priori*

As estimativas *a priori* de $E(\phi_n)$ se baseiam na Eq. (4.34). A partir dela temos

$$\frac{E(\phi_{n+1})}{E(\phi_n)} = \frac{1}{10^{p_L}} \quad (4.37)$$

Isto é, quanto maior a ordem assintótica, mais rapidamente o erro de iteração é reduzido com o aumento do número de iterações. Mas, em geral, p_L só pode ser obtido *a priori* para problemas muito simples. Para a maioria das aplicações, p_L é desconhecido. Normalmente esta dificuldade só é superada com o método exposto na próxima seção.

4.2.2 Estimativas *a Posteriori*

Dois métodos para estimar a magnitude do erro iterativo são apresentados por Ferziger & Peric (1999) e Roy & Blotner (2001). Eles são semelhantes ao método apresentado a seguir.

A partir da Eq.(4.31), vamos admitir que

$$U(\phi_n) = \frac{K}{10^{n p_U}} \quad (4.38)$$

onde

K = coeficiente que é admitido independe de n

n = número da iteração

p_U = ordem aparente da incerteza

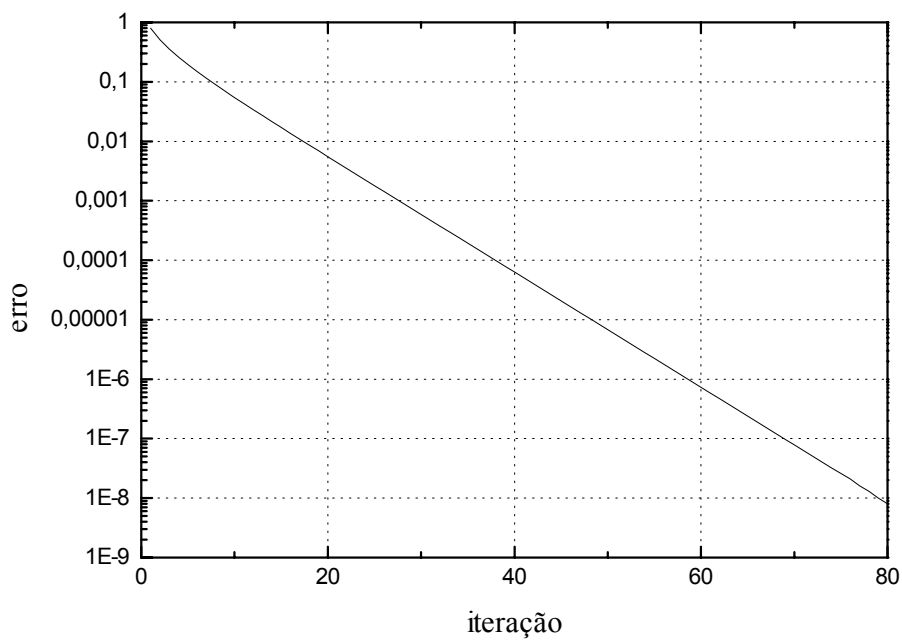


Figura 4.5 Comportamento monotônico do erro de iteração.

Com a Eq. (4.2) em (4.38),

$$\phi_\infty - \phi_n = \frac{K}{10^{n p_U}} \quad (4.39)$$

onde, aqui, ϕ_∞ representa uma estimativa da solução exata do sistema (4.31). Escrevendo-se a Eq. (4.39) para soluções numéricas obtidas em três iterações sucessivas ($n-2$, $n-1$ e n), tem-se

$$\phi_\infty - \phi_{n-2} = K 10^{-(n-2) p_U} \quad (4.40)$$

$$\phi_\infty - \phi_{n-1} = K 10^{-(n-1) p_U} \quad (4.41)$$

$$\phi_\infty - \phi_n = K 10^{-n p_U} \quad (4.42)$$

Nestas três equações, ϕ_{n-2} , ϕ_{n-1} , ϕ_n e n são conhecidos. As incógnitas são ϕ_∞ , K e p_U . Com a solução deste sistema, obtém-se

$$\phi_\infty = \phi_n + \frac{(\phi_n - \phi_{n-1})}{(10^{p_U} - 1)} \quad (4.43)$$

onde

$$p_U = \log \left(\frac{\phi_{n-1} - \phi_{n-2}}{\phi_n - \phi_{n-1}} \right) \quad (4.44)$$

A substituição da Eq. (4.43) em (4.2) finalmente fornece a incerteza (U) da solução numérica na iteração n , dada por

$$U(\phi_n) = \frac{(\phi_n - \phi_{n-1})}{(10^{p_U} - 1)} \quad (4.45)$$

Esta estimativa do erro de iteração só deve ser empregada para $p_U > 0$. Note que o erro de truncamento é eliminado na Eq. (4.45) devido à subtração das soluções numéricas na mesma malha.

Pode-se demonstrar que

$$\frac{U(\phi_n)}{E(\phi_n)} \rightarrow 1 \quad \text{para} \quad n \rightarrow \infty \quad (4.46)$$

4.2.3 Critério de convergência baseado no resíduo

Para o sistema dado na Eq. (4.31), o resíduo do sistema de equações algébricas é definido genericamente por (Ferziger e Peric, 1999)

$$R^n = B - A\lambda^n \quad (4.47)$$

onde λ^n é a solução da incógnita na iteração n . Para o caso particular da Eq. (3.13), a definição da Eq. (4.47) resulta em

$$R_P^n = (a_W)_P T_W^n + (a_E)_P T_E^n + b_P - (a_P)_P T_P^n \quad (4.48)$$

O critério para interromper o processo iterativo com base no resíduo das equações é

$$\text{Se } \frac{L^n}{L^0} \leq Tol \rightarrow \text{parar} \quad (4.49)$$

onde

L^n = norma dos resíduos na iteração n

L^0 = norma dos resíduos na iteração 0, isto é, antes de iniciar o processo iterativo;

neste caso, os resíduos são calculados com base na estimativa inicial da solução.

Tol = tolerância admitida para interromper o processo iterativo, em geral, 10^{-4} ou 10^{-5}

São usadas, geralmente, as normas (Kreyszig, 1999) L_1 , L_2 e L_∞ . No caso da primeira,

$$L_1^n = \sum_{P=1}^N |R_P^n| \quad (4.50)$$

4.2.4 Critério de convergência baseado na variação da incógnita

O critério para interromper o processo iterativo com base na variação da incógnita é (Ferziger e Peric, 1999)

$$\text{Se } L^n \leq Tol \rightarrow \text{parar} \quad (4.51)$$

onde

L^n = norma da variação da incógnita na iteração n

Tol = tolerância admitida para interromper o processo iterativo, em geral, 10^{-3} a 10^{-8}

Aqui, também são usadas, geralmente, as normas (Kreyszig, 1999) L_1 , L_2 e L_∞ . No caso da primeira,

$$L_1^n = \sum_{p=1}^N |T_p^n - T_p^{n-1}| \quad (4.52)$$

4.2.5 Procedimento recomendado

Para interromper o processo iterativo de solução de cada sistema de equações algébricas, representada pela Eq. (4.31), recomenda-se:

- 1) Com as Eqs. (4.48) e (4.50), calcular L^0 com base na estimativa inicial da solução.
- 2) Para cada iteração, com as Eqs. (4.48) e (4.50), calcular L^n .
- 3) Plotar num gráfico semi-log, semelhante à Fig. 4.5, a razão L^n/L^0 ao longo das iterações até que o valor desta razão atinja o mínimo possível, isto é, o erro de máquina. Em problemas muito simples, o erro de máquina é um valor constante, mas na maioria é uma oscilação em torno de um valor médio que permanece fixo ao longo das iterações.
- 4) Anotar o valor do erro de máquina e apresentá-lo como uma indicação do erro de iteração.

Recomenda-se também empregar sempre precisão dupla, ou maior.

4.3 ERROS DE ARREDONDAMENTO

O erro de arredondamento de um variável de interesse (ϕ) é definido por

$$E(\phi_\pi) = \Phi - \phi_\pi \quad (4.53)$$

onde

Φ = solução analítica exata da variável de interesse

ϕ_π = solução numérica sem erros de discretização, de iteração e de programação

A causa dos erros de arredondamento é a representação finita das variáveis nas computações numéricas, que está ligada à precisão dos números. Esta precisão depende do compilador (*software*) usado para gerar o código computacional e do computador (*hardware*) empregado em sua execução. Os erros de arredondamento provocam perda de precisão dos números, que ocorre basicamente por dois motivos:

- 1) Número de operações nos cálculos: provoca perda de precisão no lado direito dos números. Ocorre por exemplo, na obtenção da solução do sistema (4.31).
- 2) Cancelamento subtrativo nos cálculos: provoca perda de precisão no lado esquerdo dos números. Ocorre, por exemplo, no cálculo de erros e incertezas.

Um modelo grosseiro que permite estimar a perda de precisão (Δp) devido ao número de operações é

$$\Delta p_{\max} = O(N) \text{ a } O(N^2) \quad (4.54)$$

onde N é o número máximo de elementos nas dimensões do problema, e $O(N)$ indica a ordem de grandeza da variável entre parênteses. Alguns exemplos são apresentados a seguir.

Exemplo 1: $N = 100 = 10^2 \quad \rightarrow \quad O(N) = 2$
 $N^2 = (100)^2 = 10^4 \quad \rightarrow \quad O(N^2) = 4$

Portanto: $\Delta p_{\max} = 2$ a 4 algarismos

Se $\phi = 1,2345678 \times 10^3$, 2, 3 ou 4 algarismos do lado direito podem estar afetados por erros de arredondamento.

Exemplo 2: $N = 65536 = 6,536 \times 10^4 \quad \rightarrow \quad O(N) = 4$
 $N^2 = (65536)^2 \cong 4,29 \times 10^9 \quad \rightarrow \quad O(N^2) = 9$

Portanto: $\Delta p_{\max} = 4$ a 9 algarismos

Se $\phi = 1,2345678 \times 10^3$, é possível que todos os algarismos deste número estejam afetados por erros de arredondamento e, assim, seu valor não tem utilidade.

Exemplo 3: Malha 2D = 64 x 1024
 $N_{\max} = 1024 = 1,24 \times 10^3 \quad \rightarrow \quad O(N) = 3$
 $(N^2)_{\max} = (1024)^2 \cong 1,049 \times 10^6 \rightarrow \quad O(N^2) = 6$

Portanto: $\Delta p_{\max} = 3$ a 6 algarismos

O efeito do número de operações sobre os erros de arredondamento da solução numérica de uma variável de interesse é mostrado na Fig. 4.6. Grosseiramente, quanto menor o tamanho dos elementos da malha (h) ou maior o número de nós (N), maior é o valor do erro de arredondamento.

Um exemplo de cancelamento subtrativo é dado a seguir. Duas soluções numéricas obtidas são $\phi_1 = 1,2345678 \times 10^3$ e $\phi_2 = 1,2344567 \times 10^3$. Ao se calcular a incerteza (U) com as Eqs. (4.11), (4.19), (4.24), (4.25) ou (4.45), no numerador tem-se a diferença entre estas soluções, isto é,

$$\begin{array}{r} \phi_1 = 1,2345678 \times 10^3 \\ - \phi_2 = 1,2344567 \times 10^3 \\ \hline \phi_1 - \phi_2 = 1,111 \times 10^{-1} \end{array}$$

Portanto, neste caso, o cancelamento subtrativo provocou a perda de 4 algarismos do lado esquerdo do número.

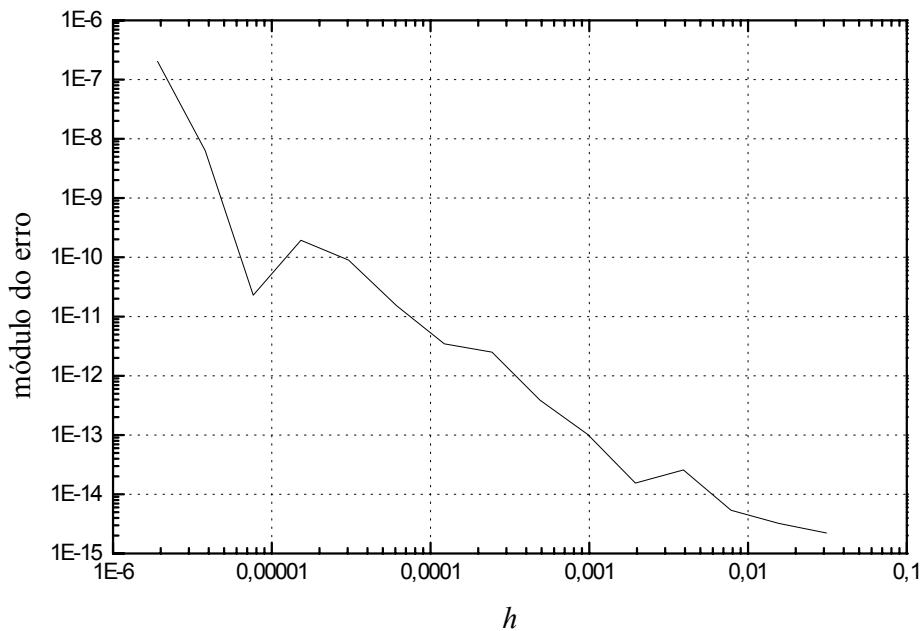


Figura 4.6 Exemplo do comportamento geral dos erros de arredondamento.

4.4 ERROS DE PROGRAMAÇÃO

O erro de programação de uma variável de interesse (ϕ) pode ser definido por

$$E(\phi_p) = \Phi - \phi_p \quad (4.55)$$

onde

Φ = solução analítica exata da variável de interesse

ϕ_p = solução numérica sem erros de discretização, de iteração e de programação

Os erros de programação podem ser causados (Roache, 1998):

- pelo uso incorreto de um modelo numérico na aproximação do modelo matemático;
- pela implementação incorreta do modelo numérico no programa computacional;
- pelo uso incorreto do programa computacional durante a obtenção da solução numérica; e
- por qualquer outra eventual fonte de erro.

Nos livros de Ferziger & Peric (1999), Maliska (2004) e Roache (1998b) existem discussões sobre erros de programação, de como evitá-los e detectá-los. Sugestões gerais são:

- 1) Inicialmente, implemente um programa enxuto, específico, depois tente generalizá-lo.
- 2) Implemente o programa em módulos. Isso facilita a detecção e eliminação dos erros de programação. Use, por exemplo, um módulo para calcular coeficientes e termos fontes e outro para resolver o sistema de equações.
- 3) Teste o *solver* para um sistema de equações simples com resultados exatos conhecidos.
- 4) Usando uma malha pequena, verifique se a solução converge. Isto é, se o erro de iteração atinge o nível do erro de máquina ou erro de arredondamento.
- 5) Resolva um problema “fabricado” para verificar se, para $h \rightarrow 0$, o erro de discretização $\rightarrow 0$, e $p_E, p_U, \rightarrow p_L$.

Na Fig. 4.7 representa-se o erro de programação detectado no tipo de teste mencionado no item 5 acima.

4.4.1 Método das soluções fabricadas

Shih (1985) e Roache (2002) apresentam procedimentos para fabricar ou conceber soluções de problemas com o intuito de verificar erros de programação, em particular, e erros numéricos, em geral. O objetivo é encontrar ou fabricar uma solução analítica exata para um problema similar ao

que se deseja resolver numericamente. O método é útil quando não se conhece a solução analítica do problema original.

Procedimento:

- 1) Propor uma solução para o problema original.
- 2) Calcular as derivadas envolvidas no problema original com a solução proposta.
- 3) Substituir a solução e suas derivadas no problema original. A equação original não será satisfeita, isto é, não haverá igualdade entre os membros à esquerda e à direita.
- 4) Calcular o resíduo que resulta do passo 3, com ele no membro da direita da equação.
- 5) Substituir o resíduo no membro da direita da equação original, multiplicado por uma constante C , para que ela passe a ser satisfeita para a solução proposta (fabricada).

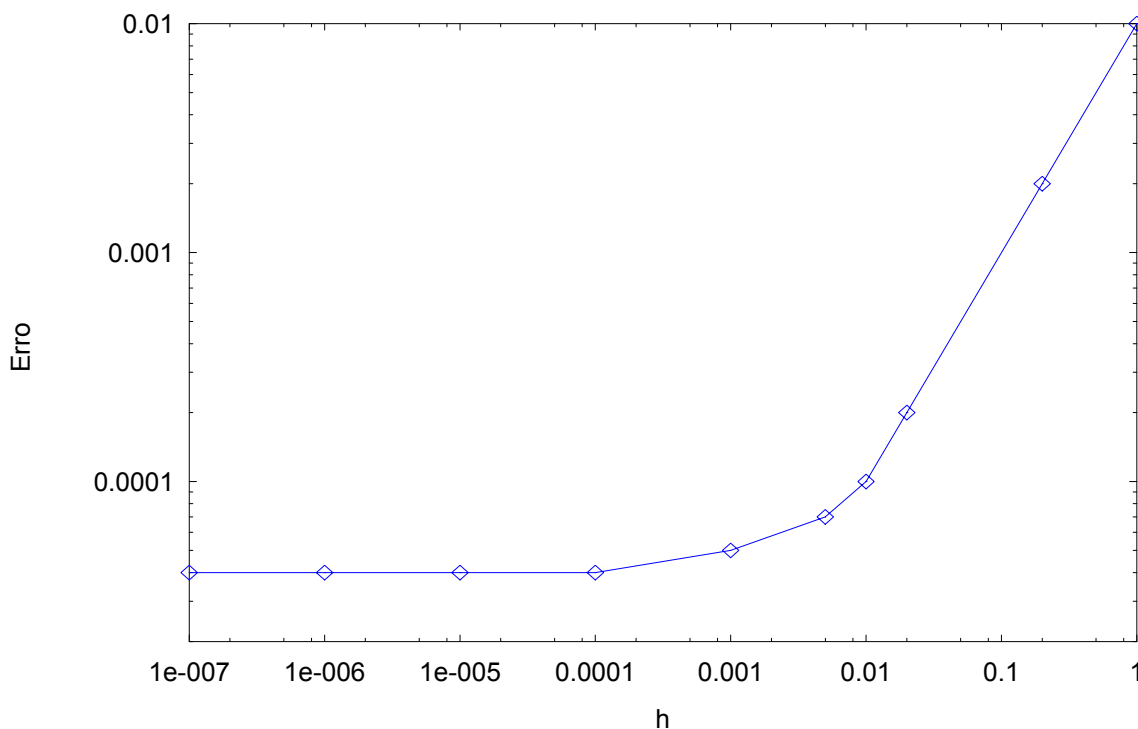


Figura 4.7 Exemplo de erro de programação.

Exemplo seguindo os cinco passos descritos acima. Equação original:

$$\frac{d^2\Lambda}{dX^2} + X \frac{d\Lambda}{dX} + \Lambda^2 = X \quad (4.56)$$

- 1) Propor uma solução para o problema original:

$$\Lambda = X^2 \quad (4.57)$$

2) Calcular as derivadas envolvidas no problema original com a solução proposta:

$$\frac{d\Lambda}{dX} = 2X \quad \frac{d^2\Lambda}{dX^2} = 2 \quad (4.58)$$

3) Substituir a solução e suas derivadas no problema original. A equação original não será satisfeita, isto é, não haverá igualdade entre os membros à esquerda e à direita:

$$2 + X 2X + X^4 = X \quad (4.59)$$

4) Calcular o resíduo que resulta do passo 3, com ele no membro da direita da equação:

$$R = X^4 + 2X^2 - X + 2 \quad (4.60)$$

5) Substituir o resíduo no membro da direita da equação original, multiplicado por uma constante C , para que ela passe a ser satisfeita para a solução proposta (fabricada):

$$\frac{d^2\Lambda}{dX^2} + X \frac{d\Lambda}{dX} + \Lambda^2 = X + RC \quad (4.61)$$

A Eq. (4.61) se constitui no problema fabricado. O coeficiente C é usado para ativar ($C = 1$) ou desativar ($C = 0$) a solução fabricada na solução numérica. Desta forma, verificando-se o erro numérico para a Eq. (4.61), espera-se que automaticamente o erro numérico também esteja verificado para a Eq. (4.56), a equação original.

4.5 EXERCÍCIOS

Exercício 4.1 (seção 4.1)

Através de uma estimativa de erro *a priori*, determine as ordens verdadeiras e assintóticas do erro de truncamento das seguintes equações diferenciais discretizadas:

- 1) Eq. (3.11), da seção 3.1
- 2) Eq. (3.54), da seção 3.4
- 3) Eq. (3.72), da seção 3.5

Exercício 4.2 (seção 4.1)

Deduza as Eqs. (4.9), (4.11), (4.17), (4.18) e (4.19).

Exercício 4.3 (seção 4.1)

Execute o programa POISSON, que resolve o problema da seção 3.1, disponível em

<ftp.demec.ufpr.br/Disciplinas/TM797>

Use os seguintes dados fixos: $S_0 = -\frac{1}{2}$; $S_1 = -\frac{3}{2}$; $S_2 = -1$; $T_0 = 0$ e $T_L = k = L = 1$.

Variáveis de interesse: $T(\frac{1}{2})$, T_m , q_0'' e q_L'' .

Resultados a apresentar para cada variável de interesse:

- 1) Solução numérica com $N = 11, 21$ e 41 nós.
- 2) Valor da ordem assintótica (p_L).
- 3) Valor da razão de refino (q).
- 4) Valor da ordem aparente (p_U).
- 5) Erro verdadeiro da solução numérica da malha mais fina.
- 6) Incerteza (U_{GCI}) da solução numérica da malha mais fina, para $F_s = 3$, de acordo com o procedimento da seção 4.1.8.
- 7) Expressar a solução numérica e sua incerteza através da Eq. (4.27).
- 8) Valor da razão entre incerteza e erro.

Exercício 4.4 (seção 4.2)

Considere o programa computacional que foi implementado para resolver o exercício 3.2, 3.3, 3.4 ou 3.5 com o método TDMA. Altere-o para resolver o problema com o método Gauss-Seidel. Siga o procedimento recomendado na seção 4.2.5. Apresente um gráfico semi-log, semelhante à Fig. 4.5, com a razão L^n/L^0 ao longo das iterações.

Exercício 4.5 (seção 4.3)

Execute o programa POISSON, que resolve o problema da seção 3.1, disponível em

<ftp.demec.ufpr.br/Disciplinas/TM797>

Use os seguintes dados fixos: $S_0 = S_1 = S_2 = T_0 = 0$ e $T_L = k = L = 1$.

Obtenha a solução numérica de $T(\frac{1}{2})$ com $N = 11, 101, 1001, 10001$ e 100001 nós.

Apresente um gráfico log-log, semelhante à Fig. 4.6, com o módulo do erro verdadeiro da solução numérica de $T(\frac{1}{2})$ versus h .