

RELATÓRIO DE PESQUISA

MÉTODOS DE MINIMIZAÇÃO PARA ANÁLISE NÃO-LINEAR DE ESTRUTURAS

José Milton de Araújo
Pablo Gaston Bignon

RP-110/93

Relatório de Pesquisa RP-110/93 desenvolvido no Curso de Pós-Graduação em Engenharia Civil da UFRGS, como parte dos créditos do curso de doutorado do primeiro autor.

Dados para referência:

Araújo, J. M.; Bignon, P. G. *Métodos de Minimização para Análise Não-Linear de Estruturas*. Relatório de Pesquisa RP-110/93, Curso de Pós-Graduação em Engenharia Civil, UFRGS, Porto Alegre, 1993.

Disponível em: <http://www.editoradunas.com.br/dunas/RP110.pdf>

Porto Alegre, 1993

RELATÓRIO DE PESQUISA

MÉTODOS DE MINIMIZAÇÃO PARA ANÁLISE

NÃO-LINEAR DE ESTRUTURAS

RP-110/93

José Milton de Araújo

Pablo Gaston Bignon

Porto Alegre - 

RESUMO

O objetivo deste trabalho é apresentar os principais métodos de minimização empregados em análise não-linear de estruturas. Iniciando com uma formulação consistente com a Programação Matemática, os fundamentos teóricos do processo de minimização vão sendo apresentados até se chegar ao problema particular de análise estrutural. Um problema estrutural não-linear é, então, analisado e a eficiência computacional dos métodos é testada. Em anexo são apresentadas as listagens dos programas em linguagem FORTRAN.

SUMÁRIO

1 - O PROBLEMA DA PROGRAMAÇÃO NÃO-LINEAR.....	1
1.1 - Introdução	1
1.2 - Programação quadrática	4
1.3 - Tipos de restrições	4
1.4 - O gradiente da função objetiva	6
1.5 - Aproximação de funções	7
2 - A FUNÇÃO ENERGIA POTENCIAL	9
2.1 - Introdução	9
2.2 - O gradiente e a Hessiana	10
3 - MÉTODOS DE MINIMIZAÇÃO IRRESTRITA	13
3.1 - Direções conjugadas	13
3.2 - Pesquisa linear	16
3.3 - Método de Powell	22
3.4 - O método descendente por degraus	23
3.5 - O método dos gradientes conjugados	24
3.6 - Os métodos de Newton-Raphson	25
3.7 - Métodos quase-Newton	28
3.7.1 - Introdução	28
3.7.2 - Método de Broyden	30
3.7.3 - Método de Davidon-Fletcher-Powell ..	30
3.7.4 - Algoritmos de Pearson	31
3.7.5 - Método BFGS	31
3.8 - O método Secante-Newton	32
3.9 - Um método de minimização bidirecional	35

4 - APLICAÇÃO À ANÁLISE ESTRUTURAL	38
4.1 - Introdução	38
4.2 - Resumo dos métodos de minimização	39
4.3 - Implementação computacional dos métodos quase- -Newton	43
4.4 - Exemplos numéricos	61
BIBLIOGRAFIA	68
ANEXO: Listagens dos programas em FORTRAN	70

1 - O PROBLEMA DA PROGRAMAÇÃO NÃO-LINEAR

1.1 - Introdução

A programação não-linear trata da otimização de funções não-lineares submetidas a um conjunto de restrições lineares ou não-lineares. Otimizar uma função significa encontrar o "melhor" valor das variáveis independentes que satisfaz todas as restrições impostas. O melhor valor, ou valor ótimo, corresponde a um ponto estacionário para a função podendo ser um ponto de máximo ou de mínimo. A função a ser otimizada é denominada função objetiva.

Matematicamente, o problema pode ser apresentado na forma

minimizar: $f(\underline{U})$ com $\underline{U} \in E^n$

sujeito a:

$$h_j(\underline{U}) = 0, \quad j=1, \dots, m$$

$$\text{e } t_j(\underline{U}) \geq 0, \quad j=m+1, \dots, p.$$

No problema acima $f(\underline{U})$ representa a função objetiva e \underline{U} é o vetor de incógnitas definido no espaço Euclidiano n -dimensional E^n . As restrições podem ser de igualdade, $h_j(\underline{U}) = 0$, e de desigualdade, $t_j(\underline{U}) \geq 0$.

A função objetiva $f(\underline{U})$ é não-linear e as restrições podem ser lineares e/ou não-lineares.

Como exemplo, seja o problema

$$\text{minimizar: } f(\underline{U}) = u_1^2 + u_2^2 + 2u_2 \quad (1.1)$$

$$\text{sujeito a: } h_1(\underline{U}) = u_1^2 + u_2^2 - 1 = 0 \quad (1.2)$$

$$t_2(\underline{U}) = u_1 + 2u_2 - 0,5 \geq 0 \quad (1.3)$$

$$t_3(\underline{U}) = u_1 \geq 0 \quad (1.4)$$

$$t_4(\underline{U}) = u_2 \geq 0 \quad (1.5)$$

$$\text{sendo } \underline{U} = \begin{Bmatrix} u_1 \\ u_2 \end{Bmatrix}^T$$

Na figura 1.1 indicam-se os contornos de $f(\underline{U})$ e as restrições.

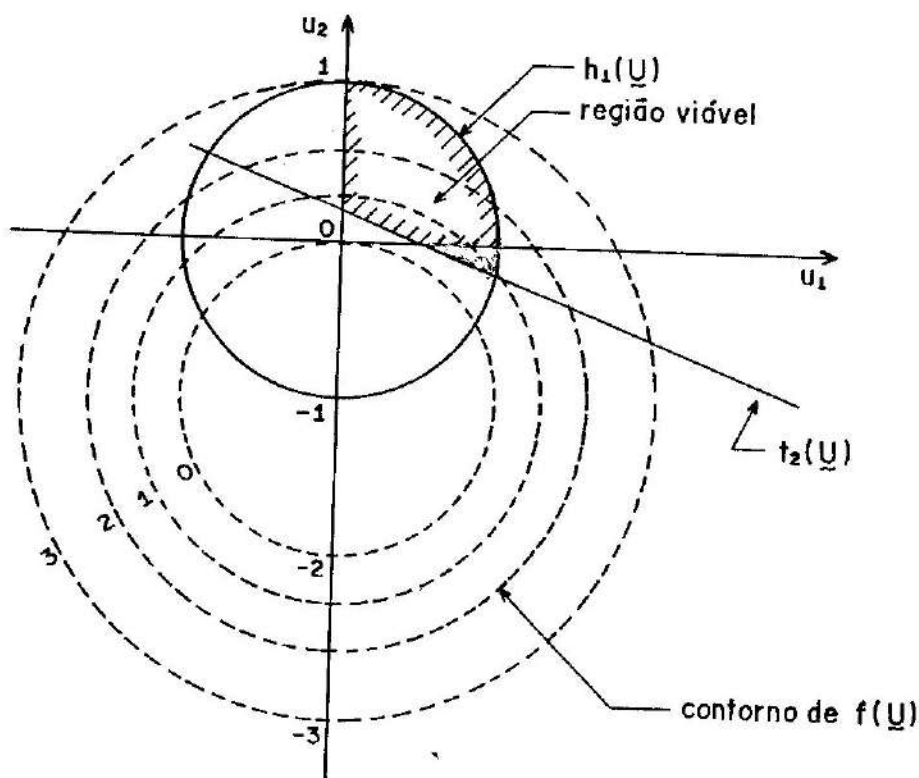


Figura 1.1 - Representação geométrica de funções

A região viável é definida como o conjunto dos pontos que atendem a todas as restrições. Os contornos da função objetiva são obtidos igualando-se $f(\underline{U})$ a um valor constante. O vetor \underline{U}^* que minimiza $f(\underline{U})$ é denominado ponto ótimo e deve situar-se na região viável.

Áreas típicas de aplicação da programação não-linear são: controle de qualidade, controle de produção, dimensionamento e análise estrutural, etc.

A solução do problema de otimização é, em geral, obtida através de processos iterativos onde em cada iteração procura-se o mínimo da função objetiva. No caso da programação não-linear não existe um método geral de otimização como existe o método simplex para funções lineares. Assim, os algoritmos são aplicáveis a problemas particulares para os quais se constata sua eficiência.

Os diversos métodos de otimização podem ser classificados como segue:

1 - Métodos analíticos

Esses métodos utilizam técnicas clássicas do cálculo diferencial e do cálculo variacional. O extremo da função $f(\underline{U})$ é encontrado fazendo com que as derivadas de $f(\underline{U})$ com relação a \underline{U} se anulem. Quando são impostas restrições ao problema utilizam-se técnicas como multiplicadores de Lagrange e variação de restrições. O problema deve ser definido em termos matemáticos para ser possível empregar o método. Para grandes problemas o método torna-se impraticável.

2 - Métodos numéricos

Esses métodos utilizam informações anteriores para gerar uma solução melhor através de um processo iterativo. Somente esses métodos serão discutidos neste trabalho.

3 - Métodos gráficos

Desenha-se a função e determina-se o ponto ótimo graficamente. Na prática o método é restrito a problemas com duas variáveis independentes.

4 - Métodos experimentais

O mínimo é obtido por tentativas sem uma sequência lógica de iterações.

1.2 - Programação quadrática

Se a função objetiva é quadrática e as restrições são lineares tem-se um problema de programação quadrática. Este problema é importante porque funções gerais podem ser aproximadas por funções quadráticas próximo ao mínimo.

O problema da programação quadrática é apresentado na forma

$$\text{minimizar: } f(\underline{U}) = a_0 + \underline{C}^T \underline{U} + \underline{U}^T \underline{Q} \underline{U} \quad (1.6)$$

$$\text{sujeito a: } \underline{A} \underline{U} \geq \underline{b} \quad (1.7)$$

$$\underline{\bar{A}} \underline{U} = \underline{\bar{b}} \quad (1.8)$$

onde a_0 é uma constante e \underline{Q} é uma matriz simétrica positivo definida. As matrizes \underline{A} e $\underline{\bar{A}}$ são quadradas ($n \times n$) e \underline{b} , $\underline{\bar{b}}$ e \underline{C} são vetores ($n \times 1$).

Se a matriz \underline{Q} é nula, a equação (1.6) representa o problema de programação linear.

1.3 - Tipos de restrições

Em um problema real de otimização diversos tipos de

restrições devem ser impostas às variáveis independentes \underline{U} . Um primeiro grupo de restrições incorpora limites às variáveis que asseguram sua realidade física ou compatibilidade.

Num segundo grupo as variáveis são conectadas por leis físicas tais como conservação da massa ou da energia, equação de equilíbrio, etc. Estas restrições consistem de uma relação funcional.

Relações empíricas e hipóteses simplificadoras constituem, também, restrições para as variáveis.

Como exemplo, seja a análise de vigas pelo método dos elementos finitos no modelo de deslocamentos. As variáveis independentes são os deslocamentos nodais \underline{U} da estrutura. A função objetiva a ser minimizada é a energia potencial total da estrutura. As restrições impostas são:

- a) Os deslocamentos nodais \underline{U} devem ser tais que preservem a condição de moderadas rotações.
- b) Em alguns pontos nodais deve-se ter $U_i = \bar{U}_i$ onde \bar{U}_i é um valor prescrito para o deslocamento.
- c) Equações constitutivas não-lineares relacionam tensões a deformações que, por sua vez, são obtidas dos deslocamentos por diferenciação.
- d) A hipótese das seções planas é introduzida para simplificar o problema resultando que

$$\varepsilon = \varepsilon_0 - z\chi \quad (1.9)$$

onde ε_0 e χ são a deformação axial e a curvatura, respectivamente, e ε é a deformação longitudinal em uma fibra genérica a uma distância z do eixo. A equação (1.9) impõe uma restrição aos deslocamentos em uma seção transversal da viga.

- e) Uma relação funcional é empregada para definir a energia potencial em função dos deslocamentos nodais.

Um problema de otimização no qual não são impostas

restrições às variáveis é conhecido como um problema de otimização irrestrita. Apesar de no caso real sempre existirem restrições, a classe de problemas de otimização irrestrita é importante porque os algoritmos de otimização irrestrita podem ser usados para a otimização com restrições.

No problema de análise estrutural as restrições são implícitas e são introduzidas automaticamente no processo de solução. Assim, no problema estrutural, pode-se trabalhar como em otimização irrestrita.

1.4 - O gradiente da função objetiva

Se a função objetiva é contínua e diferenciável o gradiente de $f(\underline{U})$ é um vetor de primeiras derivadas parciais de $f(\underline{U})$ com relação a \underline{U} em um ponto dado. Por exemplo, no ponto \underline{U}_i , vem

$$\nabla f(\underline{U}_i) = \underline{g}_i = \left\{ \begin{array}{c} \frac{\partial f(\underline{U}_i)}{\partial u_1} \\ \vdots \\ \frac{\partial f(\underline{U}_i)}{\partial u_n} \end{array} \right\} \quad (1.10)$$

O gradiente aponta na direção de maior incremento da função e é ortogonal ao contorno de $f(\underline{U}_i)$ que passa por \underline{U}_i . O negativo do gradiente aponta na direção de maior decréscimo da função.

Um vetor \underline{V} ortogonal a \underline{g}_i é tangente à superfície em \underline{U}_i e $\underline{V}^T \underline{g}_i = 0$.

1.5 - Aproximação de funções

Alguns dos procedimentos de programação não-linear requerem aproximações lineares ou quadráticas de $f(\underline{U})$, $h_j(\underline{U})$ e $t_j(\underline{U})$.

Uma aproximação linear, ou de primeira ordem, de $f(\underline{U})$ pode ser dada por série de Taylor em torno de \underline{U}_i como

$$f(\underline{U}) \cong f(\underline{U}_i) + \underline{g}_i^T (\underline{U} - \underline{U}_i) \quad (1.11)$$

onde \underline{g}_i é o gradiente de $f(\underline{U}_i)$ dado por (1.10).

Uma aproximação quadrática é

$$f(\underline{U}) \cong f(\underline{U}_i) + \underline{g}_i^T (\underline{U} - \underline{U}_i) + \frac{1}{2} (\underline{U} - \underline{U}_i)^T \underline{H}_i (\underline{U} - \underline{U}_i) \quad (1.12)$$

onde $\underline{H}_i = \underline{H}(\underline{U}_i)$ é a matriz Hessiana de $f(\underline{U}_i)$ dada por

$$\underline{H}_i = \begin{bmatrix} \frac{\partial^2 f(\underline{U}_i)}{\partial u_1^2} & \dots & \frac{\partial^2 f(\underline{U}_i)}{\partial u_1 \partial u_n} \\ \vdots & & \vdots \\ \frac{\partial^2 f(\underline{U}_i)}{\partial u_n \partial u_1} & \dots & \frac{\partial^2 f(\underline{U}_i)}{\partial u_n^2} \end{bmatrix} \quad (1.12)$$

Um ponto \underline{U}^* é um ponto de mínimo para $f(\underline{U})$ se

$$f(\underline{U}^*) < f(\underline{U}) \quad \forall \underline{U} \in E^n \quad (1.13)$$

Considerando a equação (1.12) vem que

$$f(\underline{U}) - f(\underline{U}^*) = \underline{g}^T(\underline{U} - \underline{U}^*) + \frac{1}{2}(\underline{U} - \underline{U}^*)^T \underline{H}(\underline{U}^*)(\underline{U} - \underline{U}^*) > 0 \quad (1.14)$$

Para um ponto arbitrário \underline{U} a equação (1.14) é atendida se

$$\underline{g}^* = \underline{g}(\underline{U}^*) = \underline{0} \quad (1.15)$$

e se

$$(\underline{U} - \underline{U}^*)^T \underline{H}(\underline{U}^*)(\underline{U} - \underline{U}^*) > 0 \quad (1.16)$$

A condição (1.16) é satisfeita se a matriz Hessiana $\underline{H}(\underline{U}^*)$ for positivo definida.

Então a condição necessária para que \underline{U}^* seja um ponto de mínimo é que o gradiente de $f(\underline{U}^*)$ seja nulo. A condição suficiente é que a matriz Hessiana seja positivo definida.

Se a função $f(\underline{U})$ for quadrática a equação (1.12) é exata e o mínimo local será também o mínimo global. Para uma função geral não fica garantido que \underline{U}^* seja um mínimo global.

Os algoritmos que serão apresentados levam ao ponto ótimo local. Na prática pode-se iniciar com diferentes vetores de partida mas, mesmo se somente uma solução for encontrada, não pode ser demonstrado que a solução corresponde ao ótimo global. Felizmente, para os problemas reais a função objetiva é bem comportada apresentando um único extremo.

2 - A FUNÇÃO ENERGIA POTENCIAL

2.1 - Introdução

Na análise estrutural pelo método dos elementos finitos a estrutura é discretizada em pequenos elementos ligados através de pontos nodais. Os deslocamentos nodais \underline{U} são tomados como as incógnitas do problema (modelo em deslocamentos). As cargas distribuídas ao longo dos elementos são transformadas em nodais equivalentes e superpostas às cargas aplicadas diretamente nos nós dando origem ao vetor de cargas \underline{P} .

Considerando um material elástico não-linear a energia potencial total da estrutura é dada por

$$\pi = \phi + V \quad (2.1)$$

onde $\phi = \phi(\underline{U})$ é a energia de deformação e V é o potencial das forças externas dado por

$$V = - \underline{U}^T \underline{P} \quad (2.2)$$

Então

$$\pi = \phi(\underline{U}) - \underline{U}^T \underline{P} \quad (2.3)$$

As não-linearidades física e geométrica são introduzidas em $\phi(\underline{U})$.

2.2 - O gradiente e a Hessiana

Seja um processo iterativo de minimização dado pela fórmula de recorrência

$$\underline{U}_{i+1} = \underline{U}_i + \Delta \underline{U}_i \quad (2.4)$$

onde \underline{U}_i representa os deslocamentos na iteração i e $\Delta \underline{U}_i$ é o incremento a ser dado para a próxima iteração.

Expandindo a energia potencial em série de Taylor até os termos quadráticos vem

$$\pi(\underline{U}_{i+1}) \simeq \pi(\underline{U}_i) + \underline{g}_i^T \Delta \underline{U}_i + \frac{1}{2} \Delta \underline{U}_i^T \underline{H}_i \Delta \underline{U}_i \quad (2.5)$$

sendo $\underline{g}_i = \left[\frac{\partial \pi}{\partial \underline{U}} \right]_{\underline{U}=\underline{U}_i}$ = gradiente de π em \underline{U}_i ;

$\underline{H}_i = \left[\frac{\partial^2 \pi}{\partial \underline{U}^2} \right]_{\underline{U}=\underline{U}_i}$ = matriz Hessiana de π em \underline{U}_i .

A equação (2.3) pode ser escrita na forma

$$\pi = \phi(\underline{U}) - \sum_{j=1}^n U_j P_j \quad (2.6)$$

e a componente j -ésima do gradiente é dada por

$$g_j(\underline{U}_i) = \frac{\partial \phi(\underline{U}_i)}{\partial U_j} - P_j \quad (2.7)$$

Pelo primeiro Teorema de Castigliano tem-se que

$$A_{NLj}(U_i) = \frac{\partial \phi(U_i)}{\partial U_j} \quad (2.8)$$

onde $A_{NLj}(U_i)$ é a j -ésima componente do vetor de ações nodais não-lineares para os deslocamentos U_i .

Introduzindo (2.8) em (2.7) vem

$$g_j(U_i) = A_{NLj}(U_i) - P_j = -\psi_j(U_i) \quad (2.9)$$

onde ψ_j é a j -ésima componente do vetor de forças desequilibradas.

Extendendo a equação (2.9) para todos os graus de liberdade chega-se a

$$g_i = -\psi_i \quad (2.10)$$

onde $\psi_i = P - A_{NL}(U_i)$, ou seja, o gradiente da energia potencial é igual ao vetor de forças desequilibradas com sinal trocado. Como o gradiente aponta na direção de maior crescimento de π o vetor de desequilíbrio aponta na direção de maior decréscimo de π .

Considerando a equação (2.6) pode-se definir o elemento $H_{jk}(U_i)$ da matriz Hessiana como

$$H_{jk}(U_i) = \left[\frac{\partial^2 \phi}{\partial U_j \partial U_k} \right]_{U=U_i} \quad (2.11)$$

e introduzindo (2.8) vem

$$H_{jk}(U_i) = \frac{\partial [A_{NLK}(U_i)]}{\partial U_j} \quad (2.12)$$

Assim, a matriz Hessiana de π é obtida por diferenciação das ações não-lineares em relação aos deslocamentos nodais. Isto, por definição, é a matriz de rigidez tangente da estrutura, ou seja,

$$H_{jk}(\underline{U}_i) = K_{jk}(\underline{U}_i) \quad (2.13)$$

Extendendo para todos os elementos vem

$$H_i = K_i \quad (2.14)$$

onde $K_i = K(\underline{U}_i)$ é a matriz de rigidez tangente para os deslocamentos \underline{U}_i .

Introduzindo (2.10) e (2.14) em (2.5) chega-se a

$$\pi(\underline{U}_{i+1}) \simeq \pi(\underline{U}_i) - \psi_i^T \Delta \underline{U}_i + \frac{1}{2} \Delta \underline{U}_i^T K_i \Delta \underline{U}_i \quad (2.15)$$

Pelo teorema da mínima energia potencial total, a configuração de equilíbrio estável é aquela que torna mínima a energia potencial da estrutura.

O problema de otimização em análise estrutural consiste, então, em minimizar a energia potencial da estrutura. Os diversos métodos se diferenciam pela escolha da direção de pesquisa como será apresentado no capítulo seguinte.

3 - MÉTODOS DE MINIMIZAÇÃO IRRESTRITA

3.1 - Direções conjugadas

Seja o problema de minimização de uma função quadrática dada por

$$f(\underline{U}) = a_0 + \underline{U}^T \underline{C} + \underline{U}^T \underline{Q} \underline{U} \quad (3.1)$$

Se um método de minimização converge ao mínimo de uma função quadrática e se o número de iterações é diretamente relacionado ao número de variáveis n , o método é dito possuir convergência quadrática. Se um método de convergência quadrática for aplicado a uma função geral para a qual a série de Taylor é dominada pelos termos quadráticos próximo ao mínimo, então pode-se esperar que a convergência seja obtida.

Muitos métodos de convergência quadrática são baseados, de uma forma ou de outra, no conceito de direções conjugadas.

No contexto da minimização da função quadrática dada por (3.1) um conjunto de n direções são ditas conjugadas, ou mais precisamente \underline{Q} -conjugadas, se

$$\underline{S}_i^T \underline{Q} \underline{S}_j = 0, \quad \forall \quad i \neq j. \quad (3.2)$$

Um conjunto de direções \underline{Q} -conjugadas tem a seguinte propriedade: "Se a função quadrática $f(\underline{U})$ é minimizada

sequencialmente, uma vez ao longo de cada direção de um conjunto de n direções Q -conjugadas linearmente independentes, o mínimo global de $f(\underline{U})$ será localizado em até n passos, independentemente do ponto de partida."

O teorema acima pode ser demonstrado como se segue.

Seja \underline{U}_1 o vetor de partida e considere-se a relação

$$\underline{U}_2 = \underline{U}_1 + \lambda_1 \underline{S}_1 \quad (3.3)$$

onde \underline{U}_2 é a nova aproximação e λ_1 é um escalar que minimiza $f(\underline{U}_2)$.

Substituindo (3.3) em (3.1) vem

$$f(\underline{U}_2) = a_0 + (\underline{U}_1 + \lambda_1 \underline{S}_1)^T \underline{C} + (\underline{U}_1 + \lambda_1 \underline{S}_1)^T \underline{Q} (\underline{U}_1 + \lambda_1 \underline{S}_1) \quad (3.4)$$

O mínimo de f é obtido por

$$\frac{\partial f(\underline{U}_2)}{\partial \lambda_1} = 0 \quad (3.5)$$

de onde resulta

$$\lambda_1 = - \frac{\underline{S}_1^T (2\underline{Q}\underline{U}_1 + \underline{C})}{2\underline{S}_1^T \underline{Q} \underline{S}_1} \quad (3.6)$$

Tomando a nova aproximação

$$\underline{U}_3 = \underline{U}_2 + \lambda_2 \underline{S}_2 \quad (3.7)$$

e minimizando chega-se a

$$\lambda_2 = - \frac{\underline{S}_2^T (2\underline{Q}\underline{U}_2 + \underline{C})}{2\underline{S}_2^T \underline{Q} \underline{S}_2} \quad (3.8)$$

Substituindo (3.3) em (3.8) e lembrando que $S_{2\sim 1}^T Q S_{2\sim 1} = 0$ resulta

$$\lambda_2 = - \frac{S_{2\sim 1}^T (2QU_{1\sim 1} + C)}{2S_{2\sim 1}^T Q S_{2\sim 1}} \quad (3.9)$$

O vetor $U_{3\sim}$ será dado por

$$U_{3\sim} = U_{1\sim} + \lambda_1 S_{1\sim 1} + \lambda_2 S_{2\sim 2} \quad (3.10)$$

É possível mostrar que o vetor $U_{3\sim}$ obtido nesta forma sequencial corresponde ao mínimo da função no plano contendo $S_{1\sim}$ e $S_{2\sim}$ e passando por $U_{1\sim}$.

Seja Z uma aproximação dada por

$$Z = U_{1\sim} + \alpha_1 S_{1\sim 1} + \alpha_2 S_{2\sim 2} \quad (3.11)$$

Substituindo Z em (3.1) vem

$$f(Z) = a_0 + (U_{1\sim} + \alpha_1 S_{1\sim 1} + \alpha_2 S_{2\sim 2})^T C + (U_{1\sim} + \alpha_1 S_{1\sim 1} + \alpha_2 S_{2\sim 2})^T Q (U_{1\sim} + \alpha_1 S_{1\sim 1} + \alpha_2 S_{2\sim 2}) \quad (3.12)$$

A minimização de $f(Z)$ em relação a α_1 e α_2 leva ao sistema de equações

$$\frac{\partial f(Z)}{\partial \alpha_1} = 0 \quad (3.13)$$

$$\frac{\partial f(Z)}{\partial \alpha_2} = 0$$

Diferenciando e lembrando que $S_{1\sim 2}^T Q S_{1\sim 2} = 0$ vem

$$\begin{cases} 2\alpha_1 \tilde{S}_1^T \tilde{Q} \tilde{S}_1 + 2\tilde{S}_1^T \tilde{Q} \tilde{U}_1 + \tilde{S}_1^T \tilde{C} = 0 \\ 2\alpha_2 \tilde{S}_2^T \tilde{Q} \tilde{S}_2 + 2\tilde{S}_2^T \tilde{Q} \tilde{U}_1 + \tilde{S}_2^T \tilde{C} = 0 \end{cases} \quad (3.14)$$

cuja solução é

$$\alpha_1 = - \frac{\tilde{S}_1^T (2\tilde{Q} \tilde{U}_1 + \tilde{C})}{2\tilde{S}_1^T \tilde{Q} \tilde{S}_1} \quad (3.15)$$

$$\alpha_2 = - \frac{\tilde{S}_2^T (2\tilde{Q} \tilde{U}_1 + \tilde{C})}{2\tilde{S}_2^T \tilde{Q} \tilde{S}_2} \quad (3.16)$$

Conclui-se então que $\alpha_1 = \lambda_1$ e $\alpha_2 = \lambda_2$. Logo $\tilde{Z} = \tilde{U}_3$, ou seja, a minimização sequencial leva ao mesmo resultado que a minimização simultânea.

Generalizando o resultado para j ciclos de minimização sequencial ao longo das direções conjugadas $\tilde{S}_1, \tilde{S}_2, \dots, \tilde{S}_j$ conclui-se que será obtido o mínimo da função no subespaço varrido pelos vetores $\tilde{S}_1, \tilde{S}_2, \dots, \tilde{S}_j$. Então em até n passos o mínimo de $f(\tilde{U})$ será alcançado pois $\tilde{U} \in E^n$.

3.2 - Pesquisa linear

Em um processo iterativo de minimização da função $f(\tilde{U})$ em geral emprega-se a fórmula de recorrência

$$\tilde{U}_{i+1} = \tilde{U}_i + \lambda \tilde{S}_i \quad (3.17)$$

onde \tilde{S}_i é a direção de minimização e λ é um escalar que controla o tamanho do passo.

A forma de se obter \tilde{S}_i dá origem aos diversos métodos

de minimização. O escalar λ pode ser tornado igual a 1 ou pode ser o valor que minimiza $f(\underline{U}_{i+1})$ na direção \underline{S}_i .

A minimização de $f(\underline{U}_{i+1})$ pode ser obtida expandindo a função em série de Taylor até os termos quadráticos como

$$f(\underline{U}_{i+1}) = f(\underline{U}_i) + \lambda \underline{S}_i^T \underline{g}_i + \frac{1}{2} \lambda^2 \underline{S}_i^T \underline{H}_i \underline{S}_i \quad (3.18)$$

onde \underline{g}_i é o gradiente de $f(\underline{U}_i)$ e \underline{H}_i é a matriz Hessiana da função em \underline{U}_i .

O valor ótimo para λ é obtido por

$$\frac{\partial f(\underline{U}_{i+1})}{\partial \lambda} = 0 \quad (3.19)$$

de onde resulta

$$\lambda = - \frac{\underline{S}_i^T \underline{g}_i}{\underline{S}_i^T \underline{H}_i \underline{S}_i} \quad (3.20)$$

A dificuldade em se determinar λ está na definição da matriz Hessiana. Para problemas com muitas variáveis esta avaliação de \underline{H}_i pode levar a um alto custo computacional e a redução do número de iterações que se obtém adotando λ dado por (3.20) pode ser menos eficiente que simplesmente adotar $\lambda = 1$.

Uma alternativa para encontrar o valor ótimo de λ sem a necessidade da montagem da matriz Hessiana consiste no emprego de um processo iterativo de procura denominado pesquisa linear. Este processo baseia-se no teorema seguinte:

- "Se $f(\underline{U})$ é uma função quadrática sua minimização numa direção \underline{S}_i leva a um ponto \underline{U}_{i+1} tal que \underline{g}_{i+1} é ortogonal a \underline{S}_i ."

Demonstração

Se $f(\underline{U})$ é quadrática ela é da forma

$$f(\underline{U}) = a_0 + \underline{U}^T \underline{C} + \frac{1}{2} \underline{U}^T \underline{H} \underline{U} \quad (3.21)$$

onde \underline{H} é uma matriz constante positivo definida (matriz Hessiana).

O gradiente de $f(\underline{U})$ é

$$\underline{g} = \underline{C} + \underline{H} \underline{U} \quad (3.22)$$

e

$$\underline{g}_{i+1} = \underline{C} + \underline{H} \underline{U}_{i+1} \quad (3.23)$$

Admitindo a aproximação (3.17) vem

$$f(\underline{U}_{i+1}) = a_0 + (\underline{U}_i + \lambda \underline{S}_i)^T \underline{C} + \frac{1}{2} (\underline{U}_i + \lambda \underline{S}_i)^T \underline{H} (\underline{U}_i + \lambda \underline{S}_i) \quad (3.24)$$

e minimizando em relação a λ chega-se a

$$\frac{\partial f(\underline{U}_{i+1})}{\partial \lambda} = \underline{S}_i^T \underline{C} + \underline{S}_i^T \underline{H} \underline{U}_i + \lambda \underline{S}_i^T \underline{H} \underline{S}_i = 0 \quad (3.25)$$

De (3.17) tem-se

$$\underline{U}_i = \underline{U}_{i+1} - \lambda \underline{S}_i \quad (3.26)$$

que substituído em (3.25) leva a

$$\underline{S}_i^T (\underline{C} + \underline{H} \underline{U}_{i+1}) = 0 \quad (3.27)$$

ou seja

$$\tilde{s}_i^T \tilde{g}_{i+1} = 0. \quad (3.28)$$

Em vista da propriedade (3.28) pode-se desenvolver o processo de procura. O valor ótimo para o escalar λ é aquele que torna verdadeira a equação (3.28). Como \tilde{g}_{i+1} depende de \tilde{U}_{i+1} e este de λ a equação (3.28) é da forma

$$h(\lambda) = \tilde{s}_i^T \tilde{g}_{i+1}(\lambda) = 0 \quad (3.29)$$

Pode-se, então, empregar o processo da bissecante como se segue.

Inicialmente admite-se $\lambda=0$. Assim, $\tilde{U}_{i+1} = \tilde{U}_i$ e $\tilde{g}_{i+1} = \tilde{g}_i$. Logo, a função $h(\lambda)$ em $\lambda = 0$ vale

$$h_0 = \tilde{s}_i^T \tilde{g}_i \quad (3.30)$$

Admite-se $\lambda=1$ e calcula-se \tilde{U}_{i+1} por (3.17). Com \tilde{U}_{i+1} avalia-se o gradiente de $f(\tilde{U}_{i+1})$ e calcula-se

$$h_1 = \tilde{s}_i^T \tilde{g}_{i+1} \quad (3.31)$$

Na figura 3.1 indica-se esquematicamente a função $h(\lambda)$.

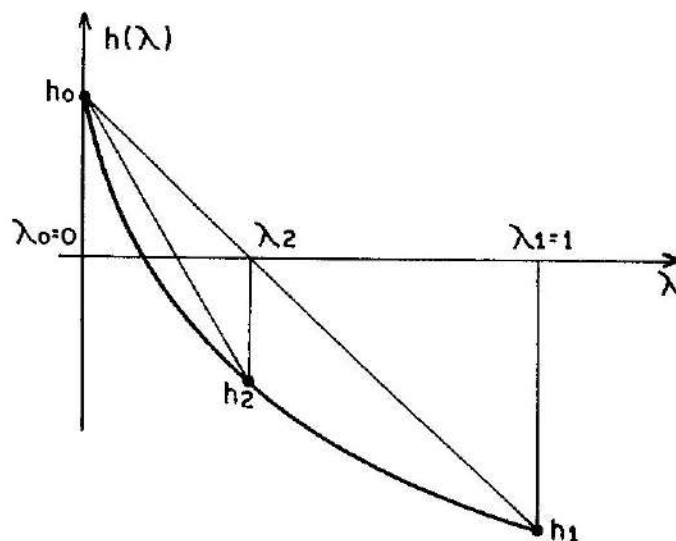


Figura 3.1 - Processo da bissecante

O novo valor $\lambda = \lambda_2$ é determinado pela intersecção da reta que une os pontos $(0, h_0)$ e $(1, h_1)$ com o eixo das abscissas. Com λ_2 calcula-se novo \underline{U}_{i+1} e \underline{g}_{i+1} e segue-se o processo de procura.

A convergência é aceita quando em determinada iteração j for atendida a desigualdade

$$\left| \frac{h_j}{h_1} \right| \leq \text{tolerância} = \epsilon \quad (3.32)$$

Deve-se observar que se o processo converge em j iterações são necessárias j avaliações do gradiente \underline{g}_{i+1} para se definir o conjunto \underline{U}_{i+1} que minimiza $f(\underline{U})$ segundo \underline{S}_i . Este pode ser o grande inconveniente da pesquisa linear. Poderá ser mais eficaz, simplesmente, adotar $\lambda=1$.

Quanto mais eficiente for o método de minimização maior poderá ser a tolerância ϵ indicada em (3.32). Em métodos como Newton-Raphson pode-se adotar $\epsilon \approx 1$ (em geral adota-se $\epsilon=1$). Para métodos como o BFGS um valor alto como $\epsilon=0,9$ pode corresponder a um valor ótimo para λ .

Deve-se observar que o processo de procura poderá indicar uma extrapolação para λ como na figura 3.2.

Um valor máximo para λ_2 deve ser imposto para evitar extrapolações que causem instabilidade numérica.

A pesquisa linear pode, também, indicar que λ deve ser negativo. Isto ocorre quando a direção \underline{S}_i indica uma direção de crescimento da função $f(\underline{U})$. Na figura 3.3 indica-se a situação.

Neste caso pode-se continuar o processo de procura com valores negativos para λ ou tomar $\lambda=-1$.

Diversos algoritmos de pesquisa linear são apresentados na literatura. A grande questão que deve ser resolvida é a seguinte: o tempo de processamento necessário

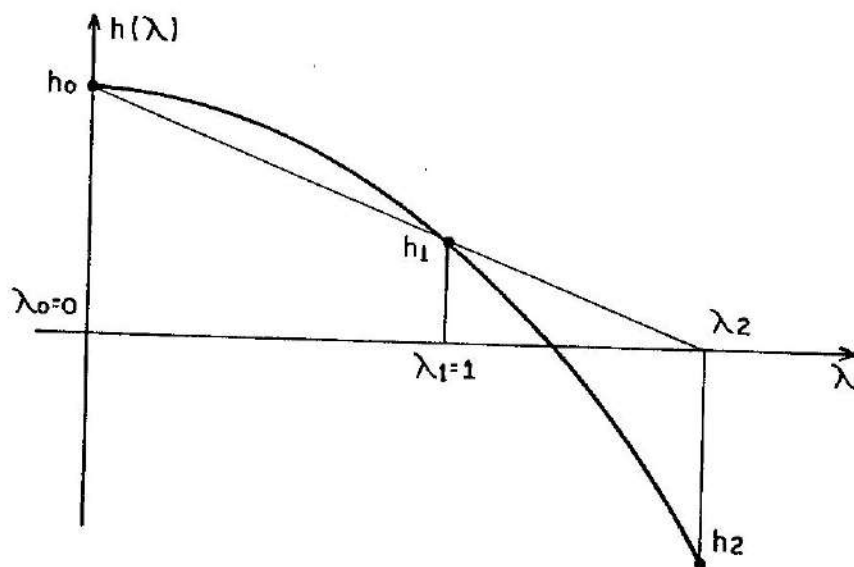


Figura 3.2 - Extrapolação positiva

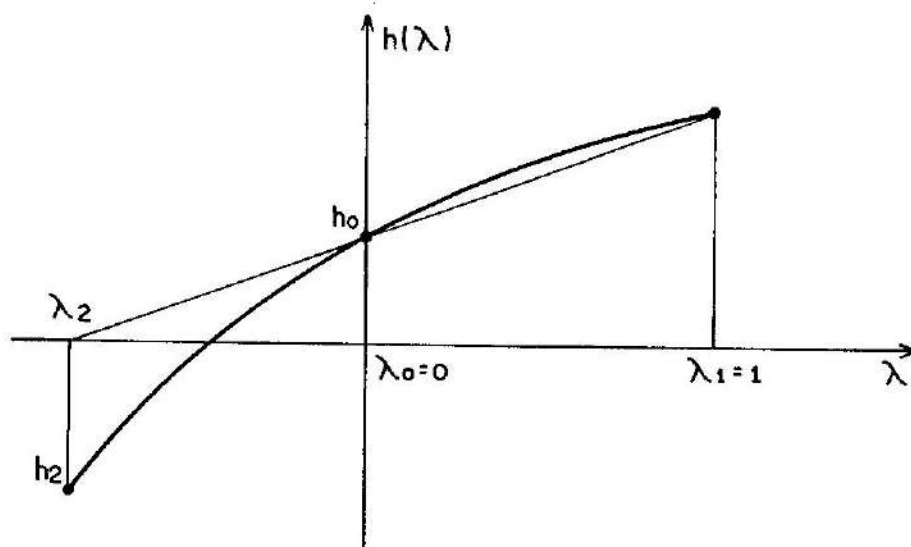


Figura 3.3 - Extrapolação negativa

para se efetuar a pesquisa linear compensa-se no tempo total? Deve ser salientado que as avaliações do gradiente no processo de procura podem ser extremamente trabalhosas, tornando mais conveniente adotar $\lambda=1$.

3.3 - Método de Powell

Como foi visto na seção 3.1 se uma função quadrática é minimizada sequencialmente em relação a direções conjugadas o mínimo da função será localizado em até n passos, independentemente do ponto de partida. O método de Powell é um método que gera automaticamente direções conjugadas.

Sejam dois vetores \underline{U}_1 e \underline{U}_2 e uma direção \underline{S} . Se \underline{Z}_1 é um ponto de mínimo para $f(\underline{U})$ a partir de \underline{U}_1 ao longo de \underline{S} e \underline{Z}_2 é o mínimo a partir de \underline{U}_2 ao longo de \underline{S} , então

$$\underline{Z}_1 = \underline{U}_1 + \lambda_1 \underline{S} \quad (3.33)$$

$$\underline{Z}_2 = \underline{U}_2 + \lambda_2 \underline{S} \quad (3.34)$$

onde

$$\lambda_1 = - \frac{\underline{S}^T (2\underline{Q}\underline{U}_1 + \underline{C})}{2\underline{S}^T \underline{Q} \underline{S}} \quad (3.35)$$

$$\lambda_2 = - \frac{\underline{S}^T (2\underline{Q}\underline{U}_2 + \underline{C})}{2\underline{S}^T \underline{Q} \underline{S}} \quad (3.36)$$

como foi visto na seção 3.1.

Então as direções $\underline{Z} = \underline{Z}_1 - \underline{Z}_2$ e \underline{S} são \underline{Q} -conjugadas, o que pode ser facilmente demonstrado.

O método de Powell se desenvolve nos seguintes passos:
Escolher o vetor de partida \underline{U}_1 e a direção de

minimização S_1 .

$$1) \begin{cases} U_2 = U_1 + \lambda_1 S_1 \\ Z_1 = U_2 \end{cases} \Rightarrow \lambda_1 \text{ que minimiza } f(U_2)$$

$$2) \begin{cases} U_3 = U_2 + \lambda_2 S_1 \\ Z_2 = U_3 \end{cases} \Rightarrow \lambda_2 \text{ que minimiza } f(U_3)$$

$$3) \text{ Nova direção de pesquisa: } S_2 = Z_2 - Z_1$$

$$4) \text{ Fazer: } \begin{cases} S_1 = S_2 \\ U_1 = U_3 \end{cases} \text{ e voltar ao passo 1.}$$

A convergência pode ser aceita quando

$$\frac{\|g(U_j)\|}{\|g(U_1)\|} \leq \text{tolerância.}$$

3.4 - O método descendente por degraus

Como já foi visto o gradiente de $f(U)$ em U_i é um vetor que aponta na direção de maior crescimento da função. O negativo do gradiente aponta na direção de maior decréscimo e, portanto, pode ser usado como uma direção de minimização. Então, o processo iterativo pode ser desenvolvido com a relação

$$U_{i+1} = U_i + \lambda S_i \quad (3.37)$$

com

$$S_i = -g_i \quad (3.38)$$

ou

$$S_i = - \frac{g_i}{\|g_i\|} \quad (3.39)$$

O negativo do gradiente dá a direção para a minimização mas não o tamanho do passo. Uma minimização em relação a λ é necessária para se avaliar U_{i+1} . Isto pode ser feito em função da matriz Hessiana ou através de uma pesquisa numérica. Substituindo (3.38) em (3.20) chega-se ao valor de λ que minimiza $f(U_{i+1})$, como

$$\lambda = \frac{g_i^T g_i}{g_i^T H_i g_i} \quad (3.40)$$

A determinação de λ por (3.40) fica prejudicada pela necessidade da montagem da matriz Hessiana H_i . Uma alternativa seria usar uma aproximação da matriz H_i , dada por um método quase-Newton que será discutido mais a frente.

3.5 - O método dos gradientes conjugados

Este método é devido a Fletcher e Reeves. O método se desenvolve nos seguintes passos:

U_0 = vetor de partida

g_0 = gradiente de $f(U_0)$

1) Tomar $S_0 = -g_0$

$U_1 = U_0 + \lambda_0 S_0 \Rightarrow \lambda_0$ que minimiza $f(U_1)$.

g_1 = gradiente de $f(U_1)$

$$\beta_0 = \frac{\|g_1\|^2}{\|g_0\|^2} \Rightarrow \beta_0 = \frac{g_1^T g_1}{g_0^T g_0}$$

2) Tomar $S_1 = -g_1 + \beta_0 S_0$

$$U_2 = U_1 + \lambda_1 S_1 \Rightarrow \lambda_1 \text{ que minimiza } f(U_2)$$

3) Em uma iteração $(i+2)$ são conhecidos:

$$g_i, g_{i+1}, S_i$$

Então

$$S_{i+1} = -g_{i+1} + \beta_i S_i$$

onde

$$\beta_i = \frac{g_{i+1}^T g_{i+1}}{g_i^T g_i}$$

$$U_{i+2} = U_{i+1} + \lambda_{i+1} S_{i+1} \Rightarrow \lambda_{i+1} \text{ que minimiza } f(U_{i+2}).$$

Para uma função quadrática o método gera automaticamente um conjunto de direções conjugadas. Então a convergência pode ser esperada em até n passos. O método exige a determinação do escalar λ com todos os inconvenientes já mencionados.

3.6 - Os métodos de Newton-Raphson

Os métodos gradientes são métodos de primeira ordem (somente primeiras derivadas são usadas). Nesta classe encontra-se o método descendente por degraus e o método dos gradientes conjugados.

Os métodos não-gradientes são de ordem zero já que nenhuma derivada da função é usada. Nesta classe encontra-se o método de Powell.

Os métodos de segunda ordem empregam as segundas derivadas da função, ou seja, fazem uso da matriz Hessiana.

Nesta classe encontram-se os métodos de Newton-Raphson e os métodos quase-Newton.

O método de Newton-Raphson clássico é baseado na minimização de uma aproximação quadrática da função $f(\underline{U})$.

Expandindo $f(\underline{U})$ em série de Taylor até os termos quadráticos vem

$$\bar{f}(\underline{U}_{i+1}) = f(\underline{U}_i) + \Delta \underline{U}_i^T \underline{g}_i + \frac{1}{2} \Delta \underline{U}_i^T \underline{H}_i \Delta \underline{U}_i \quad (3.41)$$

$$\text{onde } \Delta \underline{U}_i = \underline{U}_{i+1} - \underline{U}_i$$

Logo

$$\bar{f}(\underline{U}_{i+1}) = f(\underline{U}_i) + (\underline{U}_{i+1} - \underline{U}_i)^T \underline{g}_i + \frac{1}{2} (\underline{U}_{i+1} - \underline{U}_i)^T \underline{H}_i (\underline{U}_{i+1} - \underline{U}_i) \quad (3.42)$$

Minimizando $\bar{f}(\underline{U}_{i+1})$ em relação a \underline{U}_i vem

$$\frac{\partial \bar{f}(\underline{U}_{i+1})}{\partial \underline{U}_i} = \underline{0} \quad (3.43)$$

$$-\underline{g}_i - \underline{H}_i \underline{U}_{i+1} + \underline{H}_i \underline{U}_i = \underline{0} \quad (3.44)$$

$$\underline{H}_i \underline{U}_{i+1} = \underline{H}_i \underline{U}_i - \underline{g}_i \quad (3.45)$$

e premultiplicando pela inversa \underline{H}_i^{-1} chega-se a

$$\underline{U}_{i+1} = \underline{U}_i - \underline{H}_i^{-1} \underline{g}_i \quad (3.46)$$

Observa-se que a expressão (3.46) é do tipo

$$\underline{U}_{i+1} = \underline{U}_i + \lambda \underline{S}_i \quad (3.47)$$

com $\lambda=1$ e
$$\underline{s}_i = -\underline{H}_i^{-1} \underline{g}_i.$$

Logo, o método de Newton-Raphson fornece a direção de minimização e o tamanho do passo.

Se a função $f(\underline{U})$ for quadrática a solução será obtida em um único passo uma vez que a expansão em série de Taylor é exata.

O método pode divergir em alguns problemas ou convergir para pontos de sela ou pontos de máximo.

O método pode ser melhorado procurando-se o escalar $\lambda \neq 1$ que minimiza a função real $f(\underline{U})$, já que $\lambda=1$ só minimiza a função quadrática $\bar{f}(\underline{U})$.

Esta modificação tem as seguintes vantagens sobre o método original:

- pode assegurar a convergência quando o método original diverge;
- usualmente ela evita a convergência a um ponto de sela ou a um máximo;
- com a modificação o método se torna o mais poderoso dos métodos de minimização (em termos do número de iterações necessárias).

Entretanto, a modificação traz como desvantagem os inconvenientes da pesquisa numérica para a determinação de λ . Além disto, a propriedade que no ponto de mínimo $\underline{s}_i^T \underline{g}_{i+1} = 0$ só é válida para funções quadráticas. Um processo de procura que força $\underline{s}_i^T \underline{g}_{i+1} = 0$ para uma função geral poderá não levar ao valor ótimo de λ . Assim, usualmente, adota-se $\lambda=1$.

O método de Newton-Raphson clássico tem o inconveniente da necessidade de se montar e decompor a matriz Hessiana em cada iteração. Este fato pode tornar o método impraticável para grandes problemas.

Uma alternativa consiste em se manter a mesma matriz Hessiana em um certo número de iterações, modificando-a

quando a taxa de convergência começa a se deteriorar.

Outra alternativa seria adotar uma aproximação da Hessiana como nos métodos quase-Newton.

O método pode também ser usado na forma

$$\underline{U}_{i+1} = \underline{U}_i - \underline{H}_0^{-1} \underline{g}_i \quad (3.48)$$

onde $\underline{H}_0^{-1} = \underline{H}^{-1}(\underline{U}_0)$ é a inversa da matriz inicial.

A equação (3.48) é a fórmula de recorrência do método de Newton-Raphson modificado.

3.7 - Métodos quase-Newton

3.7.1 - Introdução

Uma classe de métodos denominados "quase-Newton" aproximam a matriz Hessiana, ou sua inversa, usando informações somente das primeiras derivadas. Assim, se $\eta(\underline{U}_i)$ é uma aproximação de \underline{H}_i^{-1} pode-se empregar a fórmula de recorrência do método de Newton-Raphson (equação 3.47) para se obter um novo ponto em direção ao mínimo.

A equação de recorrência toma a forma

$$\underline{U}_{i+1} = \underline{U}_i - \lambda \eta_i \underline{g}_i \quad (3.49)$$

No método de Newton-Raphson necessita-se das segundas derivadas da função para montar \underline{H}_i e depois é preciso decompor a matriz. Nos métodos quase-Newton a inversa \underline{H}_i^{-1} é tomada aproximadamente como η_i .

Seja $f(\underline{U})$ uma função quadrática. Então a matriz Hessiana de $f(\underline{U})$ é uma matriz constante. Expandindo o gradiente de $f(\underline{U})$ em série de Taylor em torno de \underline{U}_i vem

$$\underline{g}_{i+1} = \underline{g}_i + \underline{H}_i \Delta \underline{U}_i \quad (3.50)$$

porque a função gradiente de U não suporta derivadas segundas.

Da equação (3.50) tem-se

$$\Delta \tilde{U}_i = H_i^{-1} (g_{i+1} - g_i) \quad (3.51)$$

sendo que $\Delta \tilde{U}_i = U_{i+1} - U_i$.

Seja a aproximação

$$H_{i+1}^{-1} \approx \omega \eta_{i+1} = \omega (\eta_i + \Delta \eta_i), \quad (3.52)$$

isto é

$$\eta_{i+1} = \eta_i + \Delta \eta_i. \quad (3.53)$$

O escalar ω é um fator de escala, normalmente igual a 1. A seleção de $\Delta \eta_i$ determina o tipo de método quase-Newton.

Para garantir a convergência, $\omega \eta_{i+1}$ deve ser positivo definida e satisfazer a equação

$$\Delta \tilde{U}_i = \omega \eta_{i+1} (g_{i+1} - g_i). \quad (3.54)$$

Se $\chi_i = (g_{i+1} - g_i)$, deve-se ter

$$\eta_{i+1} \chi_i = \frac{1}{\omega} \Delta \tilde{U}_i \quad (3.55)$$

Substituindo (3.53) em (3.55) chega-se a

$$\Delta \eta_i \chi_i = \frac{1}{\omega} \Delta \tilde{U}_i - \eta_i \chi_i \quad (3.56)$$

A solução da equação (3.56), para $\Delta \eta_i$, é da forma

$$\Delta \eta_i = \frac{1}{\omega} \frac{\Delta \tilde{U}_i Y^T}{Y^T \chi_i} - \frac{\eta_i \chi_i Z^T}{Z^T \chi_i} \quad (3.57)$$

onde \underline{Y} e \underline{Z} são vetores arbitrários.

Os diversos algoritmos quase-Newton se diferenciam na escolha dos vetores \underline{Y} e \underline{Z} e do escalar ω .

3.7.2 - Método de Broyden

Neste método adota-se $\omega = 1$ e

$$\underline{Y} = \underline{Z} = \Delta \underline{U}_i - \eta_i \chi_i \quad (3.58)$$

Substituindo (3.58) em (3.57) e considerando $\omega = 1$ chega-se a

$$\Delta \eta_i = \frac{[\Delta \underline{U}_i - \eta_i \chi_i] [\Delta \underline{U}_i - \eta_i \chi_i]^T}{[\Delta \underline{U}_i - \eta_i \chi_i]^T \chi_i} \quad (3.59)$$

O algoritmo é iniciado com \underline{U}_0 e uma η_0 positivo definida. Pode-se adotar, preferencialmente, $\eta_0 = H_0^{-1}$. Numa iteração i tem-se que

$$\underline{U}_{i+1} = \underline{U}_i - \lambda \eta_i \underline{g}_i \quad (3.60)$$

Se λ é o escalar que minimiza $f(\underline{U}_{i+1})$ então o método gera direções conjugadas.

O escalar λ pode ser obtido por uma pesquisa numérica ou, simplesmente, tomado igual a 1.

3.7.3 - Método de Davidon-Fletcher-Powell (DFP)

Neste método adota-se $\omega = 1$ e

$$\underline{Y} = \Delta \underline{U}_i \quad (3.61)$$

$$\underline{Z} = \eta_i \chi_i \quad (3.62)$$

Substituindo em (3.57) vem

$$\Delta \eta_i = \frac{\Delta \underline{U}_i \Delta \underline{U}_i^T}{\Delta \underline{U}_i^T \chi_i} - \frac{\eta_i \chi_i \chi_i^T \eta_i}{\chi_i^T \eta_i \chi_i} \quad (3.63)$$

3.7.4 - Algoritmos de Pearson

Pearson propôs vários métodos para computar η . Os algoritmos podem ser obtidos por substituição de diferentes vetores \underline{Y} e \underline{Z} na equação (3.57).

Algoritmo no. 1:

Tomando $\omega=1$ e $\underline{Y}=\underline{Z}=\Delta \underline{U}_i$, vem

$$\Delta \eta_i = \frac{\left[\Delta \underline{U}_i - \eta_i \chi_i \right] \Delta \underline{U}_i^T}{\Delta \underline{U}_i^T \chi_i} \quad (3.64)$$

Algoritmo no. 2:

Tomando $\omega=1$ e $\underline{Y}=\underline{Z}=\eta_i \chi_i$, vem

$$\Delta \eta_i = \frac{\left[\Delta \underline{U}_i - \eta_i \chi_i \right] \left[\eta_i \chi_i \right]^T}{\chi_i^T \eta_i \chi_i} \quad (3.65)$$

3.7.5 - Método BFGS

Este método é devido a Broyden, Fletcher, Goldfarb e Shanno.

A expressão proposta para η_{i+1} atende a equação (3.55) com $\omega = 1$ e é da forma

$$\eta_{i+1} = \begin{bmatrix} \mathbb{I} - \frac{\Delta U_i \chi_i^T}{\Delta U_i^T \chi_i} \end{bmatrix} \eta_i \begin{bmatrix} \mathbb{I} - \frac{\chi_i \Delta U_i^T}{\Delta U_i^T \chi_i} \end{bmatrix} + \frac{\Delta U_i \Delta U_i^T}{\Delta U_i^T \chi_i} \quad (3.66)$$

3.8 - O Método Secante-Newton

Este método foi desenvolvido por Crisfield que empregou uma relação secante entre o gradiente e as direções de pesquisa. O método consiste de uma aceleração do método de Newton-Raphson modificado. Daí vem o nome Secante-Newton. O método pode ser derivado, também, por um processo de minimização da função $f(U)$.

Seja um processo iterativo onde se conhecem os vetores U_{i-1} , U_i , g_{i-1} e g_i .

Uma nova aproximação ao mínimo poderia ser tomada na forma

$$U_{i+1} = U_{i-1} + \alpha \Delta U_{i-1} \quad (3.67)$$

onde

$$\Delta U_{i-1} = U_i - U_{i-1} \quad (3.68)$$

Expandindo $f(U_{i+1})$ em série de Taylor até os termos quadráticos vem

$$f(U_{i+1}) \approx f(U_{i-1}) + \alpha \Delta U_{i-1}^T g_{i-1} + \frac{1}{2} \alpha^2 \Delta U_{i-1}^T H_{i-1} \Delta U_{i-1} \quad (3.69)$$

Minimizando em relação a α chega-se a

$$\alpha = - \frac{\Delta U_{i-1}^T g_{i-1}}{\Delta U_{i-1}^T H_{i-1} \Delta U_{i-1}} \quad (3.70)$$

Como uma aproximação adota-se

$$H_{\tilde{u}_{i-1}} \Delta \tilde{u}_{i-1} = g_i - g_{i-1} = \chi_{i-1} \quad (3.71)$$

A equação (3.71) é exata se $f(\tilde{u})$ for quadrática conforme demonstrado pela equação (3.50).

Substituindo (3.71) em (3.70) resulta

$$\alpha = - \frac{\Delta \tilde{u}_{i-1}^T g_{i-1}}{\Delta \tilde{u}_{i-1}^T \chi_{i-1}} \quad (3.72)$$

Seja, agora, a aproximação

$$\tilde{u}_{i+1} = \tilde{u}_i + \alpha \bar{\Delta \tilde{u}}_i + \beta \Delta \tilde{u}_{i-1} \quad (3.73)$$

onde α é dado por (3.72) e $\bar{\Delta \tilde{u}}_i$ é a direção de pesquisa do método de Newton-Raphson modificado, ou seja,

$$\bar{\Delta \tilde{u}}_i = - H_{\tilde{u}_0}^{-1} g_i \quad (3.74)$$

onde $H_{\tilde{u}_0}^{-1} = H^{-1}(\tilde{u}_0)$ é a inversa da matriz Hessiana inicial.

Expandindo $f(\tilde{u}_{i+1})$ em série de Taylor até os termos quadráticos vem

$$f(\tilde{u}_{i+1}) = f(\tilde{u}_i) + \left[\alpha \bar{\Delta \tilde{u}}_i^T + \beta \Delta \tilde{u}_{i-1}^T \right] g_i + \frac{1}{2} \left[\alpha \bar{\Delta \tilde{u}}_i^T + \beta \Delta \tilde{u}_{i-1}^T \right] H_{\tilde{u}_i} \left[\alpha \bar{\Delta \tilde{u}}_i + \beta \Delta \tilde{u}_{i-1} \right] \quad (3.75)$$

Minimizando em relação a β chega-se a

$$\beta = - \frac{\Delta U_{\tilde{i}-1}^T g_i + \alpha \Delta \bar{U}_{\tilde{i}}^T H_{\tilde{i}} \Delta U_{\tilde{i}-1}}{\Delta U_{\tilde{i}-1}^T H_{\tilde{i}} \Delta U_{\tilde{i}-1}} . \quad (3.76)$$

Com uma aproximação adota-se, também,

$$H_{\tilde{i}} \Delta U_{\tilde{i}} = g_i - g_{i-1} = \chi_{i-1} \quad (3.77)$$

já que para uma função quadrática a matriz Hessiana é constante, $H_{\tilde{i}} = H_{\tilde{i}-1}$, e a equação (3.77) é igual a (3.71).

Introduzindo (3.77) em (3.76) vem

$$\beta = - \frac{\left[\Delta U_{\tilde{i}-1}^T g_i + \alpha \Delta \bar{U}_{\tilde{i}}^T \chi_{i-1} \right]}{\Delta U_{\tilde{i}-1}^T \chi_{i-1}} . \quad (3.78)$$

Considerando a expressão (3.72) para α e lembrando que $\chi_{i-1} = g_i - g_{i-1}$ pode-se mostrar que a equação (3.78) é equivalente a

$$\beta = \alpha \left[1 - \frac{\Delta \bar{U}_{\tilde{i}}^T \chi_{i-1}}{\Delta U_{\tilde{i}-1}^T \chi_{i-1}} \right] - 1 . \quad (3.79)$$

Crisfield empregou o método para análise de lajes de concreto armado e concluiu que, quando o escalar β é grande em relação a α , a estrapolação com a equação (3.73) torna-se instável. Nesses casos é melhor empregar uma iteração do método de Newton-Raphson modificando tomando $\alpha = 1$ e $\beta = 0$. Então, a equação (3.73) só deve ser usada se

$$- 0,2 \leq \frac{\beta}{\alpha} \leq 0,4 . \quad (3.80)$$

Os limites apresentados em (3.81) são empíricos e foram determinados em um problema particular.

3.9 - Um método de minimização bidirecional

Este método foi desenvolvido pelo primeiro autor deste trabalho e baseia-se na minimização da função $f(\underline{U})$ em relação a duas direções simultaneamente. No método secante-Newton as minimizações são efetuadas em relação a duas direções, mas de maneira desacoplada.

Seja a aproximação

$$\underline{U}_{\sim i+1} = \underline{U}_{\sim i-1} + \lambda_1 \underline{\Delta U}_{\sim i-1} \quad (3.81)$$

onde $\underline{\Delta U}_{\sim i-1} = \underline{U}_{\sim i} - \underline{U}_{\sim i-1}$.

Minimizando $f(\underline{U}_{\sim i+1})$ em relação a λ_1 vem

$$\lambda_1 = - \frac{\underline{\Delta U}_{\sim i-1}^T \underline{g}_{i-1}}{\underline{\Delta U}_{\sim i-1}^T \underline{H}_{\sim i-1} \underline{\Delta U}_{\sim i-1}} \quad (3.82)$$

e verifica-se a propriedade

$$\underline{\Delta U}_{\sim i-1}^T \underline{H}_{\sim i-1} \underline{\Delta U}_{\sim i-1} = - \frac{1}{\lambda_1} \underline{\Delta U}_{\sim i-1}^T \underline{g}_{i-1} \quad (3.83)$$

Uma expressão análoga a (3.81) seria

$$\underline{U}_{\sim i+1} = \underline{U}_{\sim i} + \lambda_2 \underline{\Delta \bar{U}}_{\sim i} \quad (3.84)$$

onde $\underline{\Delta \bar{U}}_{\sim i} = - \underline{H}_{\sim 0}^{-1} \underline{g}_i$ é a direção de pesquisa do método de Newton-Raphson modificado.

Minimizando em relação a λ_2 obtém-se o escalar λ_2 e verifica-se a propriedade

$$\Delta \bar{U}_{\sim i}^T H_{\sim i} \Delta \bar{U}_{\sim i} = - \frac{1}{\lambda_2} \Delta \bar{U}_{\sim i}^T g_{\sim i} \quad (3.85)$$

Como uma aproximação adota-se $\lambda_1 = \lambda_2 = \lambda$ e

$$H_{\sim i} \Delta U_{\sim i-1} = H_{\sim i-1} \Delta U_{\sim i-1} = g_{\sim i} - g_{\sim i-1} = \zeta_{\sim i-1} \quad (3.86)$$

Introduzindo (3.86) em (3.82) resulta

$$\lambda = - \frac{\Delta U_{\sim i-1}^T g_{\sim i-1}}{\Delta U_{\sim i-1}^T \zeta_{\sim i-1}} \quad (3.87)$$

e da equação (3.85) tem-se

$$\Delta \bar{U}_{\sim i}^T H_{\sim i} \Delta \bar{U}_{\sim i} = - \frac{1}{\lambda} \Delta \bar{U}_{\sim i}^T g_{\sim i} \quad (3.88)$$

Seja agora a aproximação

$$U_{\sim i+1} = U_{\sim i} + \alpha \Delta U_{\sim i-1} + \beta \Delta \bar{U}_{\sim i} \quad (3.89)$$

Expandindo $f(U_{\sim i+1})$ em série de Taylor até os termos quadráticos e minimizando em relação a α e β chega-se ao sistema de equações

$$\begin{bmatrix} \Delta U_{\sim i-1}^T H_{\sim i} \Delta U_{\sim i-1} & \Delta \bar{U}_{\sim i}^T H_{\sim i} \Delta U_{\sim i-1} \\ \Delta \bar{U}_{\sim i}^T H_{\sim i} \Delta U_{\sim i-1} & \Delta \bar{U}_{\sim i}^T H_{\sim i} \Delta \bar{U}_{\sim i} \end{bmatrix} \begin{Bmatrix} \alpha \\ \beta \end{Bmatrix} = \begin{Bmatrix} -\Delta U_{\sim i-1}^T g_{\sim i} \\ -\Delta \bar{U}_{\sim i}^T g_{\sim i} \end{Bmatrix} \quad (3.90)$$

Introduzindo (3.86) e (3.88) no sistema chega-se a

$$\begin{bmatrix} \Delta \bar{U}_{i-1}^T \chi_{i-1} & \Delta \bar{U}_i^T \chi_{i-1} \\ \Delta \bar{U}_i^T \chi_{i-1} & -\frac{1}{\lambda} \Delta \bar{U}_i^T g_i \end{bmatrix} \begin{Bmatrix} \alpha \\ \beta \end{Bmatrix} = \begin{Bmatrix} -\Delta \bar{U}_{i-1}^T g_i \\ -\Delta \bar{U}_i^T g_i \end{Bmatrix} \quad (3.91)$$

onde λ é dado por (3.87).

A solução do sistema (3.91) fornece α e β com os quais se encontra a nova aproximação do mínimo pela equação (3.89). Experiências numéricas com o método indicaram que, dependendo da relação α/β , a extrapolação com a equação (3.89) leva a uma instabilidade numérica. Neste caso deve-se adotar $\alpha=0$ e $\beta=1$ e seguir como no método de Newton-Raphson modificado. Assim, a equação (3.89) só deve ser usada se

$$0 \leq \frac{\alpha}{\beta} \leq 1. \quad (3.92)$$

Os limites dados em (3.92) são empíricos e necessitam de uma aferição para cada problema específico.

4 - APLICAÇÃO À ANÁLISE ESTRUTURAL

4.1 - Introdução

No problema de análise estrutural a função objetiva a ser minimizada é a energia potencial total da estrutura. Conforme foi mostrado no capítulo 2, para esta função a matriz Hessiana é a matriz de rigidez tangente da estrutura e o gradiente é igual ao vetor das forças desequilibradas com o sinal trocado, ou seja,

$$H_{\tilde{u}_i} = K_{\tilde{u}_i} \quad (4.1)$$

$$g_i = - \psi_i \quad (4.2)$$

onde

$$\psi_i = P_{\tilde{u}_i} - A_{NL}(\tilde{u}_i) \quad (4.3)$$

O vetor $A_{NL}(\tilde{u}_i)$ representa as ações nodais não-lineares obtidas em função dos deslocamentos nodais \tilde{u}_i e $P_{\tilde{u}_i}$ é o vetor de cargas nodais.

O equilíbrio é obtido quando $\psi_i = 0$. No processo iterativo, entretanto, admite-se a convergência quando

$$\frac{\| \psi_i \|}{\| P_{\tilde{u}_i} \|} \leq \text{tolerância.} \quad (4.4)$$

Para a obtenção da resposta completa da estrutura até

um nível de carga desejado $P_{\sim \max}$, o carregamento é aplicado em incrementos ΔP_{\sim} . Em geral, adotam-se incrementos iguais de forma que

$$\Delta P_{\sim} = \frac{P_{\sim \max}}{n_{\text{etap}}} \quad (4.5)$$

onde n_{etap} é o número de etapas de carga desejado.

Os processos iterativos apresentados podem ser representados pela fórmula de recorrência

$$U_{\sim i+1} = U_{\sim i} + \Delta U_{\sim i} \quad (4.6)$$

onde em $\Delta U_{\sim i}$ já está incluído o tamanho do passo que depende do método empregado.

Desta forma a convergência é testada controlando-se simultaneamente o vetor de desequilíbrio ψ_i e o vetor $\Delta U_{\sim i}$. Assim, considera-se que o método convergiu quando

$$\frac{\|\psi_i\|}{\|\Delta P_{\sim}\|} \leq \text{tolerância} \quad (4.7)$$

e, simultaneamente,

$$\frac{\|\Delta U_{\sim i}\|}{\|U_{\sim i+1}\|} \leq \text{tolerância.} \quad (4.8)$$

Como tolerância pode-se tomar o valor 0,01.

4.2 - Resumo dos métodos de minimização

Empregando a notação da seção anterior os métodos de minimização tomam as formas apresentadas a seguir.

a) Método descendente por degraus

$$\underline{U}_{i+1} = \underline{U}_i + \lambda \underline{\psi}_i \quad (4.9)$$

sendo

$$\lambda = \frac{\underline{\psi}_i^T \underline{\psi}_i}{\underline{\psi}_i^T \underline{K}_i \underline{\psi}_i} \quad (4.10)$$

b) Método dos gradientes conjugados

$$\underline{U}_1 = \underline{U}_0 + \lambda_0 \underline{S}_0 \quad (4.11)$$

com $\underline{S}_0 = \underline{\psi}_0$.

Para uma iteração genérica vem

$$\underline{U}_{i+1} = \underline{U}_i + \lambda_i \underline{S}_i \quad (4.12)$$

com

$$\underline{S}_i = \underline{\psi}_i + \beta_{i-1} \underline{S}_{i-1} \quad (4.13)$$

$$\beta_{i-1} = \frac{\underline{\psi}_i^T \underline{\psi}_i}{\underline{\psi}_{i-1}^T \underline{\psi}_i} \quad (4.14)$$

c) Métodos de Newton-Raphson

$$\underline{U}_{i+1} = \underline{U}_i + \lambda \underline{S}_i \quad (4.15)$$

com

$$\underline{S}_i = \underline{K}_i^{-1} \underline{\psi}_i \quad (4.16)$$

No método de Newton-Raphson modificado vem

$$\tilde{S}_i = K_{i0}^{-1} \psi_i \quad (4.17)$$

Em geral, $\lambda = 1$.

d) Métodos quase-Newton

A fórmula de recorrência é

$$U_{\tilde{i}+1} = U_{\tilde{i}} + \lambda K_{\tilde{i}}^{-1} \psi_i \quad (4.18)$$

onde $K_{\tilde{i}}^{-1}$ é uma aproximação da inversa da matriz de rigidez.

d.1) Método de Broyden

$$K_{\tilde{i}}^{-1} = K_{\tilde{i}-1}^{-1} + \frac{\left[\Delta U_{\tilde{i}-1} - K_{\tilde{i}-1}^{-1} \chi_{i-1} \right] \left[\Delta U_{\tilde{i}-1} - K_{\tilde{i}-1}^{-1} \chi_{i-1} \right]^T}{\left[\Delta U_{\tilde{i}-1} - K_{\tilde{i}-1}^{-1} \chi_{i-1} \right]^T \chi_{i-1}} \quad (4.19)$$

sendo

$$\Delta U_{\tilde{i}-1} = U_{\tilde{i}} - U_{\tilde{i}-1} \quad (4.20)$$

e

$$\chi_{i-1} = -\psi_i + \psi_{i-1} \quad (4.21)$$

d.2) Método de Davidon-Fletcher-Powell

$$K_{\tilde{i}}^{-1} = K_{\tilde{i}-1}^{-1} + \frac{\Delta U_{\tilde{i}-1} \Delta U_{\tilde{i}-1}^T}{\Delta U_{\tilde{i}-1}^T \chi_{i-1}} - \frac{K_{\tilde{i}-1}^{-1} \chi_{i-1} \chi_{i-1}^T K_{\tilde{i}-1}^{-1}}{\chi_{i-1}^T K_{\tilde{i}-1}^{-1} \chi_{i-1}} \quad (4.22)$$

d.3) Algoritmos de Pearson

- Algoritmo no. 1:

$$K_{\sim i}^{-1} = K_{\sim i-1}^{-1} + \frac{\left[\Delta U_{\sim i-1} - K_{\sim i-1}^{-1} \chi_{i-1} \right] \Delta U_{\sim i-1}^T}{\Delta U_{\sim i-1}^T \chi_{i-1}} \quad (4.23)$$

- Algoritmo no. 2:

$$K_{\sim i}^{-1} = K_{\sim i-1}^{-1} + \frac{\left[\Delta U_{\sim i-1} - K_{\sim i-1}^{-1} \chi_{i-1} \right] \left[K_{\sim i-1}^{-1} \chi_{i-1} \right]^T}{\chi_{i-1}^T K_{\sim i-1}^{-1} \chi_{i-1}} \quad (4.24)$$

d.4) Método BFGS

$$K_{\sim i}^{-1} = \begin{bmatrix} I - \frac{\Delta U_{\sim i-1} \chi_{i-1}^T}{\Delta U_{\sim i-1}^T \chi_{i-1}} \\ \frac{\Delta U_{\sim i-1}^T \chi_{i-1}}{\Delta U_{\sim i-1}^T \chi_{i-1}} \end{bmatrix} K_{\sim i-1}^{-1} \begin{bmatrix} I - \frac{\chi_{i-1} \Delta U_{\sim i-1}^T}{\Delta U_{\sim i-1}^T \chi_{i-1}} \\ \frac{\Delta U_{\sim i-1}^T \chi_{i-1}}{\Delta U_{\sim i-1}^T \chi_{i-1}} \end{bmatrix} + \frac{\Delta U_{\sim i-1} \Delta U_{\sim i-1}^T}{\Delta U_{\sim i-1}^T \chi_{i-1}} \quad (4.25)$$

e) Método Secante-Newton

$$U_{\sim i+1} = U_{\sim i} + \alpha \Delta \bar{U}_{\sim i} + \beta \Delta U_{\sim i-1} \quad (4.26)$$

com

$$\Delta \bar{U}_{\sim i} = K_{\sim 0}^{-1} \psi_i \quad (4.27)$$

$$\alpha = \frac{\Delta U_{\sim i-1}^T \psi_{i-1}}{\Delta U_{\sim i-1}^T \chi_{i-1}} \quad (4.28)$$

$$\beta = \alpha \left[1 - \frac{\Delta \bar{U}_i^T \chi_{i-1}}{\Delta U_{i-1}^T \chi_{i-1}} \right] - 1 \quad (4.29)$$

$$\text{Se } \frac{\beta}{\alpha} > 0,4 \quad \text{ou} \quad \frac{\beta}{\alpha} < -0,2 \quad \Rightarrow \quad \alpha = 1 \quad \text{e} \quad \beta = 0.$$

f) Minimização bidirecional

$$U_{i+1} = U_i + \alpha \Delta U_{i-1} + \beta \Delta \bar{U}_i \quad (4.30)$$

$$\begin{bmatrix} \Delta U_{i-1}^T \chi_{i-1} & \Delta \bar{U}_i^T \chi_{i-1} \\ \Delta \bar{U}_i^T \chi_{i-1} & \frac{1}{\lambda} \Delta \bar{U}_i^T \psi_i \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} \Delta U_{i-1}^T \psi_i \\ \Delta \bar{U}_i^T \psi_i \end{bmatrix} \quad (4.31)$$

$$\text{com } \lambda = \frac{\Delta U_{i-1}^T \psi_{i-1}}{\Delta U_{i-1}^T \chi_{i-1}}.$$

$$\text{Se } \frac{\alpha}{\beta} < 0 \quad \text{ou} \quad \frac{\alpha}{\beta} > 1 \quad \Rightarrow \quad \alpha = 0 \quad \text{e} \quad \beta = 1.$$

4.3- Implementação computacional dos métodos quase-Newton

Os métodos quase-Newton exigem a montagem da inversa da matriz de rigidez para a determinação do incremento do vetor de deslocamentos. Isto pode ser um fator limitante para grandes problemas já que a banda da matriz é destruída e a matriz inversa aproximada é uma matriz cheia (n x n). Uma técnica de operações com vetores pode ser usada para evitar a montagem da matriz inversa. O procedimento é apresentado a seguir para os vários métodos.

Tomando $\lambda = 1$ na equação (4.18) a fórmula de recorrência dos métodos quase-Newton toma a forma

$$\underline{U}_{i+1} = \underline{U}_i + \Delta \underline{U}_i \quad (4.32)$$

onde

$$\Delta \underline{U}_i = \underline{K}_i^{-1} \underline{\psi}_i \quad (4.33)$$

No início do processo iterativo são conhecidos \underline{U}_0 e $\underline{\psi}_0$. Uma iteração do método de Newton-Raphson modificado leva a

$$\underline{U}_1 = \underline{U}_0 + \Delta \underline{U}_0 \quad (4.34)$$

onde

$$\Delta \underline{U}_0 = \underline{K}_0^{-1} \underline{\psi}_0 \quad (4.35)$$

A matriz de rigidez inicial \underline{K}_0 pode ser decomposta pelo método de Cholesky, por exemplo, e a determinação de $\Delta \underline{U}_0$ é feita por simples substituições. A rigor a equação (4.35) é operada na forma

$$\underline{\psi}_0 = \underline{K}_0 \Delta \underline{U}_0 \xrightarrow{\text{substituições}} \Delta \underline{U}_0 \quad (4.36)$$

Com \underline{U}_1 avaliam-se as ações nodais não-lineares e o vetor de desequilíbrio $\underline{\psi}_1$. Daí para frente segue-se com as equações (4.32) e (4.33) onde \underline{K}_i^{-1} toma a forma característica de cada método.

A) Método de Broyden

A equação (4.19) do método de Broyden pode ser escrita na forma

$$\underline{K}_i^{-1} = \underline{K}_{i-1}^{-1} + \frac{\underline{Z}_{i-1} \underline{Z}_{i-1}^T}{\underline{Z}_{i-1}^T (\underline{\psi}_{i-1} - \underline{\psi}_i)} \quad (4.37)$$

sendo

$$\tilde{z}_{i-1} = \Delta U_{i-1} - K_{i-1}^{-1} (\psi_{i-1} - \psi_i) \quad (4.38)$$

já que $\tilde{z}_{i-1} = \psi_{i-1} - \psi_i$.
Tomando $i=1$ vem,

$$\tilde{z}_0 = \Delta U_0 - K_0^{-1} (\psi_0 - \psi_1) \quad (4.39)$$

e

$$K_1^{-1} = K_0^{-1} + \frac{\tilde{z}_0 \tilde{z}_0^T}{\tilde{z}_0^T (\psi_0 - \psi_1)} \quad (4.40)$$

Tomando $i = 2$ em (4.38) vem

$$\tilde{z}_1 = \Delta U_1 - K_1^{-1} (\psi_1 - \psi_2) \quad (4.41)$$

e introduzindo (4.40) chega-se a

$$\tilde{z}_1 = \Delta U_1 - K_0^{-1} (\psi_1 - \psi_2) - \frac{\tilde{z}_0 \tilde{z}_0^T (\psi_1 - \psi_2)}{\tilde{z}_0^T (\psi_0 - \psi_1)} \quad (4.42)$$

Tomando $i=2$ em (4.37) vem

$$K_2^{-1} = K_1^{-1} + \frac{\tilde{z}_1 \tilde{z}_1^T}{\tilde{z}_1^T (\psi_1 - \psi_2)} \quad (4.43)$$

e introduzindo (4.40) chega-se a

$$K_2^{-1} = K_0^{-1} + \frac{\tilde{z}_0 \tilde{z}_0^T}{\tilde{z}_0^T (\psi_0 - \psi_1)} + \frac{\tilde{z}_1 \tilde{z}_1^T}{\tilde{z}_1^T (\psi_1 - \psi_2)} \quad (4.44)$$

Procedendo-se desta maneira é possível estabelecer as fórmulas de recorrência

$$\tilde{z}_0 = \Delta U_{\tilde{0}} - K_{\tilde{0}}^{-1}(\psi_0 - \psi_1) \quad (4.45)$$

$$\tilde{z}_i = \Delta U_{\tilde{i}} - K_{\tilde{0}}^{-1}(\psi_i - \psi_{i+1}) - \sum_{n=0}^{i-1} \frac{\tilde{z}_n \tilde{z}_n^T (\psi_i - \psi_{i+1})}{\tilde{z}_n^T (\psi_n - \psi_{n+1})} \quad (4.46)$$

$$K_{\tilde{i}}^{-1} = K_{\tilde{0}}^{-1} + \sum_{n=0}^{i-1} \frac{\tilde{z}_n \tilde{z}_n^T}{\tilde{z}_n^T (\psi_n - \psi_{n+1})} \quad (4.47)$$

$$e \quad \Delta U_{\tilde{i}} = U_{\tilde{i+1}} - U_{\tilde{i}}$$

Considerando a equação (4.33) verifica-se que

$$\Delta U_{\tilde{i}} = K_{\tilde{0}}^{-1} \psi_i + \sum_{n=0}^{i-1} \frac{\tilde{z}_n \tilde{z}_n^T \psi_i}{\tilde{z}_n^T (\psi_n - \psi_{n+1})} \quad (4.48)$$

De forma análoga a (4.48) pode-se escrever

$$\Delta U_{\tilde{i+1}} = K_{\tilde{0}}^{-1} \psi_{i+1} + \sum_{n=0}^{i-1} \frac{\tilde{z}_n \tilde{z}_n^T \psi_{i+1}}{\tilde{z}_n^T (\psi_n - \psi_{n+1})} + \frac{\tilde{z}_i \tilde{z}_i^T \psi_{i+1}}{\tilde{z}_i^T (\psi_i - \psi_{i+1})} \quad (4.49)$$

Fazendo $\Delta U_{\tilde{i}} - \Delta U_{\tilde{i+1}}$ vem

$$\Delta \tilde{U}_i - \Delta \tilde{U}_{i+1} = \tilde{K}_0^{-1} (\psi_i - \psi_{i+1}) + \sum_{n=0}^{i-1} \frac{\tilde{Z}_n \tilde{Z}_n^T (\psi_i - \psi_{i+1})}{\tilde{Z}_n^T (\psi_n - \psi_{n+1})} - \frac{\tilde{Z}_i \tilde{Z}_i^T \psi_{i+1}}{\tilde{Z}_i^T (\psi_i - \psi_{i+1})} \quad (4.50)$$

Introduzindo (4.46) em (4.50) e reordenando os termos chega-se a

$$\Delta \tilde{U}_{i+1} = \frac{\tilde{Z}_i \tilde{Z}_i^T \psi_i}{\tilde{Z}_i^T (\psi_i - \psi_{i+1})} \quad (4.51)$$

Definindo o escalar

$$\beta_n = \tilde{Z}_n^T (\psi_n - \psi_{n+1}) \quad (4.52)$$

a equação (4.46) toma a forma

$$\tilde{Z}_i = (\tilde{U}_{i+1} - \tilde{U}_i) - \tilde{K}_0^{-1} (\psi_i - \psi_{i+1}) - \sum_{n=0}^{i-1} \frac{\tilde{Z}_n \tilde{Z}_n^T (\psi_i - \psi_{i+1})}{\beta_n} \quad (4.53)$$

Desta forma os acréscimos no vetor de deslocamentos podem ser obtidos diretamente por (4.51) em função dos vetores \tilde{Z}_i , ψ_i e ψ_{i+1} . Deve-se observar que para a montagem do vetor \tilde{Z}_i em determinada iteração é necessário o conhecimento de todos os vetores \tilde{Z}_n das iterações anteriores bem como dos escalares β_n . Assim, os vetores \tilde{Z}_n e os escalares β_n devem ser armazenados ao longo do processo. Evidentemente, o número de armazenamentos deve ser limitado para não ocupar muito a memória do computador. Assim, pode-se ir armazenando \tilde{Z}_n e β_n , por exemplo, até a décima iteração e, se a convergência não for alcançada, reinicia-se o processo com a equação (4.35). O algoritmo é

apresentado a seguir:

- Síntese do algoritmo

$$i = 0$$

$$\begin{array}{l} \rightarrow \psi_i = \psi_0 \quad ; \quad U_i = U_0 \\ (2) \quad \psi_i = K_{\sim 0} \Delta U_{\sim i} \Rightarrow \Delta U_{\sim i} \end{array}$$

$$\Delta \bar{U}_{\sim i} = \Delta U_{\sim i}$$

$$U_{\sim i+1} = U_i + \Delta U_{\sim i}$$

$$\text{com } U_{\sim i+1} \Rightarrow \psi_{i+1}$$

$$\psi_{i+1} = K_{\sim 0} \Delta \bar{U}_{\sim i+1} \Rightarrow \Delta \bar{U}_{\sim i+1}$$

$$Z_{\sim i} = \Delta U_{\sim i} - (\Delta \bar{U}_{\sim i} - \Delta \bar{U}_{\sim i+1}) \Rightarrow \text{armazenar}$$

$$\beta_i = Z_{\sim i}^T (\psi_i - \psi_{i+1}) \Rightarrow \text{armazenar}$$

$$\begin{array}{l} \rightarrow i = 1 \Rightarrow \text{imax.} \\ (1) \end{array}$$

$$\Delta U_{\sim i} = \frac{Z_{\sim i} Z_{\sim i}^T \psi_i}{Z_{\sim i}^T (\psi_i - \psi_{i+1})} = \frac{Z_{\sim i} Z_{\sim i}^T \psi_i}{\beta_i}$$

$$U_{\sim i} = U_{\sim i+1} \quad ; \quad \Delta \bar{U}_{\sim i} = \Delta \bar{U}_{\sim i+1} \quad ; \quad \psi_i = \psi_{i+1}$$

$$U_{\sim i+1} = U_{\sim i} + \Delta U_{\sim i}$$

$$\text{com } U_{\sim i+1} \Rightarrow \psi_{i+1}$$

$$\text{se } \frac{\|\psi_{i+1}\|}{\|\Delta P_{\sim}\|} \leq \text{tol} \quad \text{e} \quad \frac{\|\Delta U_{\sim i}\|}{\|U_{\sim i+1}\|} \leq \text{tol} \Rightarrow U_{\sim i+1} \text{ é a solução}$$

$$\text{se não, } \psi_{i+1} = K_{\sim 0} \Delta \bar{U}_{\sim i+1} \Rightarrow \Delta \bar{U}_{\sim i+1}$$

$$Z_{\sim i} = \Delta U_{\sim i} - (\Delta \bar{U}_{\sim i} - \Delta \bar{U}_{\sim i+1}) - \sum_{n=0}^{i-1} \frac{Z_{\sim n} Z_{\sim n}^T (\psi_i - \psi_{i+1})}{\beta_n} \Rightarrow \text{armazenar}$$

$$\beta_i = \tilde{z}_i^T (\psi_i - \psi_{i+1}) \Rightarrow \text{armazenar}$$

voltar para (1)

Se não convergiu em i_{\max} iterações,

$$\psi_0 = \psi_{i+1}$$

$$\tilde{u}_0 = \tilde{u}_{i+1}$$

voltar para (2).

b) Método de Davidon-Fletcher-Powell

A equação (4.22) pode ser escrita na forma.

$$K_{\tilde{z}_i}^{-1} = K_{\tilde{z}_{i-1}}^{-1} + A_{\tilde{z}_{i-1}} - \frac{\tilde{z}_{i-1} \tilde{z}_{i-1}^T}{\tilde{z}_{i-1}^T (\psi_{i-1} - \psi_i)} \quad (4.54)$$

onde

$$A_{\tilde{z}_{i-1}} = \frac{\Delta \tilde{u}_{\tilde{z}_{i-1}} \Delta \tilde{u}_{\tilde{z}_{i-1}}^T}{\Delta \tilde{u}_{\tilde{z}_{i-1}}^T (\psi_{i-1} - \psi_i)} \quad (4.55)$$

$$\tilde{z}_{i-1} = K_{\tilde{z}_{i-1}}^{-1} (\psi_{i-1} - \psi_i) \quad (4.56)$$

Procedendo de maneira análoga ao anteriormente apresentado chega-se a

$$\tilde{z}_0 = K_{\tilde{z}_0}^{-1} (\psi_0 - \psi_1)$$

$$\tilde{z}_i = K_{\tilde{z}_0}^{-1} (\psi_i - \psi_{i+1}) + \sum_{n=0}^{i-1} A_{\tilde{z}_n} (\psi_i - \psi_{i+1}) - \sum_{n=0}^{i-1} \frac{\tilde{z}_n \tilde{z}_n^T (\psi_i - \psi_{i+1})}{\tilde{z}_n^T (\psi_n - \psi_{n+1})} \quad (4.58)$$

$$K_{\sim i}^{-1} = K_{\sim 0}^{-1} + \sum_{n=0}^{i-1} A_{\sim n} - \sum_{n=0}^{i-1} \frac{Z_{\sim n} Z_{\sim n}^T}{Z_{\sim n}^T (\psi_n - \psi_{n+1})} . \quad (4.59)$$

Considerando a equação (4.33) vem

$$\Delta U_{\sim i} = K_{\sim 0}^{-1} \psi_i + \sum_{n=0}^{i-1} A_{\sim n} \psi_i - \sum_{n=0}^{i-1} \frac{Z_{\sim n} Z_{\sim n}^T \psi_i}{Z_{\sim n}^T (\psi_n - \psi_{n+1})} \quad (4.60)$$

De forma análoga a (4.60) pode-se escrever

$$\Delta U_{\sim i+1} = K_{\sim 0}^{-1} \psi_{i+1} + \sum_{n=0}^{i-1} A_{\sim n} \psi_{i+1} - \sum_{n=0}^{i-1} \frac{Z_{\sim n} Z_{\sim n}^T \psi_{i+1}}{Z_{\sim n}^T (\psi_n - \psi_{n+1})} + A_{\sim i} \psi_{i+1} - \frac{Z_{\sim i} Z_{\sim i}^T \psi_{i+1}}{Z_{\sim i}^T (\psi_i - \psi_{i+1})} \quad (4.61)$$

Considerando as equações (4.58), (4.60), e (4.61) verifica-se que

$$\Delta U_{\sim i} - \Delta U_{\sim i+1} = Z_{\sim i} + \frac{Z_{\sim i} Z_{\sim i}^T \psi_{i+1}}{Z_{\sim i}^T (\psi_i - \psi_{i+1})} - A_{\sim i} \psi_{i+1} \quad (4.62)$$

de onde vem a relação

$$\Delta U_{\sim i+1} = \Delta U_{\sim i} + A_{\sim i} \psi_{i+1} - \frac{Z_{\sim i} Z_{\sim i}^T \psi_i}{Z_{\sim i}^T (\psi_i - \psi_{i+1})} \quad (4.63)$$

Definindo

$$\alpha_n = \Delta U_{\sim n}^T (\psi_n - \psi_{n+1}) \quad (4.64)$$

$$\beta_n = Z_n^T (\psi_n - \psi_{n+1}) \quad (4.65)$$

é fácil ver pelas equações (4.55) e (4.58) que

$$Z_i = K_{\sim 0}^{-1} (\psi_i - \psi_{i+1}) + \sum_{n=0}^{i-1} \frac{\Delta U_{\sim n} \Delta U_{\sim n}^T (\psi_i - \psi_{i+1})}{\alpha_n} - \sum_{n=0}^{i-1} \frac{Z_n Z_n^T (\psi_i - \psi_{i+1})}{\beta_n} \quad (4.66)$$

As equações (4.57), (4.63) e (4.66) são empregadas nos ciclos iterativos do método. Deve-se observar que é necessário ir armazenando os escalares α_n e β_n e os vetores $\Delta U_{\sim n}$ e Z_n durante o processo. A síntese do algoritmo é apresentada a seguir.

- Síntese do algoritmo

$$i = 0$$

$$\begin{array}{l} \rightarrow \psi_i = \psi_0 \quad ; \quad U_i = U_0 \\ (2) \quad \psi_i = K_{\sim 0} \Delta U_{\sim i} \Rightarrow \Delta U_{\sim i} \Rightarrow \text{armazenar } \Delta U_{\sim i} \end{array}$$

$$\Delta \bar{U}_{\sim i} = \Delta U_{\sim i}$$

$$U_{\sim i+1} = U_i + \Delta U_{\sim i}$$

$$\text{com } U_{\sim i+1} \Rightarrow \psi_{i+1}$$

$$\psi_{i+1} = K_{\sim 0} \Delta \bar{U}_{\sim i+1} \Rightarrow \Delta \bar{U}_{\sim i+1}$$

$$Z_i = \Delta \bar{U}_{\sim i} - \Delta \bar{U}_{\sim i+1} \Rightarrow \text{armazenar}$$

$$\beta_i = Z_i^T (\psi_i - \psi_{i+1}) \Rightarrow \text{armazenar}$$

$$\alpha_i = \Delta U_{\sim i}^T (\psi_i - \psi_{i+1}) \Rightarrow \text{armazenar}$$

$$\begin{array}{l} \rightarrow i = 1 \Rightarrow \text{imax.} \\ (1) \end{array}$$

$$\Delta U_{\sim i} = \Delta U_{\sim i} + \frac{\Delta U_{\sim i} \Delta U_{\sim i}^T \psi_{i+1}}{\alpha_i} - \frac{Z_i Z_i^T \psi_i}{\beta_i} \Rightarrow \text{armazenar}$$

$$U_i = U_{i+1} \quad ; \quad \Delta \bar{U}_{\sim i} = \Delta \bar{U}_{\sim i+1} \quad ; \quad \psi_i = \psi_{i+1}$$

$$\underline{U}_{i+1} = \underline{U}_i + \Delta \underline{U}_i$$

$$\text{com } \underline{U}_{i+1} \Rightarrow \underline{\psi}_{i+1}$$

$$\text{se } \frac{\|\underline{\psi}_{i+1}\|}{\|\Delta \underline{P}\|} \leq \text{tol} \quad \text{e} \quad \frac{\|\Delta \underline{U}_i\|}{\|\underline{U}_{i+1}\|} \leq \text{tol} \Rightarrow \underline{U}_{i+1} \text{ é a solução}$$

$$\text{se não, } \underline{\psi}_{i+1} = \underline{K}_{i+1}^{-1} \Delta \underline{U}_{i+1} \Rightarrow \Delta \underline{U}_{i+1}$$

$$\underline{Z}_i = (\Delta \underline{U}_i - \Delta \underline{U}_{i+1}) + \sum_{n=0}^{i-1} \frac{\Delta \underline{U}_n \Delta \underline{U}_n^T (\underline{\psi}_i - \underline{\psi}_{i+1})}{\alpha_n} - \sum_{n=0}^{i-1} \frac{\underline{Z}_n \underline{Z}_n^T (\underline{\psi}_i - \underline{\psi}_{i+1})}{\beta_n} \Rightarrow \text{armazenar}$$

$$\beta_i = \underline{Z}_i^T (\underline{\psi}_i - \underline{\psi}_{i+1}) \Rightarrow \text{armazenar}$$

$$\alpha_i = \Delta \underline{U}_i^T (\underline{\psi}_i - \underline{\psi}_{i+1}) \Rightarrow \text{armazenar}$$

voltar para (1)

Se não convergiu em imax iterações,

$$\underline{\psi}_0 = \underline{\psi}_{i+1}$$

$$\underline{U}_0 = \underline{U}_{i+1}$$

voltar para (2).

c) Algoritmos de Pearson

c.1) Algoritmo no.1

A equação (4.23) pode ser escrita na forma

$$\underline{K}_i^{-1} = \underline{K}_{i-1}^{-1} + \frac{\underline{Z}_{i-1} \Delta \underline{U}_{i-1}^T}{\Delta \underline{U}_{i-1}^T (\underline{\psi}_{i-1} - \underline{\psi}_i)} \quad (4.67)$$

onde

$$\tilde{z}_{i-1} = \Delta \tilde{u}_{i-1} - \tilde{k}_{i-1}^{-1} (\psi_{i-1} - \psi_i) \quad (4.68)$$

Procedendo como anteriormente chega-se a

$$\tilde{z}_0 = \Delta \tilde{u}_0 - \tilde{k}_0^{-1} (\psi_0 - \psi_1) \quad (4.69)$$

$$\tilde{z}_i = \Delta \tilde{u}_i - \tilde{k}_0^{-1} (\psi_i - \psi_{i+1}) - \sum_{n=0}^{i-1} \frac{\tilde{z}_n \Delta \tilde{u}_n^T (\psi_i - \psi_{i+1})}{\alpha_n} \quad (4.70)$$

$$\Delta \tilde{u}_{i+1} = \frac{\tilde{z}_i \Delta \tilde{u}_i^T \psi_i}{\alpha_i} \quad (4.71)$$

sendo

$$\alpha_n = \Delta \tilde{u}_n^T (\psi_n - \psi_{n+1}) \quad (4.72)$$

e

$$\alpha_i = \Delta \tilde{u}_i^T (\psi_i - \psi_{i+1}) \quad (4.73)$$

Aqui é necessário armazenar os coeficientes α_n e os vetores \tilde{z}_n e $\Delta \tilde{u}_n$.

c.2) Algoritmo no.2

A equação (4.24) pode ser escrita como

$$\tilde{k}_i^{-1} = \tilde{k}_{i-1}^{-1} + \frac{(\Delta \tilde{u}_{i-1} - \tilde{z}_{i-1}) \tilde{z}_{i-1}^T}{\tilde{z}_{i-1}^T (\psi_{i-1} - \psi_i)} \quad (4.74)$$

onde

$$\tilde{z}_{i-1} = \tilde{k}_{i-1}^{-1} (\psi_{i-1} - \psi_i) \quad (4.75)$$

Procedendo da forma usual chega-se a

$$\tilde{z}_0 = K_{\tilde{z}_0}^{-1} (\psi_0 - \psi_1) \quad (4.76)$$

$$\tilde{z}_i = K_{\tilde{z}_0}^{-1} (\psi_i - \psi_{i+1}) + \sum_{n=0}^{i-1} \frac{(\Delta U_{\tilde{z}_n} - \tilde{z}_n) Z_{\tilde{z}_n}^T (\psi_i - \psi_{i+1})}{\beta_n} \quad (4.77)$$

$$\Delta U_{\tilde{z}_{i+1}} = \frac{(\Delta U_{\tilde{z}_i} - \tilde{z}_i) Z_{\tilde{z}_i}^T \psi_i}{\beta_i} \quad (4.78)$$

$$\beta_n = Z_{\tilde{z}_n}^T (\psi_n - \psi_{n+1}) \quad (4.79)$$

Aqui é necessário armazenar os coeficientes β_n e os vetores $Z_{\tilde{z}_n}$ e $\Delta U_{\tilde{z}_n}$.

Síntese dos algoritmos

A) Pearson no. 1

$$i = 0$$

$$\begin{array}{l} \psi_i = \psi_0 \quad ; \quad U_i = U_0 \\ (2) \quad \psi_i = K_{\tilde{z}_0} \Delta U_{\tilde{z}_i} \Rightarrow \Delta U_{\tilde{z}_i} \Rightarrow \text{armazenar } \Delta U_{\tilde{z}_i} \end{array}$$

$$\Delta \bar{U}_{\tilde{z}_i} = \Delta U_{\tilde{z}_i}$$

$$U_{\tilde{z}_{i+1}} = U_i + \Delta U_{\tilde{z}_i}$$

$$\text{com } U_{\tilde{z}_{i+1}} \Rightarrow \psi_{i+1}$$

$$\psi_{i+1} = K_{\tilde{z}_0} \Delta \bar{U}_{\tilde{z}_{i+1}} \Rightarrow \Delta \bar{U}_{\tilde{z}_{i+1}}$$

$$Z_{\tilde{z}_i} = \Delta U_{\tilde{z}_i} - (\Delta \bar{U}_{\tilde{z}_i} - \Delta \bar{U}_{\tilde{z}_{i+1}}) \Rightarrow \text{armazenar}$$

$$\alpha_i = \Delta U_{\tilde{z}_i}^T (\psi_i - \psi_{i+1}) \Rightarrow \text{armazenar}$$

$$(1) \quad i = 1 \Rightarrow i_{\max.}$$

$$\Delta \underline{U}_{\sim i} = \frac{\underline{Z}_{\sim i} \Delta \underline{U}_{\sim i}^T \underline{\psi}_i}{\alpha_i} \Rightarrow \text{armazenar } \Delta \underline{U}_{\sim i}$$

$$\underline{U}_{\sim i} = \underline{U}_{\sim i+1} ; \quad \Delta \bar{\underline{U}}_{\sim i} = \Delta \bar{\underline{U}}_{\sim i+1} ; \quad \underline{\psi}_i = \underline{\psi}_{i+1}$$

$$\underline{U}_{\sim i+1} = \underline{U}_{\sim i} + \Delta \underline{U}_{\sim i}$$

$$\text{com } \underline{U}_{\sim i+1} \Rightarrow \underline{\psi}_{i+1}$$

$$\text{se } \frac{\|\underline{\psi}_{i+1}\|}{\|\Delta \underline{F}_{\sim}\|} \leq \text{tol} \quad \text{e} \quad \frac{\|\Delta \underline{U}_{\sim i}\|}{\|\underline{U}_{\sim i+1}\|} \leq \text{tol} \Rightarrow \underline{U}_{\sim i+1} \text{ é a solução}$$

$$\text{se não, } \underline{\psi}_{i+1} = K_{\sim 0} \Delta \bar{\underline{U}}_{\sim i+1} \Rightarrow \Delta \bar{\underline{U}}_{\sim i+1}$$

$$\underline{Z}_{\sim i} = \Delta \underline{U}_{\sim i} - (\Delta \bar{\underline{U}}_{\sim i} - \Delta \bar{\underline{U}}_{\sim i+1}) - \sum_{n=0}^{i-1} \frac{\underline{Z}_{\sim n} \Delta \underline{U}_{\sim n}^T (\underline{\psi}_i - \underline{\psi}_{i+1})}{\alpha_n} \Rightarrow \text{armazenar}$$

$$\alpha_i = \Delta \underline{U}_{\sim i}^T (\underline{\psi}_i - \underline{\psi}_{i+1}) \Rightarrow \text{armazenar}$$

voltar para (1)

Se não convergiu em i_{\max} iterações,

$$\underline{\psi}_0 = \underline{\psi}_{i+1}$$

$$\underline{U}_{\sim 0} = \underline{U}_{\sim i+1}$$

voltar para (2).

B) Pearson no. 2

$$i = 0$$

$$(2) \quad \underline{\psi}_i = \underline{\psi}_0 ; \quad \underline{U}_{\sim i} = \underline{U}_{\sim 0}$$

$$\underline{\psi}_i = K_{\sim 0} \Delta \underline{U}_{\sim i} \Rightarrow \Delta \underline{U}_{\sim i} \Rightarrow \text{armazenar } \Delta \underline{U}_{\sim i}$$

$$\Delta \bar{\underline{U}}_{\sim i} = \Delta \underline{U}_{\sim i}$$

$$U_{\sim i+1} = U_{\sim i} + \Delta U_{\sim i}$$

$$\text{com } U_{\sim i+1} \Rightarrow \psi_{i+1}$$

$$\psi_{i+1} = K_{\sim 0} \Delta \bar{U}_{\sim i+1} \Rightarrow \Delta \bar{U}_{\sim i+1}$$

$$Z_{\sim i} = \Delta \bar{U}_{\sim i} - \Delta \bar{U}_{\sim i+1} \Rightarrow \text{armazenar}$$

$$\beta_i = Z_{\sim i}^T (\psi_i - \psi_{i+1}) \Rightarrow \text{armazenar}$$

$$(1) \quad i = 1 \Rightarrow \text{imax.}$$

$$\Delta U_{\sim i} = (\Delta U_{\sim i} - Z_{\sim i}) \frac{Z_{\sim i}^T \psi_i}{\beta_i} \Rightarrow \text{armazenar } \Delta U_{\sim i}$$

$$U_{\sim i} = U_{\sim i+1} \quad ; \quad \Delta \bar{U}_{\sim i} = \Delta \bar{U}_{\sim i+1} \quad ; \quad \psi_i = \psi_{i+1}$$

$$U_{\sim i+1} = U_{\sim i} + \Delta U_{\sim i}$$

$$\text{com } U_{\sim i+1} \Rightarrow \psi_{i+1}$$

$$\text{se } \frac{\|\psi_{i+1}\|}{\|\Delta F_{\sim}\|} \leq \text{tol} \quad \text{e} \quad \frac{\|\Delta U_{\sim i}\|}{\|U_{\sim i+1}\|} \leq \text{tol} \Rightarrow U_{\sim i+1} \text{ é a solução}$$

$$\text{se não, } \psi_{i+1} = K_{\sim 0} \Delta \bar{U}_{\sim i+1} \Rightarrow \Delta \bar{U}_{\sim i+1}$$

$$Z_{\sim i} = (\Delta \bar{U}_{\sim i} - \Delta \bar{U}_{\sim i+1}) + \sum_{n=0}^{i-1} \frac{(\Delta U_{\sim n} - Z_{\sim n}) Z_{\sim n}^T (\psi_i - \psi_{i+1})}{\beta_n} \Rightarrow \text{armazenar}$$

$$\beta_i = Z_{\sim i}^T (\psi_i - \psi_{i+1}) \Rightarrow \text{armazenar}$$

voltar para (1)

Se não convergiu em imax iterações,

$$\psi_0 = \psi_{i+1}$$

$$U_{\sim 0} = U_{\sim i+1}$$

voltar para (2).

D) Método BFGS

A equação (4.25) pode ser escrita na forma alternativa

$$\tilde{K}_i^{-1} = \left[\tilde{I} + \tilde{W}\tilde{V}^T \right] \tilde{K}_{i-1}^{-1} \left[\tilde{I} + \tilde{V}\tilde{W}^T \right] \quad (4.80)$$

onde

$$\tilde{W}_{i-1} = \tilde{W} = \frac{\Delta U_{i-1}}{\Delta U_{i-1}^T \chi_{i-1}} \quad (4.81)$$

$$\tilde{V}_{i-1} = \tilde{V} = -\chi_{i-1} - \alpha \psi_{i-1} \quad (4.82)$$

sendo

$$\alpha_{i-1} = \alpha = \left(\frac{\Delta U_{i-1}^T \chi_{i-1}}{\Delta U_{i-1}^T \psi_{i-1}} \right)^{1/2} \quad (4.83)$$

- Demonstração

Introduzindo as expressões de \tilde{W} e \tilde{V} em (4.80) vem

$$\begin{aligned} \tilde{K}_i^{-1} &= \left[\tilde{I} - \frac{\Delta U_{i-1} \chi_{i-1}^T}{\Delta U_{i-1}^T \chi_{i-1}} \right] \tilde{K}_{i-1}^{-1} \left[\tilde{I} - \frac{\chi_{i-1} \Delta U_{i-1}^T}{\Delta U_{i-1}^T \chi_{i-1}} \right] + \alpha^2 \tilde{W} \tilde{V}^T \tilde{K}_{i-1}^{-1} \tilde{V} \tilde{W}^T + \\ &+ 2\alpha \left[-\tilde{K}_{i-1}^{-1} \tilde{V} \tilde{W}^T + \tilde{W} \chi_{i-1}^T \tilde{K}_{i-1}^{-1} \tilde{V} \tilde{W}^T \right] \end{aligned}$$

Para que esta expressão seja igual a (4.26) é necessário que

$$\alpha \tilde{\chi}_{i-1}^T \tilde{W}_{i-1} K_{i-1}^{-1} \tilde{\psi}_{i-1} \tilde{W}_{i-1}^T = \frac{\Delta U_{i-1} \Delta U_{i-1}^T}{\Delta U_{i-1}^T \tilde{\chi}_{i-1}} \quad (A)$$

e

$$\tilde{W}_{i-1} \tilde{\chi}_{i-1}^T K_{i-1}^{-1} \tilde{\psi}_{i-1} \tilde{W}_{i-1}^T - K_{i-1}^{-1} \tilde{\psi}_{i-1} \tilde{W}_{i-1}^T = \tilde{0} \quad (B)$$

Introduzindo as definições de \tilde{W} e α e lembrando que

$$K_{i-1}^{-1} \tilde{\psi}_{i-1} = \Delta U_{i-1}$$

pode-se demonstrar as relações (A) e (B) e, consequentemente, a equação (4.80).

Empregando a equação (4.80) vem:

$$K_{i-1}^{-1} = (I + \tilde{W}_{i-1} \tilde{V}_{i-1}^T) K_{i-1}^{-1} (I + \tilde{V}_{i-1} \tilde{W}_{i-1}^T) \quad (4.84)$$

$$K_{i-1}^{-1} = (I + \tilde{W}_{i-1} \tilde{V}_{i-1}^T) (I + \tilde{W}_{i-1} \tilde{V}_{i-1}^T) K_{i-1}^{-1} (I + \tilde{V}_{i-1} \tilde{W}_{i-1}^T) (I + \tilde{V}_{i-1} \tilde{W}_{i-1}^T) \quad (4.85)$$

e para uma iteração genérica i ,

$$\begin{aligned} K_i^{-1} &= (I + \tilde{W}_{i-1} \tilde{V}_{i-1}^T) \dots (I + \tilde{W}_{i-1} \tilde{V}_{i-1}^T) (I + \tilde{W}_{i-1} \tilde{V}_{i-1}^T) K_{i-1}^{-1} (I + \tilde{V}_{i-1} \tilde{W}_{i-1}^T) (I + \tilde{V}_{i-1} \tilde{W}_{i-1}^T) \dots \\ &\dots (I + \tilde{V}_{i-1} \tilde{W}_{i-1}^T) \end{aligned} \quad (4.86)$$

O incremento nos deslocamentos ΔU_i é dado por

$$\Delta U_i = K_i^{-1} \tilde{\psi}_i \quad (4.87)$$

Para avaliar ΔU_i procede-se como a seguir:

$$1 - b_{i-1} = (I + \tilde{V}_{i-1} \tilde{W}_{i-1}^T) \tilde{\psi}_i$$

$$\tilde{b}_0 = \tilde{\psi}_i + \tilde{v}_{i-1} (W_{i-1}^T \tilde{\psi}_i)$$

$$2 - \tilde{b}_1 = (I + \tilde{v}_{i-2} W_{i-2}^T) \tilde{b}_0$$

$$\tilde{b}_1 = \tilde{b}_0 + \tilde{v}_{i-2} (W_{i-2}^T \tilde{b}_0)$$

$$\vdots$$

$$i - \tilde{b}_{i-1} = (I + \tilde{v}_0 W_0^T) \tilde{b}_{i-2}$$

$$\tilde{b}_{i-1} = \tilde{b}_{i-2} + \tilde{v}_0 (W_0^T \tilde{b}_{i-2})$$

Resolve-se:

$$\tilde{b}_{i-1} = K_{00} C_0 \rightarrow \text{substituições} \rightarrow C_0$$

$$1 - C_1 = (I + W_0 V_0^T) C_0$$

$$C_1 = C_0 + W_0 (V_0^T C_0)$$

$$2 - C_2 = (I + W_1 V_1^T) C_1$$

$$C_2 = C_1 + W_1 (V_1^T C_1)$$

$$\vdots$$

$$i - C_i = (I + W_{i-1} V_{i-1}^T) C_{i-1}$$

$$C_i = C_{i-1} + W_{i-1} (V_{i-1}^T C_{i-1})$$

$$e \Delta U_i = C_i$$

Síntese do Algoritmo

$$i = 1$$

$$\rightarrow \tilde{\psi}_i = \tilde{\psi}_0 ; U_i = U_0$$

$$(2) \tilde{\psi}_i = K_{00} \Delta U_i \Rightarrow \Delta U_i$$

$$U_{i+1} = U_i + \Delta U_i$$

$$\text{com } U_{i+1} \Rightarrow \tilde{\psi}_{i+1}$$

montagem dos vetores \tilde{v}_i e W_i :

$$\beta = \Delta \underline{U}_i^T (\underline{\psi}_i - \underline{\psi}_{i+1})$$

$$\lambda = \Delta \underline{U}_i^T \underline{\psi}_i$$

se $\frac{\beta}{\lambda} < 0 \rightarrow$ vai para (3)

$$\alpha = \sqrt{\beta/\lambda}$$

$$\underline{V}_i = -(\underline{\psi}_i - \underline{\psi}_{i+1}) - \alpha \underline{\psi}_i \rightarrow \text{armazenar}$$

$$\underline{W}_i = \Delta \underline{U}_i / \beta \rightarrow \text{armazenar}$$

$$\left[\begin{array}{l} \rightarrow i = \underline{Z} \Rightarrow \text{imax} \\ (1) \quad \underline{\psi}_i = \underline{\psi}_{i+1} \quad ; \quad \underline{U}_i = \underline{U}_{i+1} \end{array} \right.$$

$$\left[\begin{array}{l} \rightarrow j = i-1 \Rightarrow 1 \\ \underline{\psi}_{i+1} = \underline{\psi}_{i+1} + \underline{V}_j (\underline{W}_j^T \underline{\psi}_{i+1}) \\ \text{voltar} \end{array} \right.$$

$$\underline{\psi}_{i+1} = \underline{K}_0 \Delta \underline{U}_i \Rightarrow \Delta \underline{U}_i$$

$$\left[\begin{array}{l} \rightarrow j = 1 \Rightarrow i-1 \\ \Delta \underline{U}_i = \Delta \underline{U}_i + \underline{W}_j (\underline{V}_j^T \Delta \underline{U}_i) \\ \text{voltar} \end{array} \right.$$

$$\underline{U}_{i+1} = \underline{U}_i + \Delta \underline{U}_i$$

$$\text{com } \underline{U}_{i+1} \Rightarrow \underline{\psi}_{i+1}$$

$$\text{se } \frac{\|\underline{\psi}_{i+1}\|}{\|\Delta \underline{P}\|} \leq \text{tol} \quad \text{e} \quad \frac{\|\Delta \underline{U}_i\|}{\|\underline{U}_{i+1}\|} \leq \text{tol} \Rightarrow \underline{U}_{i+1} \text{ é a solução}$$

se não

montagem dos vetores \underline{V}_i e \underline{W}_i

$$\beta = \Delta \underline{U}_i^T (\underline{\psi}_i - \underline{\psi}_{i+1})$$

$$\lambda = \Delta \underline{U}_i^T \underline{\psi}_i$$

se $\beta/\lambda < 0 \Rightarrow$ vai para (3)

$$\alpha = \sqrt{\beta/\lambda}$$

$$\underline{V}_i = -(\underline{\psi}_i - \underline{\psi}_{i+1}) - \alpha \underline{\psi}_i \rightarrow \text{armazenar}$$

$$W_i = \Delta U_i / \beta \Rightarrow \text{armazenar}$$

voltar para (1)

se não convergiu em i_{\max} iterações

$$\left\{ \begin{array}{l} \Psi_0 = \Psi_{i+1} \\ U_0 = U_{i+1} \end{array} \right.$$

(3) voltar para 2

4.4 Exemplos Numéricos

Nesta seção são apresentados os resultados obtidos empregando-se os métodos de minimização em um problema estrutural. Apenas os métodos de segunda ordem serão testados. A estrutura analisada é uma viga de concreto armado (viga A1) ensaiada por Bresler e Scordelis. O concreto é modelado com o emprego de elementos isoparamétricos quadráticos da família Serendipity para estado plano de tensões. As integrações numéricas necessárias para a montagem da matriz inicial e para a avaliação das ações nodais não-lineares são efetuadas usando-se 3x3 pontos de Gauss. A armadura transversal é incluída no elemento de concreto e as armaduras longitudinais são tratadas como segmentos de barra.

Os modelos constitutivos para os materiais são os apresentados por Vecchio. A carga de ruína obtida no ensaio foi aplicada em 10 incrementos iguais na análise. A matriz de rigidez inicial é armazenada em um arranjo retangular e decomposta pelo método de Cholesky. Foram analisadas as 3 malhas da figura 4.1.

Nas tabelas 4.1, 4.2 e 4.3 apresentam-se os números de iterações necessárias para a obtenção da resposta completa da estrutura com os métodos analisados.

Observa-se pelas tabelas que apenas o método de Broyden apresenta dificuldades de convergência (tabela 4.3). Verifica-se, também, que o método de Newton-Raphson modificado exige um número muito elevado de iterações para a convergência. Os menores números de iterações são obtidos

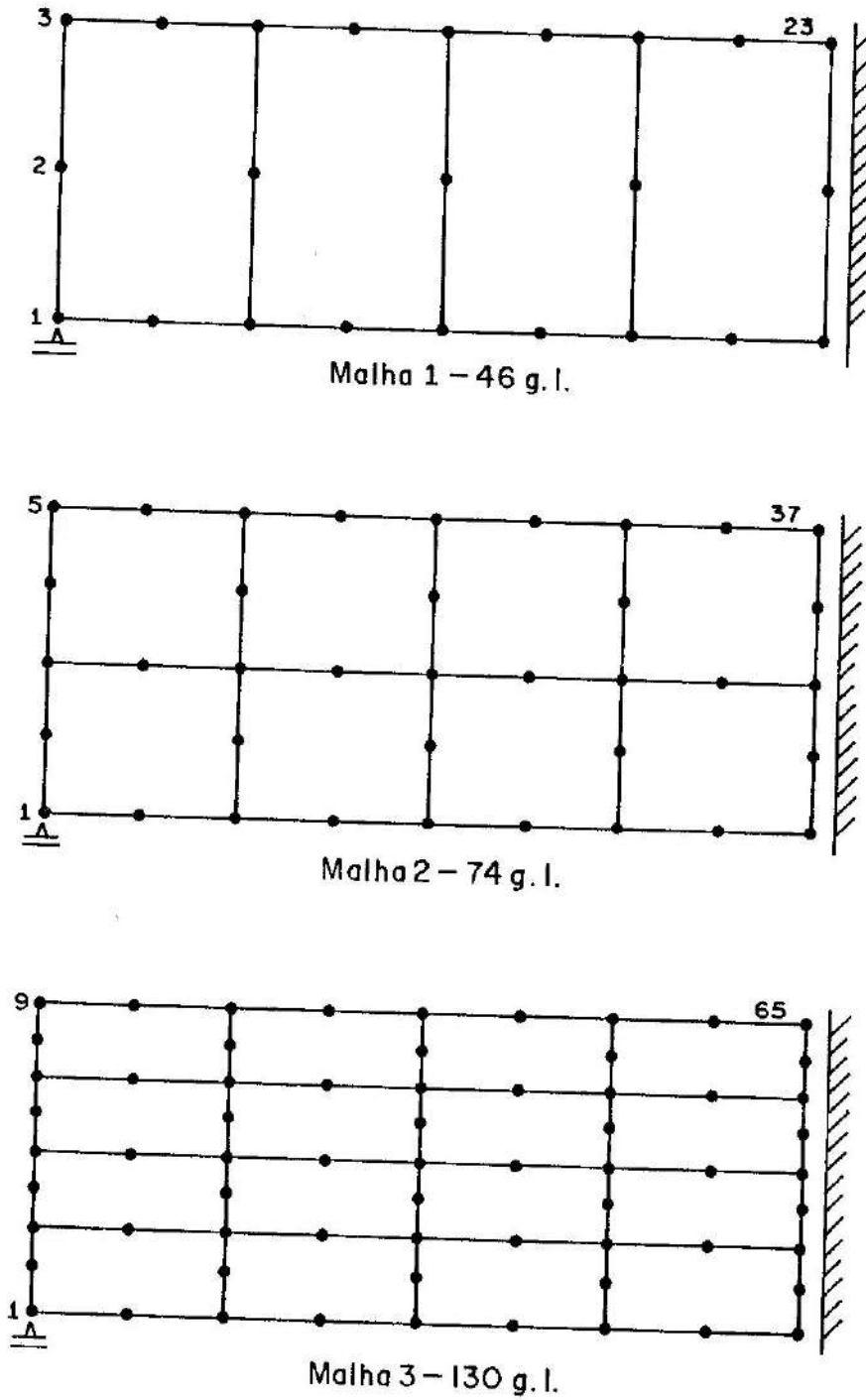


Figura 4.1 - Malhas de elementos finitos

com os algoritmos de Pearson, BFGS, Secante-Newton e minimização bidirecional.

Na tabela 4.4 são apresentados os tempos de processamento relativos obtidos nos exemplos. Para uma melhor visualização, os resultados são apresentados nas figuras 4.2 a 4.4.

Tabela 4.1 - Número de iterações: malha 1

etapa	Método							
	1	2	3	4	5	6	7	8
1	2	2	2	2	2	2	2	2
2	13	6	9	7	7	8	8	7
3	15	6	8	6	7	7	7	7
4	15	6	9	6	7	7	6	7
5	44	13	18	18	20	18	16	16
6	39	12	17	12	12	14	13	14
7	45	13	19	14	15	14	17	15
8	49	12	18	14	14	15	15	14
9	52	24	22	15	15	17	28	16
10	55	16	22	16	17	17	21	16
TOTAL	329	110	144	110	116	119	133	114

- 1 - Newton-Raphson modificado
- 2 - Broyden
- 3 - Davidon-Fletcher-Powell
- 4 - Pearson no.1
- 5 - Pearson no.2
- 6 - BFGS
- 7 - Secante-Newton
- 8 - Minimização bidirecional

Tabela 4.2 - Número de iterações: malha 2

etapa	M é t o d o							
	1	2	3	4	5	6	7	8
1	2	2	2	2	2	2	2	2
2	8	5	6	6	6	6	6	6
3	18	8	12	8	12	12	8	8
4	24	14	36	43	23	11	11	12
5	28	10	14	10	12	12	12	11
6	34	13	18	12	13	14	16	13
7	39	13	30	13	14	16	16	20
8	45	15	22	14	14	16	17	17
9	53	15	24	16	18	18	23	18
10	63	17	25	18	19	19	23	20
TOTAL	314	111	189	142	133	126	134	127

Tabela 4.3 - Número de iterações: malha 3

etapa	M é t o d o							
	1	2	3	4	5	6	7	8
1	2	2	2	2	2	2	2	2
2	10	6	7	7	7	7	8	7
3	19	13	12	9	9	10	10	10
4	24	9	12	9	10	10	9	10
5	29	10	15	12	12	12	13	12
6	33	13	19	12	12	14	13	14
7	38	13	19	13	14	15	17	15
8	49	16	23	15	16	16	27	18
9	60	NC	27	18	19	19	21	19
10	75	-	36	20	20	25	30	22
TOTAL	339	-	172	117	121	130	150	129

Tabela 4.4 - Tempo de Processamento relativo

método	Malha 1 46 gl	Malha 2 74 gl	Malha 3 130 gl
1	2,77	3,98	7,31
2	1,00	1,52	N C
3	1,30	2,54	3,96
4	1,00	1,91	2,72
5	1,05	1,80	2,81
6	1,08	1,72	3,02
7	1,18	1,78	3,37
8	1,02	1,69	2,93

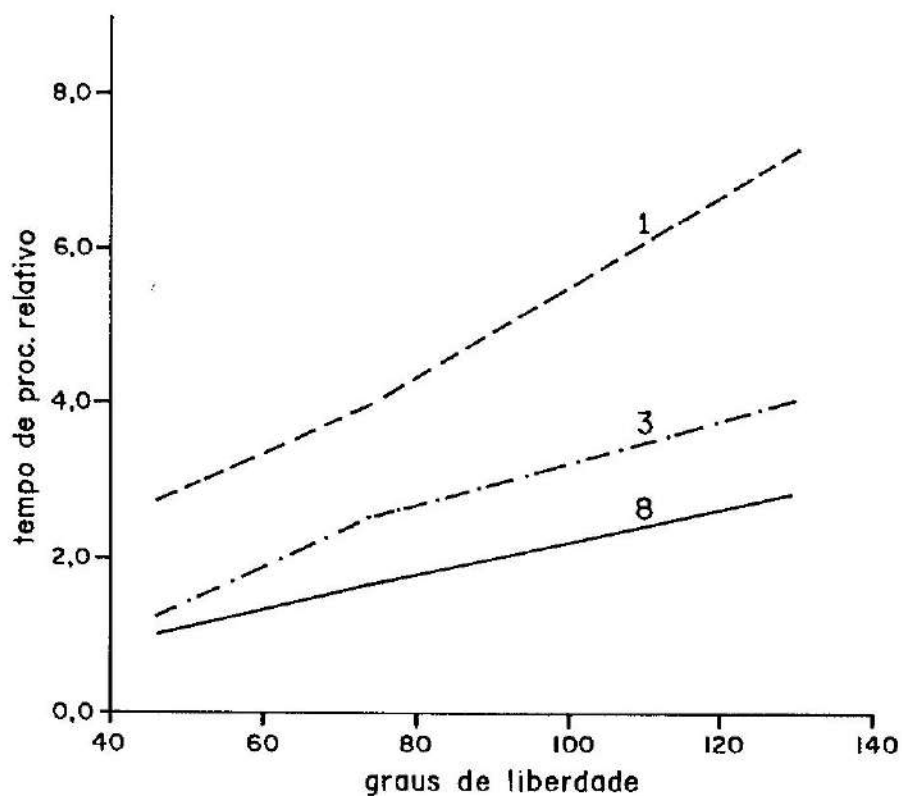


Figura 4.2 - Relação tempo de processamento x graus de liberdade

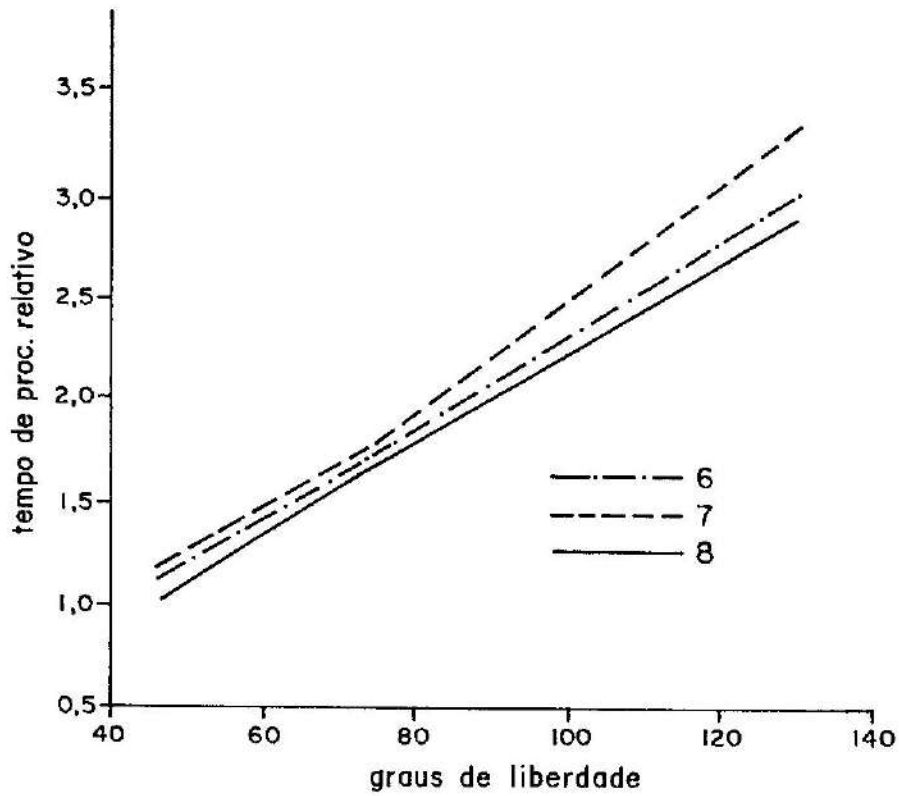


Figura 4.3 - Relação tempo de processamento x graus de liberdade

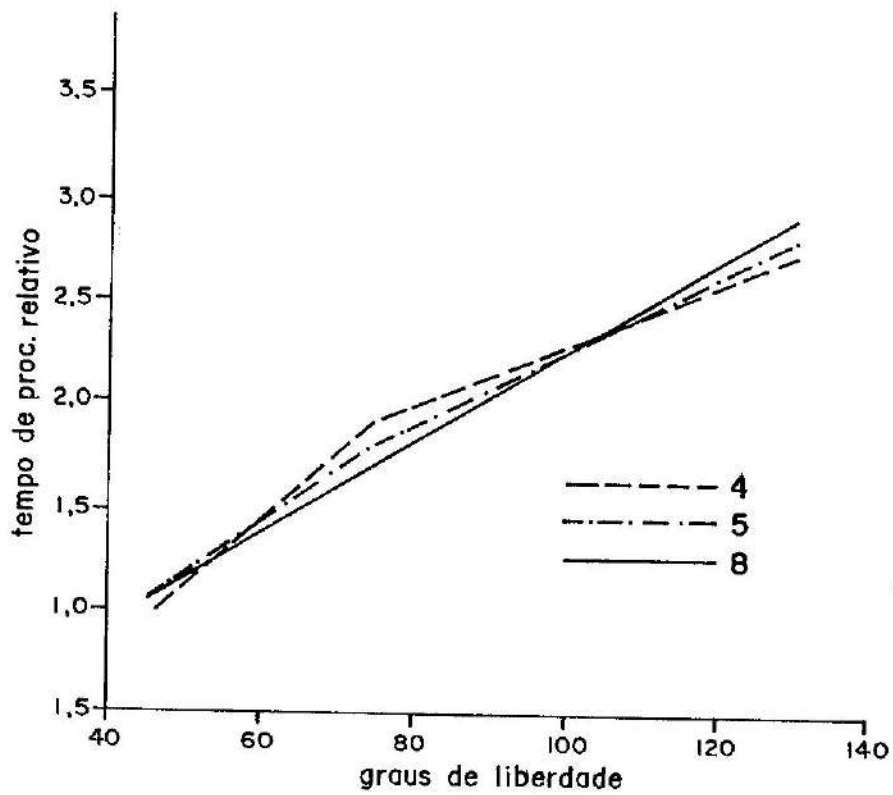


Figura 4.4 - Relação tempo de processamento x graus de liberdade

Observando as figuras conclui-se que os algoritmos de Pearson, o método BFGS e o algoritmo de minimização bidirecional apresentaram um melhor comportamento.

Deve ser lembrado, entretanto, que os métodos Secante-Newton e minimização bidirecional incluem um componente empírico nas suas formulações.

Dessa maneira é de se esperar que esses métodos apresentem um certo nível de instabilidade numérica para algum problema em particular.

Os algoritmos de Pearson apresentaram uma oscilação na relação tempo de processamento x número de graus de liberdade conforme indicado na figura 4.2. Isto pode ser um indicativo de que esses métodos têm a tendência de se tornarem instáveis.

O método BFGS, por outro lado, apresentou uma relação estável entre o tempo de processamento e o número de graus de liberdade (aproximadamente linear) assim como a minimização bidirecional. O ponto favorável ao BFGS é que o mesmo não apresenta componente empírico tendo como único inconveniente o fato de ocupar mais memória de computador. Entretanto, com o esquema de armazenamento apresentado, isto não se torna um fator limitante de forma que esse método parece ser o mais indicado para uma análise não-linear geral.

BIBLIOGRAFIA

1. ARAÚJO, J. M., Um algoritmo iterativo para análise não-linear de estruturas. XI COBEM, Campinas, 1991.
2. BRESLER, B. & SCORDELIS, A. C. Shear strength of reinforced concrete beams. ACI Journal, vol. 60, pp. 51-74, 1963.
3. CRISFIELD, M. A. Accelerated solution techniques and concrete cracking. Computer Methods in Applied Mechanics and Engineering, vol. 33, pp. 585-607, 1982.
4. FOX, Richard L. Optimization methods for engineering design. Addison-Wesley Publishing Company, 1971.
5. FRITZSCHE, Helmut. Programação não-linear - Análise e métodos. Editora Edgard Blücher Ltda, São Paulo, 1978.
6. GERADIN, M.; IDELSOHN, S. and HOGGE, M. "Computational strategies for the solution of large nonlinear problems via quasi-Newton methods". Computers and Structures, Vol. 13, pp. 73-81, 1981.
7. HIMMELBLAN, David M. - "Applied nonlinear programming". Mc Graw-Hill, 1972.
8. MATTHIES, Hermann and STRANG, Gilbert. "The solution of nonlinear finite element equations". International

- Journal for Numerical Methods in Engineering, vol. 14, pp. 1613-1626, 1979.
9. VECCHIO, F. - "Nonlinear finite element analysis of reinforced concrete membranes", ACI Journal, Vol. 86, pp. 26-35, 1989.
10. ZIENKIEWICZ, O.C. - "The finite element method", Mc Graw Hill, 3a. Ed., 1977.

ANEXO: Listagens dos programas em FORTRAN

```

SUBROUTINE ITERATIVO
IMPLICIT REAL*8(A-H,O-Z)
DIMENSION U1(200),U2(200),DESF(200)
COMMON /A1/ INN(30),IID(30),AVD(30)
COMMON /A2/ GK(200,50)
COMMON /A4/ F(200),FTOT(200)
COMMON /A6/ DF(200),DFTOT(200)
COMMON /A9/ DU2(200)
COMMON /B1/ N,NGL,NNOS
COMMON /C1/ ANL(200)
SOLUCAO DE SISTEMAS DE EQUACOES NAO-LINEARES
METODO : NEWTON-RAPHSON MODIFICADO
GK : matriz de rigidez global
FTOT : vetor de forcas ; F : idem, com contorno
DF, DFTOT : incrementos de forcas
N : numero de elementos
NGL : numero de graus de liberdade por no
NNOS : numero de nos da estrutura
ANL : vetor de acoes nodais nao-lineares
INN : vetor com os numeros dos nos com deslocamentos
      prescritos
IID : indica o numero do grau de liberdade prescrito
AVD : indica o valor do deslocamento prescrito
NTCA : numero de etapas de carga
*****
L=NNOS*NGL
DO 10 I=1,L
DF(I)=F(I)/NTCA
DFTOT(I)=FTOT(I)/NTCA
10 CONTINUE
DO 20 I=1,L
U1(I)=0
20 CONTINUE
DO 400 JEC=1,NTCA
DO 30 I=1,L
F(I)=DF(I)*JEC
FTOT(I)=DFTOT(I)*JEC
DU2(I)=DF(I)
30 CONTINUE
IF(JEC.GT.1) THEN
DO 40 I=1,L
U1(I)=U2(I)
40 CONTINUE
END IF
DO 200 NITER=1,LIM
Solucao do sistema : substituicoes com a matriz decomposta
Metodo de Cholesky
CALL RETRO(DU2,L,NB)
DO 80 I=1,L
U2(I)=U1(I)+DU2(I)
80 CONTINUE

```



```

*****
CHAMAR SUBROTINA PARA DETERMINAR O VETOR DE ACOES NODAIS
NAO-LINEARES, ANL, PARA OS DESLOCAMENTOS NODAIS U1
*****
DO 90 I=1,L
DESF(I)=FTOT(I)-ANI(I)
100 CONTINUE
***** VERIFICACAO DA CONVERGENCIA *****
A=0
B=0
C=0
D=0
Introducao das condicoes de contorno
DO 100 I=1,NDC
IL=NGL*(INN(I)-1)
M=IL+IID(I)
DESF(M)=AVD(I)
100 CONTINUE
DO 110 I=1,L
A=A+DESF(I)**2
B=B+DF(I)**2
C=C+DU2(I)**2
D=D+U2(I)**2
110 CONTINUE
A=DSQRT(A)
B=DSQRT(B)
C=DSQRT(C)
D=DSQRT(D)
ERRO1=A/B
ERRO2=C/D
IF((ERRO1.LE.0.01) .AND. (ERRO2.LE.0.01)) GO TO 1800
DO 120 I=1,L
U1(I)=U2(I)
DU2(I)=DESF(I)
120 CONTINUE
200 CONTINUE
**** Impressao dos resultados ****
1800 A=0
Imprimir : U2 (deslocamentos nodais), FTOT (acoes nodais)
400 CONTINUE
RETURN
END

```

SUBROUTINE ITERATIVO

IMPLICIT REAL*8(A-H,O-Z)

DIMENSION U0(150),U1(150)

DIMENSION BETA(10),ZMAT(150,10)

COMMON /A1/ INN(20),IID(20),AVD(20)

COMMON /A2/ GK(150,50)

COMMON /A4/ F(150),FTOT(150)

COMMON /A6/ DF(150),DFTOT(150)

COMMON /A9/ DU1(150),DU2(150),FN1(150),FN(150),UN1(150)

COMMON /B1/ N,NGL,NNOS

COMMON /C1/ ANL(150)

SOLUCAO DE SISTEMAS DE EQUACOES NAO-LINEARES

METODO DE BROYDEN

GK : matriz de rigidez global

FTOT : vetor de forcas ; F : idem, com contorno

DF, DFTOT : incrementos de forcas

N : numero de elementos

NGL : numero de graus de liberdade por no

NNOS : numero de nos da estrutura

ANL : vetor de acoes nodais nao-lineares

INN : vetor com os numeros dos nos com deslocamentos prescritos

IID : indica o numero do grau de liberdade prescrito

AVD : indica o valor do deslocamento prescrito

NTCA : numero de etapas de carga

L=NNOS*NGL

DO 10 I=1,L

DF(I)=F(I)/NTCA

DFTOT(I)=FTOT(I)/NTCA

U0(I)=0

10 CONTINUE

DO 400 JEC=1,NTCA

NITER=1

DO 20 I=1,L

F(I)=DF(I)*JEC

FTOT(I)=DFTOT(I)*JEC

FN(I)=DF(I)

20 CONTINUE

IF(JEC.GE.2) THEN

DO 30 I=1,L

U0(I)=U1(I)

30 CONTINUE

END IF

DO 40 I=1,L

DU1(I)=DF(I)

40 CONTINUE

Solucao do sistema : substituicoes com a matriz decomposta

Metodo de Cholesky

CALL RETRO(DU1,L,NB)

```

DO 50 I=1,L
U1(I)=U0(I)+DU1(I)
52 CONTINUE
DO 210 JITER=1,10
*****
C CHAMAR SUBROTINA PARA DETERMINAR O VETOR DE ACOES NODAIS
C NAO-LINEARES, ANL, PARA OS DESLOCAMENTOS NODAIS U1
C *****
DO 60 I=1,L
FN1(I)=FTOT(I)-ANL(I)
DU2(I)=FN1(I)
54 CONTINUE
C Introducao das condicoes de contorno
DO 70 I=1,NDC
IL=NGL*(INN(I)-1)
M=IL+IID(I)
FN1(M)=AVD(I)
DU2(M)=AVD(I)
70 CONTINUE
NITER=NITER+1
C Solucao do sistema
CALL RETRO(DU2,L,NB)
C MONTAGEM DO VETOR Z : Auxiliar do metodo
IND=1
DO 80 I=1,L
ZMAT(I,IND)=U1(I)-U0(I)-(DU1(I)-DU2(I))
82 CONTINUE
C Montagem do escalar BETA
AUX=0
DO 90 I=1,L
AUX=AUX+ZMAT(I,IND)*(FN(I)-FN1(I))
92 CONTINUE
BETA(IND)=AUX
DO 300 IND=2,10
AUX=0
DO 100 I=1,L
AUX=AUX+ZMAT(I,IND-1)*FN(I)
102 CONTINUE
DO 110 I=1,L
UN1(I)=ZMAT(I,IND-1)*AUX/BETA(IND-1)
112 CONTINUE
DO 120 I=1,L
U0(I)=U1(I)
U1(I)=U1(I)+UN1(I)
FN(I)=FN1(I)
DU1(I)=DU2(I)
122 CONTINUE
C *****
C CHAMAR SUBROTINA PARA DETERMINAR O VETOR DE ACOES NODAIS
C NAO-LINEARES, ANL, PARA OS DESLOCAMENTOS NODAIS U1
C *****

```

```

DO 130 I=1,L
FN1(I)=FTOT(I)-ANL(I)
DU2(I)=FN1(I)
130 CONTINUE
C Introducao das condicoes de contorno
DO 140 I=1,NDC
IL=NGL*(INN(I)-1)
M=IL+IID(I)
FN1(M)=AVD(I)
DU2(M)=AVD(I)
140 CONTINUE
C Verificacao da convergencia
A=0
B=0
C=0
D=0
DO 150 I=1,L
A=A+FN1(I)**2
B=B+DF(I)**2
C=C+UN1(I)**2
D=D+U1(I)**2
150 CONTINUE
ERRO1=DSQRT(A)/DSQRT(B)
ERRO2=DSQRT(C)/DSQRT(D)
IF((ERRO1.LT.0.01) .AND. (ERRO2.LT.0.01)) GO TO 1800
NITER=NITER+1
C Solucao do sistema
CALL RETRO(DU2,L,NB)
C MONTAGEM DO VETOR Z
DO 160 I=1,L
ZMAT(I,IND)=U1(I)-U0(I)-(DU1(I)-DU2(I))
160 CONTINUE
DO 180 NN=1,IND-1
AUX=0
DO 170 I=1,L
AUX=AUX+ZMAT(I,NN)*(FN(I)-FN1(I))
170 CONTINUE
DO 175 I=1,L
ZMAT(I,IND)=ZMAT(I,IND)-ZMAT(I,NN)*AUX/BETA(NN)
175 CONTINUE
180 CONTINUE
C Calculo do escalar beta
AUX=0
DO 190 I=1,L
AUX=AUX+ZMAT(I,IND)*(FN(I)-FN1(I))
190 CONTINUE
BETA(IND)=AUX
300 CONTINUE
DO 200 I=1,L
U0(I)=U1(I)
FN(I)=FN1(I)
DU1(I)=DU2(I)
U1(I)=U0(I)+DU2(I)
200 CONTINUE
210 CONTINUE

```

```
1 ***** Impressao dos resultados *****  
1300 A=0  
1 Imprimir : UI (deslocamentos nodais) ; FTOT (acoes nodais)  
420 CONTINUE  
RETURN  
END
```

SUBROUTINE ITERATIVO

IMPLICIT REAL*8(A-H,O-Z)

DIMENSION U0(150),U1(150)

DIMENSION BETA(10),ZMAT(150,10),UMAT(150,10),ALFA(10)

COMMON /A1/ INN(20),IID(20),AVD(20)

COMMON /A2/ GK(150,50)

COMMON /A4/ F(150),FTOT(150)

COMMON /A6/ DF(150),DFTOT(150)

COMMON /A9/ DU1(150),DU2(150),FN1(150),FN(150),UN1(150)

COMMON /B1/ N,NGL,NNOS

COMMON /C1/ ANL(150)

SOLUCAO DE SISTEMAS DE EQUACOES NAO-LINEARES

METODO DE DAVIDON-FLETCHER-POWELL

GK : matriz de rigidez global

FTOT : vetor de forcas ; F : idem, com contorno

DF, DFTOT : incrementos de forcas

N : numero de elementos

NGL : numero de graus de liberdade por no

NNOS : numero de nos da estrutura

ANL : vetor de acoes nodais nao-lineares

INN : vetor com os numeros dos nos com deslocamentos prescritos

IID : indica o numero do grau de liberdade prescrito

AVD : indica o valor do deslocamento prescrito

NTCA : numero de etapas de carga

L=NNOS*NGL

DO 10 I=1,L

DF(I)=F(I)/NTCA

DFTOT(I)=FTOT(I)/NTCA

U0(I)=0

10 CONTINUE

DO 400 JEC=1,NTCA

NITER=1

DO 20 I=1,L

F(I)=DF(I)*JEC

FTOT(I)=DFTOT(I)*JEC

FN(I)=DF(I)

20 CONTINUE

IF(JEC.GE.2) THEN

DO 30 I=1,L

U0(I)=U1(I)

30 CONTINUE

END IF

DO 40 I=1,L

DU1(I)=DF(I)

40 CONTINUE

Solucao do sistema : substituicoes com a matriz decomposta

Metodo de Cholesky

CALL RETRO(DU1,L,NB)

```

DO 50 I=1,L
U1(I)=U0(I)+DU1(I)
50 CONTINUE
DO 210 JITER=1,10
IND=1
*****
CHAMAR SUBROTINA PARA DETERMINAR O VETOR DE ACOES NODAIS
NAO-LINEARES, ANL, PARA OS DESLOCAMENTOS NODAIS U1
*****
DO 60 I=1,L
FN1(I)=FTOT(I)-ANL(I)
DU2(I)=FN1(I)
60 CONTINUE
Introducao das condicoes de contorno
DO 70 I=1,NDC
IL=NGL*(INN(I)-1)
M=IL+IID(I)
FN1(M)=AVD(I)
DU2(M)=AVD(I)
70 CONTINUE
Solucao do sistema
CALL RETRO(DU2,L,NB)
MONTAGEM DE ZMAT E UMAT : Vetores auxiliares do metodo
DO 80 I=1,L
ZMAT(I,IND)=DU1(I)-DU2(I)
UMAT(I,IND)=DU1(I)
80 CONTINUE
Montagem dos escalares BETA e ALFA : Auxiliares do metodo
AUX1=0
AUX2=0
DO 90 I=1,L
AUX1=AUX1+ZMAT(I,IND)*(FN(I)-FN1(I))
AUX2=AUX2+UMAT(I,IND)*(FN(I)-FN1(I))
90 CONTINUE
BETA(IND)=AUX1
ALFA(IND)=AUX2
DO 300 IND=2,10
AUX1=0
AUX2=0
DO 100 I=1,L
AUX1=AUX1+ZMAT(I,IND-1)*FN(I)
AUX2=AUX2+UMAT(I,IND-1)*FN1(I)
100 CONTINUE
DO 110 I=1,L
UN1(I)=UMAT(I,IND-1)+UMAT(I,IND-1)*AUX2/ALFA(IND-1)
*-ZMAT(I,IND-1)*AUX1/BETA(IND-1)
110 CONTINUE
DO 115 I=1,L
UMAT(I,IND)=UN1(I)
115 CONTINUE

```

```

DO 120 I=1,L
U0(I)=U1(I)
U1(I)=U1(I)+UN1(I)
FN(I)=FN1(I)
DU1(I)=DU2(I)
120 CONTINUE
*****
CHAMAR SUBROTINA PARA DETERMINAR O VETOR DE ACOES NODAIS
NAO-LINEARES, ANL, PARA OS DESLOCAMENTOS NODAIS U1
*****
Vetor de disequilibrio
DO 130 I=1,L
FN1(I)=FTOT(I)-ANL(I)
DU2(I)=FN1(I)
130 CONTINUE
Introducao das condicoes de contorno
DO 140 I=1,NBC
IL=NGL*(INN(I)-1)
H=IL+IIP(I)
FN1(H)=AVD(I)
DU2(H)=AVD(I)
140 CONTINUE
Verificacao da convergencia
A=0
B=0
C=0
D=0
DO 150 I=1,L
A=A+FN1(I)**2
B=B+DF(I)**2
C=C+UN1(I)**2
D=D+U1(I)**2
150 CONTINUE
ERRO1=DSQRT(A)/DSQRT(B)
ERRO2=DSQRT(C)/DSQRT(D)
IF((ERRO1.LT.0.01) .AND. (ERRO2.LT.0.01)) GO TO 1800
NITER=NITER+1
Solucao do sistema
CALL RETRO(DU2,L,NB)
MONTAGEM DO VETOR Z
DO 160 I=1,L
ZMAT(I,IND)=DU1(I)-DU2(I)
160 CONTINUE
DO 180 NN=1,IND-1
AUX1=0
AUX2=0
DO 170 I=1,L
AUX1=AUX1+ZMAT(I,NN)*(FN(I)-FN1(I))
AUX2=AUX2+UMAT(I,NN)*(FN(I)-FN1(I))
170 CONTINUE

```



```

      DO 175 I=1,L
      ZMAT(I,IND)=ZMAT(I,IND)-ZMAT(I,NN)*AUX1/BETA(NN)
      *+UMAT(I,NN)*AUX2/ALFA(NN)
175  CONTINUE
180  CONTINUE
C    Calculo dos escalares BETA e ALFA
      AUX1=0
      AUX2=0
      DO 190 I=1,L
      AUX1=AUX1+ZMAT(I,IND)*(FN(I)-FN1(I))
      AUX2=AUX2+UMAT(I,IND)*(FN(I)-FN1(I))
190  CONTINUE
      BETA(IND)=AUX1
      ALFA(IND)=AUX2
200  CONTINUE
      DO 200 I=1,L
      U0(I)=U1(I)
      FN(I)=FN1(I)
      DU1(I)=DU2(I)
      U1(I)=U0(I)+DU2(I)
200  CONTINUE
210  CONTINUE
C    ***** Impressao dos resultados *****
1800 a=0
C    Imprimir : U1 (deslocamentos nodais) ; FTOT (acoes nodais)
400  CONTINUE
      RETURN
      END

```

```

SUBROUTINE ITERATIVO
IMPLICIT REAL*8(A-H,O-Z)
DIMENSION U0(150),U1(150)
DIMENSION BETA(10),ZMAT(150,10),UMAT(150,10),ALFA(10)
COMMON /A1/ INN(20),IID(20),AVD(20)
COMMON /A2/ GK(150,50)
COMMON /A4/ F(150),FTOT(150)
COMMON /A6/ DF(150),DFTOT(150)
COMMON /A9/ DU1(150),DU2(150),FN1(150),FN(150),UN1(150)
COMMON /B1/ N,NGL,NNOS
COMMON /C1/ ANL(150)

```

```

SOLUCAO DE SISTEMAS DE EQUACOES NAO-LINEARES
METODO DE PEARSON NUMERO 1
GK : matriz de rigidez global
FTOT :: vetor de forcas; F: idem,com contorno
DF, DFTOT : incrementos de forcas
N : numero de elementos
NGL : numero de graus de liberdade ppor no
NNOS : numero de nos da estrutura
ANL : vetor de acoes nodais nao-lineares
INN : vetor com o numero dos nos com deslocamentos
      prescritos

```

```

IID : indica o numero do grau de liberdade prescrito
AVD : indica o valor dos deslocamentos prescritos
NTCA : numero de etapas de carga

```

```

*****
L=NNOS*NGL
DO 10 I=1,L
DF(I)=F(I)/NTCA
DFTOT(I)=FTOT(I)/NTCA
U0(I)=0

```

```

10 CONTINUE
DO 400 JEC=1,NTCA
NITER=1
DO 20 I=1,L
F(I)=DF(I)*JEC
FTOT(I)=DFTOT(I)*JEC
FN(I)=DF(I)

```

```

20 CONTINUE
IF(JEC.GE.2) THEN
DO 30 I=1,L
U0(I)=U1(I)

```

```

30 CONTINUE
END IF
DO 40 I=1,L
DU1(I)=DF(I)

```

```

40 CONTINUE
Solucao do sistema : substituicoes com a matriz decomposta
Metodo de Cholesky
CALL RETRO(DU1,L,NB)
DO 50 I=1,L
U1(I)=U0(I)+DU1(I)

```

```

50 CONTINUE

```

```

DO 210 JITER=1,10
IND=1
*****
CHAMAR SUBROTINA PARA DETERMINAR O VETOR DE ACOES NODAIS
NAO-LINEARES, ANL, PARA OS DESLOCAMENTOS NODAIS U1
*****
DO 60 I=1,L
FN1(I)=FTOT(I)-ANL(I)
DU2(I)=FN1(I)
80 CONTINUE
Introducao das condicoes de contorno
DO 70 I=1,NDC
IL=NGL*(INN(I)-1)
M=IL+IID(I)
FN1(M)=AVD(I)
DU2(M)=AVD(I)
70 CONTINUE
NITER=NITER+1
Solucão do sistema
CALL RETRO(DU2,L,NB)
MONTAGEM DE ZMAT E UMAT : Vetores auxiliares do metodo
DO 80 I=1,L
ZMAT(I,IND)=(DU1(I)-(DU1(I)-DU2(I)))
UMAT(I,IND)=DU1(I)
80 CONTINUE
Montagem do escalar ALFA
AUX=0
DO 90 I=1,L
AUX=AUX+UMAT(I,IND)*(FN(I)-FN1(I))
90 CONTINUE
ALFA(IND)=AUX
DO 300 IND=2,10
AUX=0
DO 100 I=1,L
AUX=AUX+UMAT(I,IND-1)*FN(I)
100 CONTINUE
DO 110 I=1,L
UN1(I)=ZMAT(I,IND-1)*AUX/ALFA(IND-1)
110 CONTINUE
DO 115 I=1,L
UMAT(I,IND)=UN1(I)
115 CONTINUE
DO 120 I=1,L
U0(I)=U1(I)
U1(I)=U1(I)+UN1(I)
FN(I)=FN1(I)
DU1(I)=DU2(I)
120 CONTINUE
*****
CHAMAR SUBROTINA PARA DETERMINAR O VETOR DE ACOES NODAIS
NAO-LINEARES, ANL, PARA OS DESLOCAMENTOS NODAIS U1
*****

```

```

DO 130 I=1,L
  FN1(I)=FTOT(I)-ANL(I)
  DU2(I)=FN1(I)
130 CONTINUE
  Introducao das condicoes de contorno
DO 140 I=1,NDC
  IL=NGL*(INN(I)-1)
  M=IL+IID(I)
  FN1(M)=AVD(I)
  DU2(M)=AVD(I)
140 CONTINUE
  Verificacao da convergencia
  A=0
  B=0
  C=0
  D=0
DO 150 I=1,L
  A=A+FN1(I)**2
  B=B+DF(I)**2
  C=C+UN1(I)**2
  D=D+U1(I)**2
150 CONTINUE
  ERRO1=DSQRT(A)/DSQRT(B)
  ERRO2=DSQRT(C)/DSQRT(D)
  IF((ERRO1.LT.0.01) .AND. (ERRO2.LT.0.01)) GO TO 1800
  NITER=NITER+1
  Solucao do sistema
  CALL RETRO(DU2,L,NB)
  MONTAGEM DO VETOR Z
DO 160 I=1,L
  ZMAT(I,IND)=U1(I)-U0(I)-(DU1(I)-DU2(I))
160 CONTINUE
DO 180 NN=1,IND-1
  AUX=0
DO 170 I=1,L
  AUX=AUX+UMAT(I,NN)*(FN(I)-FN1(I))
170 CONTINUE
DO 175 I=1,L
  ZMAT(I,IND)=ZMAT(I,IND)-ZMAT(I,NN)*AUX/ALFA(NN)
175 CONTINUE
180 CONTINUE
  Calculo do escalar ALFA
  AUX=0
DO 190 I=1,L
  AUX=AUX+UMAT(I,IND)*(FN(I)-FN1(I))
190 CONTINUE
  ALFA(IND)=AUX
300 CONTINUE
DO 200 I=1,L
  U0(I)=U1(I)
  FN(I)=FN1(I)
  DU1(I)=DU2(I)
  U1(I)=U0(I)+DU2(I)
200 CONTINUE
210 CONTINUE

```

```
***** Impressao dos resultados *****  
1320 A=0  
14 Imprimir: U1 (deslocamentos nodais), FTOT(acoes nodais)  
1520 CONTINUE  
RETURN  
END
```

```

SUBROUTINE ITERATIVO
IMPLICIT REAL*8(A-H,O-Z)
DIMENSION U0(150),U1(150)
DIMENSION BETA(10),ZMAT(150,10),UMAT(150,10),ALFA(10)
COMMON /A1/ INN(20),IID(20),AVI(20)
COMMON /A2/ GK(150,50)
COMMON /A4/ F(150),FTOT(150)
COMMON /A6/ DF(150),DFTOT(150)
COMMON /A9/ DU1(150),DU2(150),FN(150),FN(150),UN1(150)
COMMON /B1/ N,NGL,NNOS
COMMON /C1/ ANL(150)

```

```

FAZ O ITERATIVO : METODO DE PEARSON NUMERO 2

```

```

GK : Matriz de rigidez global

```

```

FTOT : Vetor de forcas; F : idem, com contorno

```

```

DF, DFTOT : incrementos de forcas

```

```

N : numero de elementos

```

```

NGL : numero de graus de liberdade por no

```

```

NNOS : numero de nos da estrutura

```

```

ANL : vetor de acoes nodais nao-lineares

```

```

INN : vetor com o numero dos nos com deslocamentos prescritos

```

```

IID : indica o numero do grau de liberdade prescrito

```

```

AVI : indica o valor dos deslocamentos prescritos

```

```

NTCA : numero de etapas de carga

```

```

*****

```

```

L=NNOS*NGL

```

```

DO 10 I=1,L

```

```

DF(I)=F(I)/NTCA

```

```

DFTOT(I)=FTOT(I)/NTCA

```

```

U0(I)=0

```

```

10 CONTINUE

```

```

DO 400 JEC=1,NTCA

```

```

NITER=1

```

```

DO 20 I=1,L

```

```

F(I)=DF(I)*JEC

```

```

FTOT(I)=DFTOT(I)*JEC

```

```

FN(I)=DF(I)

```

```

20 CONTINUE

```

```

IF(JEC.GE.2) THEN

```

```

DO 30 I=1,L

```

```

U0(I)=U1(I)

```

```

30 CONTINUE

```

```

END IF

```

```

DO 40 I=1,L

```

```

DU1(I)=DF(I)

```

```

40 CONTINUE

```

```

Solucao do sistema : substituicoes com a matriz decomposta

```

```

Metodo de Cholesky

```

```

CALL RETRO(DU1,L,NB)

```

```

DO 50 I=1,L

```

```

U1(I)=U0(I)+DU1(I)

```

```

50 CONTINUE

```

```

DO 210 JITER=1,10
IND=1
*****
Chamar subrotina para determinar o vetor de acoes nodais
neo-lineares, ANL, para os deslocamentos nodais Ui
*****
DO 60 I=1,L
FN1(I)=FTDT(I)-ANL(I)
DU2(I)=FN1(I)
60 CONTINUE
Introducao das condicoes de contorno
DO 70 I=1,NDC
IL=NGL*(INN(I)-1)
M=IL+IID(I)
FN1(M)=AVD(I)
DU2(M)=AVD(I)
70 CONTINUE
NITER=NITER+1
Solucao do sistema
CALL RETRO(DU2,L,NB)
MONTAGEM DE ZMAT E UMAT
DO 80 I=1,L
ZMAT(I,IND)=DU1(I)-DU2(I)
UMAT(I,IND)=DU1(I)
80 CONTINUE
MONTAGEM DO ESCALAR BETA
AUX=0
DO 90 I=1,L
AUX=AUX+ZMAT(I,IND)*(FN(I)-FN1(I))
90 CONTINUE
BETA(IND)=AUX
DO 300 IND=2,10
AUX=0
DO 100 I=1,L
AUX=AUX+ZMAT(I,IND-1)*FN(I)
UN1(I)=U1(I)-U0(I)-ZMAT(I,IND-1)
100 CONTINUE
DO 110 I=1,L
UN1(I)=UN1(I)*AUX/BETA(IND-1)
110 CONTINUE
ARMAZENAMENTO DE UMAT
DO 115 I=1,L
UMAT(I,IND)=UN1(I)
115 CONTINUE
DO 120 I=1,L
U0(I)=U1(I)
U1(I)=U1(I)+UN1(I)
FN(I)=FN1(I)
DU1(I)=DU2(I)
120 CONTINUE

```

```

C *****
C Chamar subrotina para determinar o vetor de acoes nodais
C nao-lineares, ANL, para os deslocamentos nodais U1
C *****
DO 130 I=1,L
FN1(I)=FTOT(I)-ANL(I)
DU2(I)=FN1(I)
130 CONTINUE
C Introducao das condicoes de contorno
DO 140 I=1,NDC
IL=NGL*(INN(I)-1)
M=IL+IID(I)
FN1(M)=AVD(I)
DU2(M)=AVD(I)
140 CONTINUE
C VERIFICACAO DA CONVERGENCIA
A=0
B=0
C=0
D=0
DO 150 I=1,L
A=A+FN1(I)**2
B=B+DF(I)**2
C=C+UN1(I)**2
D=D+U1(I)**2
150 CONTINUE
ERRO1=DSQRT(A)/DSQRT(B)
ERRO2=DSQRT(C)/DSQRT(D)
IF((ERRO1.LT.0.01) .AND. (ERRO2.LT.0.01)) GO TO 1800
NITER=NITER+1
C Solucao do sistema
CALL RETRO(DU2,L,NB)
C MONTAGEM DO VETOR Z
DO 160 I=1,L
ZMAT(I,IND)=DU1(I)-DU2(I)
160 CONTINUE
DO 180 NN=1,IND-1
AUX=0
DO 170 I=1,L
AUX=AUX+ZMAT(I,NN)*(FN(I)-FN1(I))
170 CONTINUE
DO 175 I=1,L
ZMAT(I,IND)=ZMAT(I,IND)+(UMAT(I,NN)-ZMAT(I,NN))*AUX/BETA(NN)
175 CONTINUE
180 CONTINUE
C CALCULO DO ESCALAR BETA
AUX=0
DO 190 I=1,L
AUX=AUX+ZMAT(I,IND)*(FN(I)-FN1(I))
190 CONTINUE
BETA(IND)=AUX
300 CONTINUE

```



```
DO 200 I=1,L
U0(I)=U1(I)
FN(I)=FN1(I)
DU1(I)=DU2(I)
U1(I)=U0(I)+DU2(I)
220 CONTINUE
310 CONTINUE
***** Impressao dos resultados *****
1300 A=0
Imprimir: U1 (deslocamentos nodais), FTOT (acoes nodais)
420 CONTINUE
RETURN
END
```

```

SUBROUTINE ITERATIVO
IMPLICIT REAL*8(A-H,O-Z)
DIMENSION U0(150),U1(150)
DIMENSION WMAT(150,10),VMAT(150,10)
COMMON /A1/ INN(20),IID(20),AVD(20)
COMMON /A2/ GK(150,50)
COMMON /A4/ F(150),FTOT(150)
COMMON /A6/ DF(150),DFTOT(150)
COMMON /A9/ DU1(150),FN1(150),FN(150)
COMMON /B1/ N,NGL,NNOS
COMMON /C1/ ANL(150)
SOLUCAO DE SISTEMAS DE EQUACOES NAO-LINEARES
METODO: BFGS

```

```

GK : matriz de rigidez global
FTOT : vetor de forcas; F : idem, com contorno
DF, DFTOT : incrementos de forcas
N : numero de elementos
NGL : numero de graus de liberdade por no
NNOS : numero de nos da estrutura
ANL : vetor de acoes nodais nao-lineares
INN : vetor com o numero dos nos com deslocamentos
prescritos
IID : indica o numero do grau de liberdade
AVD : indica o valor do deslocamento prescrito
NTCA : numero de etapas de carga

```

```

L=NNOS*NGL
DO 10 I=1,L
DF(I)=F(I)/NTCA
DFTOT(I)=FTOT(I)/NTCA
U0(I)=0

```

```

10 CONTINUE
DO 400 JEC=1,NTCA
NITER=0

```

```

DO 20 I=1,L
F(I)=DF(I)*JEC
FTOT(I)=DFTOT(I)*JEC
FN(I)=DF(I),

```

```

20 CONTINUE
IF(JEC.GE.2) THEN
DO 30 I=1,L
U0(I)=U1(I)

```

```

30 CONTINUE
END IF
DO 40 I=1,L
DU1(I)=DF(I)

```

```

40 CONTINUE
DO 210 JITER=1,10
NITER=NITER+1
IND=1

```

Solucao do sistema : substituicoes com a matriz decomposta
Metodo de Cholesky

CALL RETRO(DU1,L,NB)

DO 50 I=1,L

U1(I)=U0(I)+DU1(I)

CONTINUE

CHAMAR SUBROTINA PARA DETERMINAR O VETOR DE ACOES NODAIS
NAO-LINEARES, ANL, PARA OS DESLOCAMENTOS NODAIS U1

DO 60 I=1,L

FN1(I)=FTOT(I)-ANL(I)

CONTINUE

Introducao das condicoes de contorno

DO 70 I=1,NDC

IL=NGL*(INN(I)-1)

M=IL+IID(I)

FN1(M)=AVD(I)

CONTINUE

MONTAGEM DE VMAT E WMAT : Vetores auxiliares do metodo

BETA=0

ALAMB=0

DO 76 I=1,L

BETA=BETA+DU1(I)*(FN(I)-FN1(I))

ALAMB=ALAMB+DU1(I)*FN(I)

CONTINUE

AUX=BETA/ALAMB

IF(AUX.LT.0) GO TO 301.

ALFA=DSQRT(AUX)

DO 80 I=1,L

VMAT(I,IND)=FN1(I)-(1.+ALFA)*FN(I)

WMAT(I,IND)=DU1(I)/BETA

CONTINUE

DO 300 IND=2,10

DO 100 I=1,L

FN(I)=FN1(I)

CONTINUE

Calculo do incremento dos deslocamentos

DO 103 K=1,IND-1

J=IND-K

AUX=0

DO 101 I=1,L

AUX=AUX+WMAT(I,J)*FN1(I)

CONTINUE

DO 102 I=1,L

FN1(I)=FN1(I)+AUX*VMAT(I,J)

CONTINUE

CONTINUE

DO 104 I=1,L

DU1(I)=FN1(I)

CONTINUE

```

NITER=NITER+1
- Solucao do sistema
CALL RETRO(DU1,L,NB)
DO 108 J=1,IND-1
AUX=0
DO 106 I=1,L
AUX=AUX+VMAT(I,J)*DU1(I)
108 CONTINUE
DO 107 I=1,L
DU1(I)=DU1(I)+AUX*WMAT(I,J)
107 CONTINUE
123 CONTINUE
DO 120 I=1,L
U1(I)=U1(I)+DU1(I)
120 CONTINUE
122 *****
122 CHAMAR SUBROTINA PARA DETERMINAR O VETOR DE ACOES NODAIS
122 NAO-LINEARES, ANL, PARA OS DESLOCAMENTOS NODAIS U1
122 *****
DO 130 I=1,L
FN1(I)=FTOT(I)-ANL(I)
130 CONTINUE
C Introducao das condicoes de contorno
DO 140 I=1,NDC
IL=NGL*(INN(I)-1)
M=IL+IID(I)
FN1(M)=AVD(I)
140 CONTINUE
E Verificacao da convergencia
A=0
B=0
C=0
D=0
DO 150 I=1,L
A=A+FN1(I)**2
B=B+DF(I)**2
C=C+DU1(I)**2
D=D+U1(I)**2
150 CONTINUE
ERRO1=DSQRT(A)/DSQRT(B)
ERRO2=DSQRT(C)/DSQRT(D)
IF((ERRO1.LT.0.01) .AND. (ERRO2.LT.0.01)) GO TO 1800
C MONTAGEM DE VMAT E WMAT
BETA=0
ALAMB=0
DO 160 I=1,L
BETA=BETA+DU1(I)*(FN(I)-FN1(I))
ALAMB=ALAMB+DU1(I)*FN(I)
160 CONTINUE
AUX=BETA/ALAMB
IF(AUX.LT.0) GO TO 301
ALFA=DSQRT(AUX)

```

```

DO 165 I=1,L
WHAT(I,IND)=FN1(I)-(1.+ALFA)*FN(I)
WHAT(I,IND)=DU1(I)/BETA
165 CONTINUE
170 CONTINUE
180 DO 200 I=1,L
U0(I)=U1(I)
FN(I)=FN1(I)
DU1(I)=FN1(I)
190 CONTINUE
200 CONTINUE
***** Impressao dos resultados *****
1500 A=0
Imprimir : U1 (deslocamentos nodais) ; FTOT (acoes nodais)
122 CONTINUE
RETURN
END

```

```

SUBROUTINE ITERATIVO
IMPLICIT REAL*8(A-H,O-Z)
DIMENSION U0(200),U1(200),U2(200),DESF(200)
COMMON /A1/ INN(30),IID(30),AVD(30)
COMMON /A2/ GK(200,50)
COMMON /A4/ F(200),FTOT(200)
COMMON /A6/ DF(200),DFTOT(200)
COMMON /A9/ DU1(200),DU2(200),FN1(200),FN(200),UN1(200)
COMMON /B1/ N,NGL,NNOS
COMMON /C1/ ANL(200)
SOLUCAO DE SISTEMAS DE EQUACOES NAO-LINEARES
METODO SECANTE-NEWTON
GK : matriz de rigidez global
FTOT : vetor de forcas ; F : idem, com contorno
DF, DFTOT : incrementos de forcas
N : numero de elementos
NGL : numero de graus de liberdade por no
NNOS : numero de nos da estrutura
ANL : vetor de acoes nodais nao-lineares
INN : Vetor com os numeros dos nos com deslocamentos
      prescritos
IID : indica o numero do grau de liberdade prescrito
AVD : indica o valor do deslocamento prescrito
NTCA : numero de etapas de carga
*****
L=NNOS*NGL
DO 10 I=1,L
DF(I)=F(I)/NTCA
DFTOT(I)=FTOT(I)/NTCA
10 CONTINUE
DO 20 I=1,L
U0(I)=0
DU1(I)=DF(I)
20 CONTINUE
DO 400 JEC=1,NTCA
NITER=1
DO 30 I=1,L
F(I)=DF(I)*JEC
FTOT(I)=DFTOT(I)*JEC
FN1(I)=DFTOT(I)
30 CONTINUE
IF(JEC.GT.1) THEN
ETAPA 2 E DEMAIS
DO 40 I=1,L
U0(I)=U1(I)
DU1(I)=DF(I)
40 CONTINUE
END IF
Solucao do sistema : substituicoes com a matriz decomposta
Metodo de Cholesky
CALL RETRO(DU1,L,NB)

```

```

ALFA=C/A
BETA=ALFA*(1.-B/A)-1.
AUX=BETA/ALFA
BAUX=-0.2
IF((AUX.GT.0.4) .OR. (AUX.LT.BAUX)) THEN
ALFA=1
BETA=0
END IF
DO 120 I=1,L
FN(I)=FN(I)
U2(I)=U1(I)+ALFA*DU2(I)+BETA*DU1(I)
DU1(I)=U2(I)-U1(I)
U1(I)=U2(I)
121 CONTINUE
122 CONTINUE
RETURN
***** Impressao dos resultados *****
1300 A=0
Imprimir : U1(deslocamentos nodais) ; FTOT (acoes nodais)
122 CONTINUE
RETURN
END

```

SUBROUTINE ITERATIVO

IMPLICIT REAL*8(A-H,O-Z)

DIMENSION U0(200),U1(200),U2(200),DESF(200)

DIMENSION U(200)

COMMON /A1/ INN(30),IID(30),AVD(30)

COMMON /A2/ GK(200,50)

COMMON /A4/ F(200),FTOT(200)

COMMON /A6/ DF(200),DFTOT(200)

COMMON /A9/ DU1(200),DU2(200),FN1(200),FN(200),UN1(200)

COMMON /B1/ N,NGL,NNOS

COMMON /C1/ ANL(200)

SOLUCAO DE SISTEMAS DE EQUACOES NAO-LINEARES

METODO : MINIMIZACAO BIDIRECIONAL

GK : matriz de rigidez global

FTOT : vetor de forcas ; F : idem, com contorno

DF, DFTOT : incrementos de forcas

N : numero de elementos

NGL : numero de graus de liberdade por no

NNOS : numero de nos da estrutura

ANL : vetor de acoes nodais nao-lineares

INN : vetor com os numeros dos nos com deslocamentos prescritos

IID : indica o numero do grau de liberdade prescrito

AVD : indica o valor do deslocamento prescrito

NTCA : numero de etapas de carga

L=NNOS*NGL

DO 10 I=1,L

DF(I)=F(I)/NTCA

DFTOT(I)=FTOT(I)/NTCA

10 CONTINUE

DO 20 I=1,L

U0(I)=0

DU1(I)=DF(I)

20 CONTINUE

DO 400 JEC=1,NTCA

DO 30 I=1,L

F(I)=DF(I)*JEC

FTOT(I)=DFTOT(I)*JEC

FN1(I)=DFTOT(I)

30 CONTINUE

IF(JEC.GT.1) THEN

DO 40 I=1,L

U0(I)=U1(I)

DU1(I)=DF(I)

40 CONTINUE

END IF

Solucao do sistema : substituicoes com a matriz decomposta

Metodo de Cholesky

CALL RETRO(DU1,L,NB)


```

DO 50 I=1,L
U1(I)=U0(I)+DU1(I)
CONTINUE
DO 200 NITER=1,LIM
*****
CHAMAR SUBROTINA PARA DETERMINAR O VETOR DE ACOES NODAIS
NAO-LINEARES,ANL, PARA OS DESLOCAMENTOS NODAIS U1
*****
DO 60 I=1,L
DESF(I)=FTOT(I)-ANL(I)
FN(I)=DESF(I)
DU2(I)=DESF(I)
CONTINUE
Introducao das condicoes de contorno
DO 70 I=1,NDC
IL=NGL*(INN(I)-1)
H=IL+IID(I)
DU2(H)=AVD(I)
DESF(H)=AVD(I)
CONTINUE
**** Verificacao da convergencia ****
A=0
B=0
C=0
D=0
DO 80 I=1,L
A=A+DESF(I)**2
B=B+DF(I)**2
C=C+DU1(I)**2
D=D+U1(I)**2
CONTINUE
A=DSQRT(A)
B=DSQRT(B)
C=DSQRT(C)
D=DSQRT(D)
ERRO1=A/B
ERRO2=C/D
IF((ERRO1.LE.0.01) .AND. (ERRO2.LE.0.01)) GO TO 1800
Solucao do sistema
CALL RETRO(DU2,L,NB)
A=0
B=0
C=0
X=0
Y=0
DO 90 I=1,L
A=A+DU1(I)*(FN1(I)-FN(I))
B=B+DU2(I)*(FN1(I)-FN(I))
C=C+DU1(I)*FN1(I)
X=X+DU1(I)*FN(I)
Y=Y+DU2(I)*FN(I)
CONTINUE

```

```

C1=A
C2=B
C3=A*Y/C
AUX=C1*C3-C2*C2
IF(AUX.NE.0) THEN
ALFA=(C3*X-C2*Y)/AUX
BETA=(C1*Y-C2*X)/AUX
AUX=ALFA/BETA
IF((AUX.LE.0) .OR. (AUX.GE.1)) THEN
ALFA=0
BETA=1
END IF
ELSE
ALFA=0
BETA=1
END IF
C EXTRAPOLACAO
DO 100 I=1,L
FN1(I)=FN(I)
U2(I)=U1(I)+ALFA*DU1(I)+BETA*DU2(I)
DU1(I)=U2(I)-U1(I)
U1(I)=U2(I)
100 CONTINUE
200 CONTINUE
C ***** Impressao dos resultados ***
1800 A=0
C Imprimir : U1 (deslocamentos nodais) ; FTOT (acoes nodais)
400 CONTINUE
RETURN
END

```