MÉTODOS QUASE - NEWTON PARA RESOLUÇÃO DE SISTEMAS NÃO LINEARES ESPARSOS E DE GRANDE PORTE

Este exemplar corresponde à redação final
da Tese de Doutorado defendida por
Márcia Aparecida Gomes Ruggiero
e aprovada pela comissão julgadora em
26 de outubro de 1990

Orientador: PROF. JOSÉ MARIO MARTÍNEZ
DMA - IMECC - UNICAMP

Dissertação apresentada à
Faculdade de Engenharia Elétrica
Como Requisito Parcial à
Obtenção do Título de Doutor
em Engenharia Elétrica

Outubro - 1990

MÉTODOS QUASE - NEWTON PARA

RESOLUÇÃO DE SISTEMAS NÃO LINEARES

ESPARSOS E DE GRANDE PORTE

Autora: Márcia Aparecida Gomes Ruggiero
DMA - IMECC - UNICAMP

Orientador: Prof. Dr. José Mario Martínez
DMA - IMECC - UNICAMP

"Ao se escrever um programa, tem-se a sensação que para cada IF é sempre possível achar a resposta e a contra-resposta: IF...Then

.....

Else

.....

No início deste trabalho tentava encontrar a contra-resposta, o Else, para um momento pessoal difícil. Não poderia achar, simplesmente porque não existia. Algumas pessoas me ajudaram a colocar um End IF neste momento, me mostrando que a vida nem sempre apresenta contra-respostas para IFs que não se verificam. (Fácil de falar, difícil quando se vive!). Mas, a importância de se fechar com End IF é que só assim outros IFs poderão acontecer.

Dedico este trabalho a estas pessoas".

Agradecimentos:

a Mario Martínez pela proposta e orientação deste trabalho e pelo apoio e incentivo que foram essenciais na elaboração do pacote Rouxinol;

a Vera e Cheti pelo interesse em ler e discutir vários tópicos deste trabalho;

a Moretti pela contribuição decisiva na estrutura básica de Rouxinol e pelo apoio constante em todo o trabalho;

a Lúcio e Ana pelas sugestões e críticas construtivas;

aos colegas do grupo de otimização pelas várias contribuições;

a Margarida pela colaboração nas figuras e tabelas;

aos professores do depto. de Matemática Aplicada pelo incentivo;

aos funcionários Fátima, Bene e Dorival pela atenção e disponibilidade;

a Leila que processou este trabalho no Latex, pela paciência em inserir, modificar e corrigir; e a Elda e Lourdes pelo auxílio nas tabelas;

a Sérgio pela compreensão, colaboração e carinho;

a meus pais e irmãos, pelo incentivo;

a Pedro Henrique por ser do jeitinho que é;

e, finalmente, a todas pessoas aqui citadas ou não que me deram amizade e carinho em todos estes anos.

Resumo

O objetivo deste trabalho é o estudo e a análise do desempenho computacional do método de Newton e oito métodos tipo quase-Newton quando aplicados a resolução de sistemas não lineares esparsos, e de grande porte. Por razões de estabilidade numérica optamos pela fatoração LU com estratégia de pivoteamento parcial para resolver os sistemas lineares; através de uma manipulação simbólica sobre a estrutura original da matriz Jacobiana, obtém-se uma estrutura estática de dados sobre a qual são realizadas as operações algébricas necessárias para a fatoração LU. Incorporamos aos algoritmos locais uma estratégia de globalização tolerante com o objetivo de prevenir divergência quando a aproximação inicial é ruim. Introduzimos novos métodos e novas implementações de métodos já conhecidos para problemas de grande porte. Desenvolvemos o pacote Rouxinol que possibilitou a comparação numérica entre os vários métodos implementados.

ÍNDICE

CAPÍTULO 1

INTRODUÇÃO

A resolução de um sistema não linear é uma tarefa necessária durante a resolução de problemas das mais diversas áreas.

Dada a função não linear: $F : D \subset \mathbb{R}^n \to \mathbb{R}^n$, $F = (f_1, \ldots, f_n)^T$, D um conjunto aberto e convexo e $F \in C^1(D)$, o objetivo é encontrar as soluções para:

$$F(x) = 0 \tag{1.1}$$

Denotaremos a matriz Jacobiana de $F(x)$ por $J(x)$:

$$J(x) \equiv F'(x) \equiv \begin{pmatrix} \nabla f_1(x)^T \\ \vdots \\ \nabla f_n(x)^T \end{pmatrix} \equiv \begin{pmatrix} \frac{\partial f_1}{\partial x_1}(x) & \ldots & \frac{\partial f_1}{\partial x_n}(x) \\ & & \\ \frac{\partial f_n}{\partial x_1}(x) & \ldots & \frac{\partial f_n}{\partial x_n}(x) \end{pmatrix}$$

Na maioria dos problemas reais o sistema não linear resultante é de grande porte e a matriz Jacobiana é estruturalmente esparsa, isto é, muitas componentes $\frac{\partial f_i}{\partial x_j}(x)$ de $J(x)$ são nulas, [7, 11, 36].

Vários métodos têm sido propostos para a resolução de sistemas não lineares e, nosso objetivo neste trabalho é estudar e analisar o desempenho computacional destes vários métodos nos casos em que o sistema não linear é esparso e de grande porte.

O mais conhecido, entre os métodos propostos, é o método de Newton, que é um método iterativo onde a sequência de aproximações $\{x^k\}$ é gerada por:

$$x^{k+1} = x^k - J(x^k)^{-1} F(x^k) \tag{1.2}$$

e, portanto, uma iteração de Newton requer essencialmente:
- a avaliação da matriz Jacobiana em x^k;
- a resolução do sistema linear:

$$J(x^k) s_k = -F(x^k) \tag{1.3}$$

Sob o ponto de vista computacional uma iteração Newton é considerada cara uma vez que as derivadas parciais $\frac{\partial f_i}{\partial x_j}(x)$ podem ser funções complicadas e a resolução de um sistema linear é uma tarefa que envolve um esforço computacional considerável no caso em que n é grande.

A vantagem do método de Newton é que sob condições específicas é obtida a taxa quadrática de convergência, o que significa que poucas iterações serão necessárias para se obter uma solução aproximada para $F(x) = 0$, desde que a aproximação inicial x^0 seja convenientemente escolhida.

Os métodos quase Newton [2-5, 8-11, 17, 25, 27-30, 32-35, 39, 40, 42] foram introduzidos com a proposta inicial de evitar a avaliação de $J(x)$ a cada iteração.

Nestes métodos, a sequência $\{x^k\}$ é gerada através da fórmula:

$$x^{k+1} = x^k + s_k \tag{1.4}$$

onde

$$B_k s_k = -F(x^k) \tag{1.5}$$

A matriz B_{k+1} é obtida a partir de B_k através de fórmulas de recorrência que envolvem x^k, x^{k+1}, $F(x^k)$ e $F(x^{k+1})$ como informações. Em geral, B_{k+1} é escolhida entre todas as matrizes que satisfazem a equação secante:

$$B_{k+1}(x^{k+1} - x^k) = F(x^{k+1}) - F(x^k) \tag{1.6}$$

Para o caso esparso, deve-se ainda esperar que seja imposta sobre as matrizes B_k a condição de preservar ou explorar de alguma forma a estrutura de esparsidade de $J(x)$.

O método proposto por Broyden em 1965, [2, 10, 11], é um dos mais conhecidos entre os quase Newton; a fórmula para B_{k+1} consiste numa correção de posto um sobre a matriz B_k:

$$B_{k+1} = B_k + \frac{(y_k - B_k s_k)s_k^T}{s_k^T s_k} \tag{1.7}$$

onde $y_k = F(x^{k+1}) - F(x^k)$ e $s_k = x^{k+1} - x^k$.

O objetivo inicial é se evitar a avaliação da matriz Jacobiana a cada iteração, mas, a resolução do sistema linear é também simplificada pois, o fato de B_{k+1} ser

uma correção de posto um sobre B_k permite o cálculo de B_{k+1}^{-1} com $O(n^2)$ operações através da fórmula de Sherman-Morrison, [22, pg. 3], ou, se for usada a fatoração QR para resolver os sistemas, é possível obter a fatoração QR de B_{k+1} a partir da fatoração QR de B_k com $O(n^2)$ operações.

O uso da fórmula (1.7) não é adequado para o caso esparso, pois ainda que B_k tenha o mesmo padrão de esparsidade que $J(x)$, B_{k+1} pode não resultar esparsa. Daí, a aplicação do método de Broyden ao caso esparso só é viável através do uso da fórmula de Sherman-Morrison para o cálculo de B_{k+1}^{-1}. Conforme será detalhado no capítulo 2, esta implementação requer o armazenamento dos vetores que definem a correção de posto um a cada iteração.

A fórmula (1.7) é obtida ao se impor como condições que B_{k+1} satisfaça a equação secante e obedeça um princípio de variação mínima em relação a B_k. A fórmula para B_k, proposta por Schubert, 1970, [33, 40], acrescenta a estas condições, que B_{k+1} preserve a estrutura de esparsidade de $J(x)$ e, por esta razão é também denominada de "atualização esparsa de Broyden".

Na fórmula de Schubert, B_{k+1} não resulta mais de uma correção de posto um sobre B_k e, portanto, a resolução do sistema linear é uma tarefa computacionalmente cara.

Uma vez que grande parte do esforço computacional de uma iteração Newton está concentrada na resolução do sistema linear, vários métodos quase Newton têm como proposta obter B_{k+1} já na forma fatorada através da atualização direta de uma fatoração da matriz B_k.

Dennis e Marwil, em 1982, [8], propuseram o primeiro método quase Newton com esta característica; basicamente, conhecida a fatoração LU de B_k a matriz B_{k+1} é obtida atualizando-se o fator U e mantendo-se fixo o fator L. A convergência local do método só é obtida sob a hipótese de recomeços , isto é, se uma iteração Newton for efetuada a cada q iterações, q um número inteiro e fixo.

Martínez, [28] em 1983 propôs um método onde a matriz B_k é fatorada na forma LD_kM e apenas o fator D_k é atualizado para se obter a fatoração de B_{k+1}. Este método pertence à família de métodos com escalamento da fatoração introduzida em 1987 por Martínez [29]. Todos os métodos desta família apresentam taxa de convergência linear, sendo que a taxa superlinear é obtida quando se introduz

recomeços.

Chadee, em 1985, [5], generalizou o método proposto por Johnson e Austria [27], introduzindo um método com taxa superlinear para o qual a fatoração LU de B_{k+1} é obtida atualizando-se simultaneamente os fatores LU de B_k. Porém, o método de Chadee parece ser aplicável a problemas com estruturas especiais para a matriz Jacobiana uma vez que as inversas das matrizes triangulares L devem ser esparsas para que o método seja vantajoso. Em 1988, Martínez [32] introduziu uma família de métodos à qual pertencem a maioria dos métodos para resolução de sistemas não lineares com taxa superlinear. O método de Dennis Marwil não pertence a este conjunto de métodos, mas, é o caso limite de uma subfamília que contém o método de Chadee.

No método de Atualização de uma Coluna por Iteração proposto por Martínez em 1983, [30] a matriz B_{k+1} é obtida adicionando-se a B_k uma correção de posto um, e, da mesma forma que no método de Broyden, a implementação no caso esparso só é possível através da fórmula de Sherman-Morrison para B_{k+1}^{-1}.

Neste trabalho analisamos o desempenho dos seguintes métodos: Newton Modificado, Broyden, Schubert, Dennis Marwil, Atualização de uma Coluna por Iteração e três métodos da família de métodos com escalamento da fatoração.

A resolução do sistema linear: $B_k s_k = -F(x^k)$ é um passo comum a todos os métodos e tem importância central na implementação dos algoritmos. A opção quanto ao método para resolução do sistema linear deve ser feita considerando que a solução do sistema linear é o passo s_k que define a aproximação x^{k+1} e, que instabilidades numéricas podem resultar em maus resultados nos métodos testados. Daí, a opção pela estabilidade numérica ainda que esta opção resulte em prejuízo para a manutenção da esparsidade.

Optamos então pela fatoração LU com estratégia de pivoteamento parcial para resolver $B_k s_k = -F(x^k)$. Esta escolha nos conduziu a adotar uma estrutura de dados estática obtida através da implementação do algoritmo proposto por George e Ng, 1987, [21], que, essencialmente consiste em construir a partir da estrutura da matriz Jacobiana um conjunto de dados que fixa a posição de qualquer elemento não nulo que possa surgir durante a fatoração LU, qualquer que seja a sequência pivotal.

Desde que todos os métodos analisados apresentam resultados de convergência local, incorporamos aos algoritmos locais uma estratégia de convergência global na tentativa de se evitar divergência caso a aproximação inicial não seja convenientemente escolhida.

Para analisar e comparar o desempenho computacional dos vários métodos desenvolvemos o pacote Rouxinol, escrito em Fortran 77 e implementado no sistema VAX 11/785 da Unicamp. A resolução de um sistema não linear, em Rouxinol, é separada em duas fases: simbólica e numérica. Na fase simbólica a estrutura de dados para a fatoração LU é fixada através da execução do algoritmo da fatoração simbólica; na fase numérica o sistema não linear é resolvido de acordo com as opções do usuário e, no caso de comparação entre os vários métodos na resolução de um mesmo sistema não linear, vários testes consecutivos poderão ser realizados, aproveitando a estrutura de dados fixada na fase simbólica.

Os novos resultados neste trabalho são:

i) introdução dos métodos de Escalamento de Colunas e Escalamento de Linhas pertencentes à família de métodos com escalamento da fatoração proposta por Martínez [29];

ii) introdução de nova fórmula recursiva para a implementação do método de Broyden para o caso esparso;

iii) primeira implementação esparsa do método de Atualização de uma Coluna por Iteração;

iv) introdução de uma estratégia de globalização tolerante;

v) implementação computacional dos métodos de Newton e quase Newton com estrutura de dados estática obtida com o algoritmo de George e Ng [21];

vi) primeira comparação extensiva de métodos quase Newton para problemas esparsos.

Este trabalho está assim organizado: no capítulo 2 descrevemos cada método analisado e apresentamos o algoritmo correspondente; no capítulo 3 analisamos os principais resultados de convergência local; a estratégia de convergência global é

descrita no capítulo 4; o capítulo 5 consiste essencialmente no estudo da fatoração simbólica na forma proposta por George e Ng. Finalmente, no capítulo 6 descrevemos as principais características de Rouxinol e apresentamos: a comparação entre o desempenho computacional dos vários métodos; uma análise do uso da estrutura estática em relação à estrutura de dados dinâmica, usando os resultados dos testes com Rouxinol e os resultados obtidos ao se usar o pacote MA28 de Harwell, [12] para resolver os sistemas lineares.

CAPÍTULO 2

ALGORITMOS

Neste capítulo, descrevemos os seguintes métodos, escolhidos para análise neste trabalho:
- método de Newton;
- método de Newton Modificado;
- método de Broyden;
- método de Schubert;
- método de Dennis-Marwil;
- método de Atualização de uma Coluna por Iteração;
- 3 métodos da família de métodos com escalamento da fatoração:
  - Atualização do Fator Diagonal;
  - Escalamento de Linhas;
  - Escalamento de Colunas.

A iteração inicial de todos os métodos é uma iteração de Newton, isto é, $B_0 = J(x^0)$, e, conforme já colocado na introdução, a resolução dos sistemas lineares é através da fatoração LU com estratégia de pivoteamento parcial.

A possibilidade da matriz B_k ser singular é considerada e prevenida nos algoritmos por um teste relativo que usa como tolerância um parâmetro que denotamos Tolsing.

Teoricamente, o novo ponto x^{k+1} é obtido pela fórmula: $x^{k+1} = x^k + s_k$ onde s_k é solução do sistema linear $B_k s = -F(x^k)$. Contudo, convém escalar o vetor solução deste sistema para se evitar problemas provocados por passos excessivamente grandes. Pode-se esperar tais ocorrências, no caso de B_k estar próxima de uma matriz singular. Efetuando um controle sobre o tamanho do passo, s_k, o ponto x^{k+1} é assim obtido:

$$x^{k+1} = x^k + \theta s_k \quad \text{onde :}$$

$B_k s_k = -F(x^k)$ e,

$\theta = \min\{1, \beta/||s_k||_\infty\}$ e $\beta > 0$, é uma estimativa para a distância entre a aproximação inicial x^0 e a solução do sistema não linear x^*.

2.1. Método de Newton

A sequência $\{x^k\}$ gerada no método de Newton é tal que a aproximação x^{k+1} é um zero de um modelo local linear para $F(x)$ construído em torno de x^k.

Dado que $F : D \subset \mathbb{R}^n \to \mathbb{R}^n$, $F \in C^1(D)$, temos que $f_i \in C^1(D)$ para $i = 1, \ldots, n$.

Então, conhecida a aproximação $x^k \in D$, para qualquer $x \in D$, existe $\xi_i \in D$, ξ_i entre x^k e x tal que :

$$f_i(x) - f_i(x^k) = \nabla f_i(\xi_i)^T (x - x^k) \qquad i = 1, \ldots, n$$

Aproximando $\nabla f_i(\xi_i)$ por $\nabla f_i(x^k)$, $i = 1, \ldots, n$, temos um modelo local linear para $F(x)$ em torno de x^k:

$$L_k(x) = F(x^k) + J(x^k)(x - x^k)$$

Agora,

$$L_k(x) = 0 \Leftrightarrow J(x^k)(x - x^k) = -F(x^k)$$

e, portanto,

$$x^{k+1} = x^k - (J(x^k))^{-1} F(x^k)$$

Uma iteração de Newton requer basicamente:
- a avaliação da matriz Jacobiana em x^k;
- a resolução do sistema linear: $J(x^k)s = -F(x^k)$

e, por este motivo, cada iteração é considerada computacionalmente cara.

Esta desvantagem é compensada pela taxa quadrática de convergência obtida sob certas hipóteses na vizinhança de x^*.

Algoritmo 2.1.

Dados x^0, a aproximação inicial e os parâmetros $\beta > 0$ e Tolsing, execute:

Passo 1: calcule $F(x^k)$ e $J(x^k)$ e faça $B_k = J(x^k)$

Passo 2: calcule as matrizes P, L e U, tais que
$$PB_k = LU$$

Passo 3: (salvaguarda contra singularidade):

$$\left[\begin{array}{ll} \text{para} & i = 1, \ldots, n, \quad \text{faça :} \\ \text{se} & |u_{ii}| < \text{Tolsing} \max_{ij}\{|e_i^T B_k e_j|\} \quad \text{faça} \\ & u_{ii} \leftarrow sgn(u_{ii})\ \text{Tolsing} \end{array}\right.$$

Passo 4: resolva $Lw = P(-F(x^k))$
$Us_k = w$

Passo 5: obtenha $\theta = \min\{1, \beta/||s_k||_\infty\}$
se $\theta \neq 1$ faça $s_k \leftarrow \theta s_k$

Passo 6: calcule $x^{k+1} = x^k + s_k$

Passo 7: faça $k = k + 1$
volte ao passo 1

2.2. Método de Newton Modificado

A modificação sobre o método de Newton consiste em se tomar a cada iteração k, $B_k = J(x^0)$. Desta forma, tanto a avaliação da matriz Jacobiana, quanto sua fatoração LU são efetuadas uma única vez.

Algoritmo 2.2.

Dados x^0 a aproximação inicial, os parâmetros $\beta > 0$ e Tolsing, execute:

Passo 1: calcule $F(x^0)$ e $J(x^0)$ e faça $B_0 = J(x^0)$

Passo 2: calcule as matrizes P, L e U, tais que

$$PB_0 = LU$$

Passo 3: (salvaguarda contra singularidade):

$$\left[\begin{array}{ll} \text{para} & i = 1, \ldots, n, \quad \text{faça :} \\ \text{se} & |u_{ii}| < \text{Tolsing} \max_{ij}\{|e_i^T B_0 e_j|\} \quad \text{faça} \\ & u_{ii} \leftarrow sgn(u_{ii})\ \text{Tolsing} \end{array}\right.$$

Passo 4: resolva $Lw = P(-F(x^k))$
$Us_k = w$

Passo 5: obtenha $\theta = \min\{1, \beta/||s_k||_\infty\}$
se $\theta \neq 1$ faça $s_k \leftarrow \theta s_k$

Passo 6: calcule $x^{k+1} = x^k + s_k$

Passo 7: faça $k = k + 1$
volte ao passo 4

2.3. Método de Broyden

A classe de métodos proposta por Broyden em 1965, [2], tem como objetivo central, evitar a avaliação da matriz Jacobiana $J(x)$.

As iterações são realizadas de acordo com a fórmula:

$$x^{k+1} = x^k - B_k^{-1} F(x^k),$$

sendo que as matrizes B_{k+1} devem satisfazer a equação:

$$B_{k+1}(x^{k+1} - x^k) = F(x^{k+1}) - F(x^k)$$

denominada "equação secante", devido à seguinte motivação:
"conhecidos x^k, $F(x^k)$, x^{k+1}, $F(x^{k+1})$, o modelo afim:

$$L_{k+1}(x) = F(x^{k+1}) + B_{k+1}(x - x^{k+1})$$

pode ser considerado como uma aproximação para $F(x)$ em torno de x^{k+1}, e, a igualdade: $L_{k+1}(x^{k+1}) = F(x^{k+1})$ é satisfeita, para qualquer matriz B_{k+1}. Colocando

também a condição que o modelo afim assuma o mesmo valor que $F(x)$ no ponto anterior x^k, teremos:

$$L_{k+1}(x^k) = F(x^{k+1}) + B_{k+1}(x^k - x^{k+1}) = F(x^k) \Leftrightarrow$$
$$\Leftrightarrow B_{k+1}(x^{k+1} - x^k) = F(x^{k+1}) - F(x^k) \qquad (2.3.1)$$

Se $x^{k+1} - x^k = s_k$ e $F(x^{k+1}) - F(x^k) = y_k$, 2.3.1 pode ser escrita:

$$B_{k+1}s_k = y_k \text{ "} \qquad (2.3.2)$$

São chamados de Métodos Secantes os Métodos Quase Newton que impõem a condição (2.3.2) sobre as matrizes B_k.

A equação secante, não é suficiente para determinar uma única matriz, (quando $n > 1$).

Conhecidos $s_k \neq 0$ e y_k denotaremos por $Q(s_k, y_k)$ o conjunto das matrizes de ordem n que satisfazem (2.3.2):

$$Q(s_k, y_k) = \{B \in I\!R^{n\times n} | Bs_k = y_k\} \qquad (2.3.3)$$

É importante observar que $Q(s_k, y_k) \neq \phi$, pois da hipótese que: para todo $i = 1, \ldots, n$, $f_i(x) \in C^1(D)$, decorre que existe $\xi_i \in I\!R^n$, ξ_i entre x^k e $x^k + s_k$, tal que:

$$f_i(x^k + s_k) = f_i(x^k) + \nabla fi(\xi_i)^T s_k$$

e, portanto, a matriz $\tilde{J} = [\nabla f_i(\xi_i)^T]$, pertence ao conjunto $Q(s_k, y_k)$.

Os métodos secantes diferem entre si, pelas condições adicionais impostas sobre B_{k+1}, tais como:
– preservar alguma estrutura especial da matriz Jacobiana, como simetria e esparsidade;
– obedecer algum princípio de variação mínima em relação à B_k.

O primeiro método proposto por Broyden coloca como condição adicional que B_{k+1} deve ser construída de modo que a troca no modelo afim seja mínima, isto porque o cálculo de x^{k+1} não resulta em novas informações a respeito da matriz Jacobiana ou do modelo linear, de modo que é razoável preservar tanto quanto possível o modelo atual.

Conforme veremos abaixo, esta condição implicará em se impor que B_{k+1} não seja diferente de B_k no espaço ortogonal a s_k:

$$B_{k+1}t = B_k t \quad \text{para todo} \quad t \in I\!R^n, \quad t^T s_k = 0$$

Com efeito, para qualquer $x \in I\!R^n$:

$$\begin{aligned} L_{k+1}(x) - L_k(x) &= F(x^{k+1}) + B_{k+1}(x - x^{k+1}) - F(x^k) - B_k(x - x^k) \\ &= F(x^{k+1}) - F(x^k) - B_{k+1}(x^{k+1} - x^k) + (B_{k+1} - B_k)(x - x^k) \end{aligned}$$

B_{k+1} deve pertencer à $Q(s_k, y_k)$, então:

$$L_{k+1}(x) - L_k(x) = (B_{k+1} - B_k)(x - x^k) \tag{2.3.4}$$

Agora, para cada $x \in I\!R^n$, existem: $\lambda \in I\!R$ e $v \in I\!R^n$, $v^T s_k = 0$, tais que:

$$x = \lambda s_k + v$$

Então, $(x - x^k)$ pode ser escrito como:

$$x - x^k = \alpha s_k + t \qquad \alpha \in I\!R \quad \text{e} \quad t \in I\!R^n, \ t^T s_k = 0,$$

portanto, (2.3.4) fica:

$$\begin{aligned} L_{k+1}(x) - L_k(x) &= (B_{k+1} - B_k)(\alpha s_k + t) \\ &= \alpha(B_{k+1} - B_k)s_k + (B_{k+1} - B_k)t \end{aligned}$$

O primeiro termo do lado direito independe de x e, como B_{k+1} deve pertencer à $Q(s_k, y_k)$:

$$\alpha(B_{k+1} - B_k)s_k = \alpha(y_k - B_k s_k)$$

Então, a troca no modelo afim será mínima para todo $x \in I\!R^n$ se B_{k+1} for escolhida, entre as matrizes de $Q(s_k, y_k)$, de modo que:

$$(B_{k+1} - B_k)t = 0 \qquad \forall t \in I\!R^n \ , \ t^T s_k = 0 \tag{2.3.5}$$

É fácil concluir que (2.3.5) se cumpre quando $(B_{k+1} - B_k)$ é uma matriz de posto um da forma: us_k^T, $u \in I\!R^n$.

Daí: $B_{k+1} = B_k + us_k^T$ e B_{k+1} deve pertencer a $Q(s_k, y_k)$, então:

$$(B_k + us_k^T)s_k = y_k \Rightarrow u = \frac{y_k - B_k s_k}{s_k^T s_k}$$

Finalmente:

$$B_{k+1} = B_k + \frac{(y_k - B_k s_k)}{s_k^T s_k} s_k^T \tag{2.3.6}$$

que é a "atualização secante de Broyden" para B_k.

Esta fórmula para B_{k+1} também será obtida, ao se impor que B_{k+1} deve ser a matriz de $Q(s_k, y_k)$ mais próxima de B_k, considerando a norma de Frobenius. Geometricamente, isto significa que B_{k+1} é a projeção ortogonal de B_k em $Q(s_k, y_k)$.

Este resultado é formalizado através do teorema:

Teorema 2.3.1. Dada a matriz $B \in \mathbb{R}^{n\times n}$ e os vetores: $y, s \in \mathbb{R}^n$, $s \neq 0$.
A matriz $\overline{B}$, dada por:

$$\overline{B} = B + \frac{(y - Bs)s^T}{s^T s} \text{ é a única solução do problema:}$$

$$\left[\begin{array}{ll} \text{Min :} & ||\widehat{B} - B||_F \\ \text{Suj.a :} & \widehat{B} \in Q(s, y) \end{array}\right.$$

Dem: $\overline{B}$ é de fato uma solução para o problema, pois:

$$\begin{aligned} ||\overline{B} - B||_F &= \left\| \left(B + \frac{(y - Bs)s^T}{s^T s} \right) - B \right\|_F = \left\| \frac{(y - Bs)s^T}{s^T s} \right\|_F = \\ &= \left\| \frac{(\widehat{B}s - Bs)s^T}{s^T s} \right\| = \left\| (\widehat{B} - B)\frac{ss^T}{s^T s} \right\|_F \\ &\leq \min\left\{ ||\widehat{B} - B||_2 \left\| \frac{ss^T}{s^T s} \right\|_F \ , \ ||\widehat{B} - B||_F \left\| \frac{ss^T}{s^T s} \right\|_2 \right\} \\ &= \min\left\{ ||\widehat{B} - B||_2 \left\| \frac{ss^T}{s^T s} \right\|_F \ , \ ||\widehat{B} - B||_F \right\} \leq ||\widehat{B} - B||_F \end{aligned}$$

Além disto, $\overline{B}$ é a única solução porque a aplicação: $h : \mathbb{R}^{n\times n} \to \mathbb{R}$ definida por $h(M) = ||M - B||_F$ é estritamente convexa e o conjunto $Q(s, y)$ é convexo. □

O fato de B_{k+1} ser obtida adicionando-se um fator de posto um à B_k, dificulta a implementação deste método no caso esparso, pois B_{k+1} pode não resultar esparsa, mesmo que B_k o seja. Ainda que B_{k+1} resulte esparsa, a estrutura de esparsidade

de B_k não será preservada, o que significa que a esparsidade da matriz Jacobiana não pode ser explorada.

Descreveremos a implementação deste método no caso esparso usando a fórmula de Sherman–Morrison para o cálculo de B_{k+1}^{-1}.

Para facilitar a notação, definiremos:

$$\begin{aligned} u_k &= \frac{y_k - B_k s_k}{s_k^T s_k} \quad \text{e, daí, teremos:} \\ B_{k+1} &= B_k + u_k s_k^T \end{aligned}$$

Através da fórmula de Sherman–Morrison, se B_k^{-1} existe e, se $(1 + s_k^T B_k^{-1} u_k) \neq 0$, então, B_{k+1}^{-1} existe e, é dada por

$$B_{k+1}^{-1} = B_k^{-1} - \frac{B_k^{-1} u_k s_k^T B_k^{-1}}{1 + s_k^T B_k^{-1} u_k} \tag{2.3.7}$$

Substituíndo u_k, vem que:

$$B_{k+1}^{-1} = B_k^{-1} + \frac{(s_k - B_k^{-1} y_k) s_k^T B_k^{-1}}{s_k^T B_k^{-1} y_k} \tag{2.3.8}$$

Definindo agora:

$$w_k = \frac{s_k - B_k^{-1} y_k}{s_k^T B_k^{-1} y_k} \tag{2.3.9}$$

a expressão para B_{k+1}^{-1} fica:

$$B_{k+1}^{-1} = (I + w_k s_k^T) B_k^{-1} \tag{2.3.10}$$

e, aplicando recursivamente (2.3.9)–(2.3.10), vem

$$B_{k+1}^{-1} = (I + w_k s_k^T)(I + w_{k-1} s_{k-1}^T) \ldots (I + w_0 s_0^T) B_0^{-1} \tag{2.3.11}$$

Conforme se pode observar da expressão (2.3.11), o uso da fórmula de Sherman–Morrison implica no armazenamento de dois vetores adicionais à cada iteração k

efetuada: w_k e s_k, e, por esta razão, o número de iterações consecutivas do método é limitado pela disponibilidade de memória da máquina.

Considerando que há espaço para armazenar m pares de vetores s_k e w_k, é então possível efetuar uma iteração Newton e m iterações Broyden consecutivas, e, neste caso, (2.3.11) deve ser escrita:

$$B_{k+1}^{-1} = (I + w_k s_k^T)(I + w_{k-1} s_{k-1}^T) \ldots (I + w_\ell s_\ell^T) B_\ell^{-1} \qquad (2.3.12)$$

onde $\ell \equiv 0 \pmod{(m+1)}$

Algoritmo 2.3.

Dados x^0, aproximação inicial e os parâmetros $\beta > 0$ e Tolsing, e, o número inteiro m, execute:

Passo 0: $k = 0, \ \ell = 0$

Passo 1: obtenha $B_k = J(x^k)$

Passo 2: obtenha as matrizes P, L, U, tais que:
$$PB_k = LU$$

Passo 3: resolva $LU\tilde{s}_k = P(-F(x^k))$

Passo 4: obtenha $\theta = \min\{1, \beta / \ ||\tilde{s}_k||_\infty\}$
faça $s_k = \theta \tilde{s}_k$

Passo 5: faça $x^{k+1} = x^k + s_k$

Passo 6: obter q tal que $k \equiv q(\ \mathrm{mod}\ (m+1))$
se $k \neq \ell$ e $q = 0$ faça $k = k + 1$
$\ell = k.$
volte ao passo 1
caso contrário:

Passo 7: (cálculo do vetor w_k)
execute os passos 7.1 à 7.3

Passo 7.1: (cálculo de $t = B_k^{-1} F(x^{k+1})$)
execute os passos 7.1.1 à 7.1.2

Passo 7.1.1: (resolução de $B_\ell t = -F(x^{k+1})$)
resolva $LUt = -PF(x^{k+1})$

Passo 7.1.2: para $j = (\ell), \ldots, (k-1)$ faça:
$t \leftarrow (I + w_j s_j^T)t$

Passo 7.2: (cálculo de $v = B_k^{-1} y_k = B_k^{-1}(F(x^{k+1}) - F(x^k))$:
$v = \tilde{s}_k - t$

Passo 7.3: faça $w_k = (s_k - v)/s_k^T v$

Passo 8: (completa o cálculo do produto: $-B_{k+1}F(x^{k+1})$)
faça $\tilde{s}_{k+1} = t + w_k s_k^T t$

Passo 9: $k = k + 1$
volte ao passo 4

A condição para que B_{k+1} seja inversível: $(1 + s_k^T B_k^{-1} u_k) \neq 0$, equivale a impor que $s_k^T B_k^{-1} y_k \neq 0$ (conforme a expressão 2.3.9).

Na implementação do algoritmo, considerando que $v = B_k^{-1} y_k$, se:

$$|s_k^T v| < \text{Tolsing}\ ||s_k||||v||\ , \quad \text{definimos} \quad B_{k+1} = B_k$$

2.4. Método de Schubert

Vimos que a estrutura de esparsidade da matriz Jacobiana $J(x)$ não pode ser explorada ao se usar a fórmula de atualização de Broyden para se obter B_{k+1}.

A fórmula de atualização proposta por Schubert em 1970, [40], tem um objetivo mais amplo que conservar a estrutura de esparsidade, uma vez que não altera qualquer elemento constante da matriz Jacobiana. Isto é, para todo i, j tais que:

$\frac{\partial f_i}{\partial x_j}(x) = \alpha_{ij}$, onde α_{ij} é uma constante conhecida, nula ou não, o elemento correspondente em B_k, $b_{ij}^{(k)}$, será igual a α_{ij}, para todo k.

Denotaremos a i-ésima linha de B_k por b_i^k.

Para cada linha $i = 1, \ldots, n$ de $J(x)$, construímos o conjunto : $I_i = \left\{ j \middle| \frac{\partial f_i}{\partial x_j}(x) = \alpha_{ij},\ \forall x \in I\!R^n \right\}$ e seja $(n - r_i)$ o número de índices em I_i.

Definiremos :
o vetor coluna $\hat{s}_{ik}$, obtido a partir do vetor coluna s_k e do conjunto I_i, por:

$$\hat{s}_{ik}(j) = 0 \quad \text{se} \quad j \in I_i \quad \text{e}$$
$$\hat{s}_{ik}(j) = s_k(j) \quad \text{se} \quad j \notin I_i$$

e o vetor linha $\bar{b}_i$, por:

$$\bar{b}_{ij} = 0 \quad \text{se} \quad j \notin I_i \quad \text{e} \quad \bar{b}_{ij} = \alpha_{ij} \quad \text{se} \quad j \in I_i$$

O papel do vetor $\bar{b}_i$ é fixar as componentes da linha i de B_k que correspondem às componentes constantes da linha i de $J(x)$ e, por esta razão, $\bar{b}_i$ independe da iteração k.

Além da condição de não alterar qualquer elemento constante em $J(x)$, que chamaremos de condição de Schubert, as matrizes B_k devem satisfazer a equação secante.

Seja S o conjunto definido por:

$$S = \{B \in I\!R^{n \times n} |\ \text{i)}\ b_{ij} = \alpha_{ij}\ \text{ para }\ j \in I_i\ \text{ e ii)}\ Bs_k = y_k\}$$

Observemos que $S \neq \phi$, uma vez que pelas hipóteses sobre $f_i(x)$, $i = 1, \ldots, n$, existe $\xi_i \in I\!R^n$, ξ_i entre x^k e $x^k + s_k$ tal que:

$$f_i(x^k + s_k) = f_i(x^k) + \nabla f_i(\xi_i)^T s_k$$

Daí, a matriz $\tilde{J} \in I\!R^{n \times n}$, dada por $\tilde{J} = [\nabla f_i(\xi_i)^T]$ pertence ao conjunto S.

A estas duas condições, acrescenta-se a condição de variação mínima entre os

modelos afins para que a matriz B_{k+1} seja unicamente determinada.

Então, conhecidos os vetores x^k, x^{k+1}, $F(x^k)$, $F(x^{k+1})$ e a matriz B_k, deduziremos uma fórmula para b_i^{k+1}, $i = 1, \ldots, n$, a partir das condições colocadas.

A construção de b_i^{k+1} se inicia pelas componentes $j \in I_i$:

$$b_{ij}^{k+1} = \alpha_{ij} \qquad j \in I_i$$

Em seguida, a equação secante equivale a: $b_i^{k+1} s_k = y_i^k$, $i = 1, \ldots, n$, onde y_i^k representa a componente i do vetor y^k.

Considerando a definição dos vetores: $\overline{b}_i$ e $\widehat{s}_{ik}$, teremos:

$$b_i^{k+1} s_k = y_i^k \;\Leftrightarrow\; \overline{b}_i s_k + b_i^{k+1} \widehat{s}_{ik} = y_i^k \,, \quad i = 1, 2, \ldots n \tag{2.4.1}$$

A condição de variação mínima, implica em:

$$\begin{aligned} B_{k+1} t = B_k t \qquad & \forall t \in I\!R^n, \quad t^T s_k = 0 \;\Leftrightarrow \\ b_i^{k+1} t = b_i^k t \qquad & i = 1, 2, \ldots, n \;\text{ e }\; \forall t \in I\!R^n \,, \; t^T s_k = 0 \end{aligned}$$

Para cada $i = 1, \ldots, n$ a linha i de B_k possui $(n - r_i)$ componentes constantes em qualquer iteração k; então, a condição de variação mínima deve ser imposta separadamente a cada linha i, $i = 1, \ldots, n$, e, deve ser restrita ao subespaço de $I\!R^n$, de dimensão r_i ao qual pertence o vetor $\widehat{s}_{ik}$:

$$\begin{aligned} & b_i^{k+1} \widehat{t} = b_i^k \widehat{t} \,, \;\; \forall \widehat{t} \in I\!R^n \,, \quad \text{tal que} \;\; \widehat{t}^T \widehat{s}_{ik} = 0 \,, \;\; i = 1, \ldots, n \\ & \Leftrightarrow (b_i^{k+1} - b_i^k) \widehat{t} = 0 \quad \forall \widehat{t} \in I\!R^n \,, \; \widehat{t}^T \widehat{s}_{ik} = 0 \quad i = 1, \ldots, n \end{aligned} \tag{2.4.2}$$

se, $b_i^{k+1} - b_i^k = \lambda_i \widehat{s}_{ik}^T \quad i = 1, \ldots, n$ a condição (2.4.2) será satisfeita e, substituíndo b_i^{k+1} por $b_i^k + \lambda_i \widehat{s}_{ik}^T$ em (2.4.1), teremos:

$$(b_i^k + \lambda_i \widehat{s}_{ik}^T) \widehat{s}_{ik} = y_i^k - \overline{b}_i s_k \quad i = 1, \ldots, n$$

Agora, se $\widehat{s}_{ik}^T \widehat{s}_{ik} \neq 0$ teremos:

$$\lambda_i = \frac{y_i^k - \overline{b}_i s_k - b_i^k \widehat{s}_{ik}}{\widehat{s}_{ik}^T \widehat{s}_{ik}} \quad i = 1, \ldots, n$$

e, da definição dos vetores $\bar{b}_i$ e $\hat{s}_{ik}$:

$$\lambda_i = \frac{y_i^k - b_i^k s_k}{\hat{s}_{ik}^T \hat{s}_{ik}} \qquad i = 1, \ldots, n$$

e, neste caso, a linha i, de B_{k+1} será dada por:

$$b_i^{k+1} = b_i^k + \frac{(y_i^k - b_i^k s_k)\hat{s}_{ik}^T}{\hat{s}_{ik}^T \hat{s}_{ik}} \qquad i = 1, \ldots, n$$

No caso em que $\hat{s}_{ik}^T \hat{s}_{ik} = 0$ a equação i do sistema de equações secantes será satisfeita qualquer que seja b_i^{k+1} que satisfaça a condição de Schubert, pois:

$$\hat{s}_{ik}^T \hat{s}_{ik} = 0 \Rightarrow \hat{s}_{ik} = 0 \Rightarrow x_j^{k+1} = x_j^k \qquad \forall j \notin I_i$$

e, daí, teremos que:

$$y_i = f_i(x^{k+1}) - f_i(x^k) = \sum_{j \in I_i} \alpha_{ij}(x_j^{k+1} - x_j^k) = \bar{b}_i s_k$$

Para atender a condição de variação mínima, escolhemos: $b_i^{k+1} = b_i^k$.

Concluíndo, a matriz B_{k+1} que satisfaz a condição de Schubert, a equação secante e a condição de variação mínima é dada por:

$$\left[\begin{array}{ll} \text{para} & i = 1, \ldots, n : \\ & b_i^{k+1} = b_i^k + \dfrac{(y_i - b_i^k s_k)\hat{s}_{ik}^T}{\hat{s}_{ik}^T \hat{s}_{ik}} \quad \text{se} \quad \hat{s}_{ik}^T \hat{s}_{ik} \neq 0 \\ & \text{e} \\ & b_i^{k+1} = b_i^k \quad \text{caso contrário.} \end{array}\right. \tag{2.4.3}$$

Da mesma forma que no método de Broyden, a fórmula (2.4.3) para B_{k+1} é também obtida ao se impor que B_{k+1} deve ser a matriz do conjunto S, (definido anteriormente) mais próxima de B_k, considerando a norma de Frobenius; isto é, B_{k+1} resolve o problema:

$$\left[\begin{array}{ll} \text{Min :} & ||B - B_k||_F \\ \text{Suj.a :} & B \in S \end{array}\right.$$

Algoritmo 2.4.

Dados x^0, a aproximação inicial, os parâmetros $\alpha > 0$, $\beta > 0$ e Tolsing, as constantes α_{ij} e os conjuntos I_i, $i = 1, \ldots, n$ definidos por:

$$I_i = \left\{ j | \frac{\partial f_i}{\partial x_j} = \alpha_{ij} \right\} \text{ , execute :}$$

Passo 1: $k = 0$. Calcule $F(x^0)$ e $J(x^0)$

Passo 2: faça $B_0 = J(x^0)$

Passo 3: obtenha as matrizes P, L, U, tais que:
$$PB_k = LU$$

Passo 4: (salvaguarda contra singularidade):

$$\left[\begin{array}{l} \text{para } i = 1, \ldots, n \quad \text{faça :} \\ \text{se } |u_{ii}| < \text{Tolsing } \max_{i,j}\{|e_i^T B_k e_j|\} \quad \text{faça :} \\ \qquad u_{ii} \leftarrow sgn(u_{ii})\text{Tolsing} \end{array}\right.$$

Passo 5: resolva : $LUs_k = P(-F(x^k))$

Passo 6: obtenha $\theta = \min\{1, \beta/||s_k||_\infty\}$
se $\theta \neq 1$, faça $s_k \leftarrow \theta s_k$

Passo 7: $x^{k+1} = x^k + s_k$

Passo 8: (atualização da matriz B_k)

$$\left[\begin{array}{l} \text{para cada } i = 1, \ldots, n \quad \text{execute :} \\ \\ \text{passo 8.1 : obtenha o vetor } z \text{ por :} \\ \qquad z_j = 0 \text{ se } j \in I_i \\ \qquad z_j = s_j^k \text{ caso contrário} \\ \\ \text{passo 8.2 : se } ||z||_2 > \alpha||s_k||_2 \text{ faça :} \\ \qquad b_i^{k+1} = b_i^k + \dfrac{(f_i(x^{k+1}) - (1-\theta)f_i(x^k))z^T}{z^T z} \\ \qquad \text{caso contrário:} \\ \qquad b_i^{k+1} = b_i^k \end{array}\right.$$

Passo 9: $k = k + 1$

volte ao passo 3.

2.5. Método de Dennis–Marwil

Este foi o primeiro método quase Newton colocado com o objetivo de reduzir o esforço na etapa de resolução do sistema linear.

O argumento que originou o método, proposto por Dennis e Marwil em 1982, [8], foi o seguinte:

"conhecidas as matrizes B_k, P_k, L_k e U_k, tais que: $P_k B_k = L_k U_k$, e, se B_k está próxima de J_*, então a matriz $L_k^{-1} P_k J_*$ pode ser aproximada por uma matriz triangular superior.

Considerando U_k como sendo uma possível aproximação para $L_k^{-1} P_k J_*$, a matriz U_{k+1} é obtida através da atualização esparsa de Broyden aplicada sobre U_k, na tentativa de se obter uma aproximação melhor para $L_k^{-1} P_k J_*$ ao se incorporar as informações obtidas no cálculo de x^{k+1}".

Por atualização esparsa de Broyden sobre U_k, entenda-se que U_{k+1} é construída de modo que:
- a estrutura de esparsidade de U_k é preservada;
- B_{k+1}, definida por $PB_{k+1} = LU_{k+1}$, satisfaz a equação secante;
- a variação nos modelos afins $L_{k+1}(x)$ e $L_k(x)$ é mínima.

Chamaremos a primeira condição de condição de Dennis–Marwil.

Seja S o conjunto definido por:

$$\begin{aligned} S = \{ & U \in \mathbb{R}^{n \times n},\ U \text{ triangular superior} | \\ & \text{i) } u_{ij} = 0 \text{ se } u_{ij}^k = 0 \ ; \\ & \text{ii) } B \in Q(s_k,\ y_k),\ B \text{ definida por } P_k B = L_k U \} \end{aligned}$$

O conjunto S pode ser vazio e neste caso é dada prioridade a condição de Dennis-Marwil durante a construção da matriz U.

Uma vez que as matrizes P_k e L_k não serão alteradas, serão denotadas por P e L.

Para cada linha i de U_k, u_i^k, definiremos: o conjunto $I_i = \{j | u_{ij}^k = 0\}$; e, r_i como

sendo o total de componentes a se determinar para obter a linha i de U_{k+1} : u_i^{k+1}.

B_{k+1} deve satisfazer a equação secante, então:

$$B_{k+1}s_k = y_k \Leftrightarrow PB_{k+1}s_k = Py_k \Leftrightarrow (LU_{k+1})s_k = Py_k \Leftrightarrow U_{k+1}s_k = L^{-1}Py_k \quad (2.5.1)$$

Denotando o vetor coluna $L^{-1}Py_k$ por v_k e sua i-ésima componente por v_i^k, a relação (2.5.1) acima equivale a:

$$u_i^{k+1}s_k = v_i^k \qquad i = 1, \ldots, n \quad (2.5.2)$$

A condição de variação mínima implica que para todo $t \in \mathbb{R}^n, t$ ortogonal a s_k, deve-se ter:

$$B_{k+1}t = B_k t \Leftrightarrow U_{k+1}t = U_k t \Leftrightarrow u_i^{k+1}\, t = u_i^k\, t, \quad i = 1, \ldots, n \quad (2.5.3)$$

Uma vez que para qualquer $j \in I_i$, as componentes u_{ij}^{k+1} já estão determinadas: $u_{ij}^{k+1} = u_{ij}^k = 0$ a relação (2.5.3) deve ser restrita a um subespaço de $\mathbb{R}^n$, de dimensão r_i;

$$u_i^{k+1}\hat{t} = u_i^k\hat{t}, \quad i = 1, \ldots, n \quad \text{e todo} \quad \hat{t} \in \mathbb{R}^n \quad \text{tal que} \quad z_i^T\hat{t} = 0 \quad (2.5.4)$$

onde o vetor z_i é definido por:

$$z_j^i = s_j^k \quad \text{se} \quad j \notin I_i \quad \text{e} \quad z_j^i = 0 \quad \text{caso contrário} \quad (2.5.5)$$

Se $u_i^{k+1} = u_i^k + \lambda z_i^T$, (2.5.4) será satisfeita.

Substituíndo esta expressão para u_i^{k+1} em (2.5.2), vem que:

$$(u_i^k + \lambda z_i^T)s_k = v_i^k \Leftrightarrow \lambda z_i^T s_k = v_i^k - u_i^k s_k \Leftrightarrow \lambda z_i^T z_i = v_i^k - u_i^k\, s_k$$

Se, $z_i^T z_i \neq 0$, $\lambda = \dfrac{v_i^k - u_i^k s_k}{z_i^T z_i}$, e, neste caso, para cada $j \notin I_i$, teremos:

$$u_{ij}^{k+1} = u_{ij}^k + \frac{(v_i^k - u_i^k z_i)}{z_i^T z_i} s_j^k \qquad i = 1, \ldots, n \quad (2.5.6)$$

onde o vetor z_i é o vetor acima definido.

Agora, $z_i^T z_i = 0$ significa que $s_j^k = 0 \ \forall j \notin I_i$ e, daí, a equação i do sistema de equação secante, fica:

$$\begin{aligned} u_i^{k+1}s_k = v_i \Leftrightarrow v_i = 0 \quad &\Leftrightarrow \quad [L^{-1}P(F(x^{k+1}) - F(x^k))]_i = 0 \\ &\Leftrightarrow \quad (L^{-1}PF(x^{k+1}))_i - (L^{-1}PF(x^k))_i = 0 \end{aligned} \quad (2.5.7)$$

como $u_i^k s_k = (L^{-1}PF(x^k))_i = 0$, temos que a equação secante só será satisfeita se $(L^{-1}PF(x^{k+1}))_i = 0$.

Porém, $(L^{-1}PF(x^{k+1}))_i$ pode ser não nulo e neste caso, o conjunto S é vazio, uma vez que não existe um vetor linha com a mesma estrutura que u_i^k e que satisfaça a i-ésima equação secante. Caso ocorra esta situação fazemos $u_i^{k+1} = u_i^k$ de modo a atender a condição de Dennis-Marwil e o princípio de variação mínima.

Se $S \neq \phi$, a matriz U_{k+1} obtida conforme (2.5.6), é solução do problema:

$$\left[\begin{array}{l} \min \|U - U_k\|_F \\ \text{S a} \ : \ U \in S \end{array}\right.$$

Algoritmo 2.5.
Dados, x^0 a aproximação inicial, os parâmetros $\alpha > 0$, $\beta > 0$, e Tolsing, execute:

Passo 1: obtenha $B_0 = J(x^0)$

Passo 2: obtenha as matrizes P, L, U_0, tais que:
$$PB_0 = LU_0$$

Passo 3: (salvaguarda contra singularidade)

$$\left[\begin{array}{ll} \text{para} & i = 1, \ldots, n \text{ faça :} \\ \text{se} & |u_{ii}^0| < \text{Tolsing} \max_{ij}\{|e_i^T B_0 e_j|\} \quad \text{faça} \\ & u_{ii}^0 \leftarrow sgn(u_{ii}^0)\ \text{Tolsing} \end{array}\right.$$

Passo 4: resolva $Lw = P(-F(x^k))$
$Us_k = w$

Passo 5: obtenha $\theta = \min\{1,\ \beta/\|s_k\|_\infty\}$
se $\theta \neq 1$, faça $s_k \leftarrow \theta s_k$

Passo 6: faça $x^{k+1} = x^k + s_k$

Passo 7: (cálculo de $t = Us_k$)

faça $t = \theta w$

Passo 8: (atualização da matriz U_k)
execute os passos 8.1 à 8.3

Passo 8.1: faça $\overline{w} = w$
obtenha $w = L^{-1}(P(-F(x^{k+1}))$

Passo 8.2: (cálculo de $L^{-1}(Py) = L^{-1}P(F(x^{k+1}) - F(x^k)))$
faça: $v = \overline{w} - w$

Passo 8.3: (atualização da matriz U_k)

para $i = 1, \ldots, n$ execute :

passo 8.3.1 : obtenha $I_i = \{j \mid u_{ij}^k = 0\}$

passo 8.3.2 : faça $u_{ij}^{k+1} = 0$ se $j \in I_i$

passo 8.3.3 : obtenha o vetor z :
$z_j = s_j$ se $j \notin I_i$
$= 0$ caso contrário
calcule $\gamma = z^T z$

passo 8.3.4 : se $\gamma > \alpha ||s_k||_2$ faça :
$u_{ij}^{k+1} = u_{ij}^k + \dfrac{(v_i - t_i)}{\gamma} s_j \quad j \notin I_i;$
caso contrário
$u_i^{k+1} = u_i^k$

Passo 9: (salvaguarda contra singularidade)

Para $i = 1, \ldots, n$ faça :
se $|u_{ii}^{k+1}| < \text{Tolsing} \max_{ij}\{|e_i^T B_o e_j|\}$ faça :
$u_{ii}^{k+1} \leftarrow sgn(u_{ii}^{k+1})\text{Tolsing}$

Passo 10: $k = k + 1$
volte ao passo 4.

2.6. Métodos Quase Newton com Escalamento da Fatoração

2.6.0. Introdução

Nesta família de métodos, introduzida por Martínez em 1987, [29], a matriz B_{k+1} é obtida implicitamente através da atualização de uma fatoração da matriz B_k . Desta forma, como no método de Dennis - Marwil, a esparsidade da matriz Jacobiana pode ser explorada, a resolução do sistema linear é simplificada, e, os métodos desta família apresentam fórmulas de atualização bastante simples.

A proposta básica é:
conhecida a matriz B_k e as matrizes P_k, C_k, D_k, E_k, tais que:

$$P_k\ B_k = C_k D_k E_k,$$

onde:
P_k : matriz de permutação,
C_k, E_k : matrizes $n \times n$,
D_k : matriz diagonal; obter a matriz B_{k+1}, modificando apenas o fator diagonal, de modo que a equação secante seja satisfeita, isto é:

$$P_k\ B_{k+1} = C_k D_{k+1} E_k \quad \text{e,}$$

$$B_{k+1}\ s_k = y_k$$

$$\text{daí:}\quad P_k\ B_{k+1}\ s_k = P_k\ y_k \Leftrightarrow (C_k\ D_{k+1}\ E_k) s_k = P_k\ y_k \qquad (2.6.0.1)$$

e, D_{k+1}, é obtida através da resolução do sistema linear:

$$d_i^{k+1}\ (E_k\ s_k)_i = (C_k^{-1}\ P_k\ y_k)_i \qquad i = 1, \ldots, n$$

se $(E_k\ s_k)_i \neq 0$:

$$d_i^{k+1} = \frac{(C_k^{-1}\ P_k\ y_k)_i}{(E_k\ s_k)_i} \qquad i = 1, \ldots, n$$

Observamos que ao se impor a condição secante, a matriz D_{k+1} fica unicamente determinada, caso o sistema linear (2.6.0.1) tenha solução. No caso de se ter $(E_k\ s_k)_i = 0$ para algum i, a proposta é definir $d_i^{k+1} = d_i^k$ e, neste caso a equação secante só será satisfeita se o termo correspondente $(C_k^{-1}\ P_k\ y_k)_i$, for nulo.

Formalmente, a família de métodos quase Newton com Escalamento da Fatoração (FQNEF) é assim definida:

Considerando:

i) $F : D \subset I\!R^n \rightarrow I\!R^n$;

ii) $x^0 \in D$

iii) a matriz B_0, não singular;

iv) as matrizes: $C_0 = C(B_0)$,
$D_0 = D(B_0)$, diagonal,
$E_0 = E(B_0)$,
$P_0 = P(B_0)$;

a sequência $\{x^k\}$ é gerada, através de:

$$x^{k+1} = x^k - (C_0 D_k E_0)^{-1} F(x^k) \tag{2.6.0.2}$$

onde

$$D_k = (d_i^k)$$

e, D_{k+1} é obtida por:

definindo: $s_k = x^{k+1} - x^k$ e $y_k = F(x^{k+1}) - F(x^k)$:

se $|[E_0\ s_k]_i| > \alpha ||s_k||$ então

$$d_i^{k+1} = \frac{(C_0^{-1}\ P_0\ y_k)_i}{(E_0\ s_k)_i} \tag{2.6.0.3}$$

caso contrário

$$d_i^{k+1} = d_i^k$$

Neste trabalho foram estudados 3 métodos desta família: Modificação do Fator Diagonal, Escalamento de Linhas e Escalamento de Colunas.

2.6.1. Método de Atualização do Fator Diagonal

Neste método a matriz B_0 é fatorada na forma $L_0 D_0 U_0$, onde L_0 e U_0 são matrizes triangulares, inferior e superior respectivamente, com diagonal unitária e D_0 é matriz diagonal.

A matriz B_{k+1} , $k \geq 0$, é obtida, através da modificação do fator D_k, de modo que a equação secante seja satisfeita, isto é

$$P_0\ B_{k+1} = L_0\ D_{k+1}\ U_0 \tag{2.6.1.1}$$

e

$$B_{k+1}\ s_k = y_k$$

Este é um método da família $FQNEF$, para o qual

$$C(B_0) = L_0$$
$$D(B_0) = D_0$$
$$E(B_0) = U_0$$

Portanto, o fator D_{k+1} é obtido através da resolução do sistema linear, equivalente ao deduzido na secção 2.6.0.

$$(L_0 D_{k+1} U_0) s_k = P_0\ y_k \qquad (2.6.1.2)$$

e, como em 2.6.0.3, teremos que, dada uma tolerância $\alpha > 0$:

$$\begin{aligned} d_i^{k+1} &= \frac{(L_0^{-1}\ P_0\ y_k)_i}{(U_0 s_k)_i} \quad \text{se} |(U_0\ s_k)_i| > \alpha ||s_k|| \\ &= d_i^k \quad \text{caso contrário} \end{aligned}$$

Algoritmo 2.6.1.

Sejam x^0, a aproximação inicial, os parâmetros $\alpha > 0$, $\beta > 0$ e Tolsing, execute:

Passo 1: calcule $F(x^0)$, $J(x^0)$ e faça: $B_0 = J(x^0)$

Passo 2: calcule as matrizes P, L, D_0, U, tais que:
$$PB_0 = LD_0U$$

Passo 3: (salvaguarda contra singularidade)

$$\left[\begin{array}{l} \text{para} \quad i = 1, \ldots, n \\ \text{se} \quad |d_{ii}^0| < \text{Tolsing} \max_{ij}\{|e_i^T\ B_o e_j|\} \\ \qquad d_{ii}^0 \leftarrow sgn(d_{ii}^0)\ \text{Tolsing} \end{array}\right.$$

Passo 4: resolva: $Lr = P(-F(x^0))$

Passo 5: resolva $D_k\ w = r$

$$U s_k = w$$

Passo 6: obtenha $\theta = \min\{1,\ \beta\ /\ ||s_k||_\infty\}$

Passo 7: se $\theta \neq 1$ faça $s_k \leftarrow \theta s_k$
$w \leftarrow \theta w$

Passo 8: faça $x^{k+1} = x^k + s_k$

Passo 9: (atualização do fator diagonal)

passo 9.1 : faça $\bar{r} = r$
obtenha : $r = L^{-1}P(-F(x^{k+1}))$

passo 9.2 : para $i = 1, \ldots, n$ faça :
se $|w_i| > \alpha||s_k||_\infty$ faça
$d_{ii}^{k+1} = (\bar{r}_i - r_i)/w_i$
caso contrário
$d_{ii}^{k+1} = d_{ii}^k$

Passo 10: (salvaguarda contra singularidade)

para $i = 1, \ldots, n$ faça :
se $|d_{ii}^{k+1}| < \text{Tolsing} \max_{ij}\{|e_i^T B_0 e_j|\}$ faça :
$d_{ii}^{k+1} \leftarrow sgn(d_{ii}^{k+1})\,\text{Tolsing}$

Passo 11: $k = k + 1$
volte ao passo 5.

2.6.2. Método de Escalamento de Colunas

Conhecida a matriz B_0, a matriz B_{k+1}, $k \geq 0$ é obtida pós-multiplicando-se B_0 por uma matriz diagonal, D_{k+1}, de modo que $B_{k+1} = B_0 D_{k+1}$ satisfaça a equação secante:

$$B_{k+1}\ s_k = y_k \Leftrightarrow (B_0 D_{k+1}) s_k = y_k$$

A matriz D_{k+1} é obtida através da resolução do sistema linear:

$$D_{k+1} s_k = B_0^{-1} y_k \qquad (2.6.2.1)$$

Este é um método da família $FQNEF$, sendo que:

$$C_0 = B_0$$
$$E_0 = I$$

Uma vez que estamos considerando que a fatoração LU com estratégia de pivoteamento parcial está sendo aplicada à resolução dos sistemas lineares, teremos que B_0 é conhecida através dos seus fatores L e U e da matriz de permutação P, então, o sistema linear (2.6.2.1) pode ser escrito:

$$D_{k+1}s_k = (LU)^{-1}Py_k$$

então, para um parâmetro de tolerância $\alpha > 0$ teremos:

$$\begin{aligned} d_i^{k+1} &= \frac{[(LU)^{-1}\ P\ y_k]_i}{s_i^k} \quad \text{se} \quad |s_i^k| > \alpha||s_k|| \\ &= d_i^k \qquad\qquad\qquad\qquad \text{caso contrário} \end{aligned}$$

Algoritmo 2.6.2.

Sejam x^0, a aproximação inicial, os parâmetros $\alpha > 0$, $\beta > 0$ e Tolsing, execute:

Passo 1: calcule $F(x^0)$, $J(x^0)$ e faça $B_0 = J(x^0)$.

Passo 2: calcular as matrizes P, L, U tais que:
$$PB_0 = LU$$

Passo 3: (salvaguarda contra singularidade)

$$\left[\begin{array}{ll} \text{para} & i = 1, \ldots, n \\ \text{se} & |u_{ii}| < \text{Tolsing} \max_{ij}\{|e_i^T\ B_0 e_j|\} \\ & u_{ii} \leftarrow sgn(u_{ii})\ \text{Tolsing} \end{array}\right.$$

Passo 4: resolva $LUw = P(-F(x^0))$
faça $D_0 = I$.

Passo 5: resolva: $D_k s_k = w$

Passo 6: obtenha $\theta = \min\{1,\ \beta/||s_k||_\infty\}$

se $\theta \neq 1$ faça $s_k \leftarrow \theta s_k$.

Passo 7: calcule $x^{k+1} = x^k + s_k$

Passo 8: (cálculo do fator D_{k+1})

passo 8.1 : faça $\overline{w} = w$
passo 8.2 : obtenha : $w = -(LU)^{-1}PF(x^{k+1})$
passo 8.3 : para $i = 1, \ldots, n$ faça
se $|s_i^k| > \alpha ||s_k||_\infty$
$d_{ii}^{k+1} = (\overline{w}_i - w_i)/s_i^k$
caso contrário
$d_{ii}^{k+1} = d_{ii}^k$

Passo 9: (salvaguarda contra singularidade)

para $i = 1, \ldots, n$
se $|d_{ii}^{k+1}| <$ Tolsing $\max_{ij}\{|e_i^T B_0 e_j|\}$
$d_{ii}^{k+1} \leftarrow sgn(d_{ii}^{k+1})$ Tolsing

Passo 10: faça $k = k + 1$
volte ao passo 5.

2.6.3. Método de Escalamento de Linhas

Conhecida a matriz B_0, a matriz B_{k+1}, $k \geq 0$, é obtida pré-multiplicando-se B_0 por uma matriz diagonal, D_{k+1}, de tal forma que $B_{k+1} = D_{k+1}B_0$ satisfaça a equação secante:

$$B_{k+1}s_k = y_k \Leftrightarrow D_{k+1}(B_0 s_k) = y_k \qquad (2.6.3.1)$$

e, a matriz D_{k+1} é obtida pela resolução deste sistema linear.

Este é um método da família $FQNEF$, pois: $C_0 = I$ e $E_0 = B_0$, e, conforme foi colocado na secção 2.6.0, a matriz D_{k+1} fica unicamente determinada se o sistema linear (2.6.3.1) tiver solução.

Considerando que:

$$B_0\, s_k = D_k^{-1}(-F(x^k))$$

teremos para $\alpha > 0$:

$$\begin{aligned} d_i^{k+1} &= \frac{(y_k)_i}{(-F(x^k))_i} d_i^k \quad \text{se} \quad |(F(x^k))_i| > \alpha \, ||F(x^k)||_\infty \\ &= d_i^k \quad \text{caso contrário} \end{aligned}$$

Algoritmo 2.6.3.

Sejam x^0, a aproximação inicial, e, os parâmetros $\alpha > 0$, $\beta > 0$ e Tolsing, execute:

Passo 1: calcule $F(x^0)$, $J(x^0)$ e faça $B_0 = J(x^0)$.

Passo 2: calcule as matrizes P, L, U tais que:

$$PB_0 = LU$$

Passo 3: (salvaguarda contra singularidade):

$$\left[\begin{array}{ll} \text{para} & i = 1, \ldots, n \\ \text{se} & |u_{ii}| < \text{Tolsing} \max_{ij} \{|e_i^T \, B_0 e_j|\} \quad \text{faça} \\ & u_{ii} \leftarrow sgn(u_{ii}) \, \text{Tolsing} \end{array} \right.$$

Passo 4: faça $D_o = I$

Passo 5: resolva: $D_k t = -F(x^k)$
$LU s_k = Pt$

Passo 6: obtenha $\theta = \min\{1, \ \beta / ||s_k||_\infty\}$
se $\theta \neq 1$ faça $s_k \leftarrow \theta s_k$.

Passo 7: obtenha $x^{k+1} = x^k + s_k$

Passo 8: (cálculo do fator D_{k+1})

$$\left[\begin{array}{l}
\text{passo 8.1 : faça: } v = -\theta F(x^k) \\
\qquad\qquad\qquad\quad w = F(x^{k+1}) - F(x^k) \\
\\
\text{passo 8.2 : para } i = 1, \ldots, n \text{ faça} \\
\qquad\qquad \text{se } |v_i| > \alpha ||F(x^k)||_\infty \\
\qquad\qquad\quad d_{ii}^{k+1} = (w_i/v_i) d_{ii}^k \\
\qquad\qquad \text{caso contrário} \\
\qquad\qquad\quad d_{ii}^{k+1} = d_{ii}^k \\
\\
\text{passo 8.3 : (salvaguarda contra singularidade) :} \\
\qquad\qquad \text{para } i = 1, \ldots, n \text{ faça} \\
\qquad\qquad \text{se } |d_{ii}^{k+1}| < \text{ Tolsing } \max_{ij}\{|e_i^T B_0 e_j|\} \\
\qquad\qquad\quad d_{ii}^{k+1} \leftarrow sgn(d_{ii}^{k+1}) \text{ Tolsing}
\end{array}\right.$$

Passo 9: $k = k + 1$
volte ao passo 5.

2.7. Método de Atualização de uma Coluna por Iteração (ACI)

Proposto por Martínez, em 1983, [30], este método obtém a matriz B_{k+1} através da atualização de uma coluna de B_k, de modo que B_{k+1} satisfaça a equação secante: $B_{k+1}s_k = y_k$.

Conforme veremos, desta motivação resulta que B_{k+1} é uma matriz da forma: $B_k + uv^T$ e, por esta razão, como no método de Broyden, a implementação no caso esparso é feita com a aplicação da fórmula de Sherman-Morrison para o cálculo de B_{k+1}^{-1}.

Conhecidos então, os vetores: x^k, x^{k+1}, $F(x^k)$, e $F(x^{k+1})$, e a matriz B_k, o cálculo de B_{k+1} se inicia escolhendo o índice j_k da coluna de B_k que será modificada. Ficará claro, abaixo que uma escolha conveniente é o índice j_k, para o qual $|s_{j_k}^k| > \alpha ||s_k||_\infty$, $\alpha > 0$.

Como só a coluna j_k é modificada, podemos escrever B_{k+1} como

$$B_{k+1} = B_k + u_k e_{j_k}^T$$

onde o vetor u_k deve ser tal que a equação secante seja satisfeita:

$$B_{k+1}s_k = y_k \quad \Leftrightarrow \quad (B_k + u_k e_{j_k}^T)s_k = y_k \Leftrightarrow u_k e_{j_k}^T s_k = y_k - B_k s_k \qquad (2.7.1)$$
$$\Leftrightarrow \quad u_k = \frac{y_k - B_k s_k}{e_{j_k}^T s_k}, \text{ se } e_{j_k}^T s_k \neq 0$$

Daí,

$$B_{k+1} = B_k + \frac{y_k - B_k s_k}{e_{j_k}^T s_k} e_{j_k}^T \qquad (2.7.2)$$

Usando agora a fórmula de Sherman–Morrison, a inversa de $B_{k+1} = B_k + u_k\, e_{j_k}^T$, existe, se B_k for inversível e, se $(1 + e_{j_k}^T\, B_k^{-1} u_k) \neq 0$, e, é dada por:

$$B_{k+1}^{-1} = B_k^{-1} - \frac{B_k^{-1} u_k e_{j_k}^T B_k^{-1}}{1 + e_{j_k}^T\, B_k^{-1} u_k} \qquad (2.7.3)$$

Então,

$$B_{k+1}^{-1} = B_k^{-1} - \frac{B_k^{-1}(\frac{y_k - B_k s_k}{e_{j_k}^T s_k}) e_{j_k}^T B_k^{-1}}{1 + e_{j_k}^T B_k^{-1}(\frac{y_k - B_k s_k}{e_{j_k}^T s_k})}$$

Daí,

$$B_{k+1}^{-1} = B_k^{-1} + \frac{(s_k - B_k^{-1} y_k) e_{j_k}^T B_k^{-1}}{e_{j_k}^T B_k^{-1} y_k} \qquad (2.7.4)$$

Definindo,

$$w_k = \frac{(s_k - B_k^{-1} y_k)}{e_{j_k}^T B_k^{-1} y_k} \qquad (2.7.5)$$

a fórmula (2.7.4) pode ser escrita:

$$B_{k+1}^{-1} = (I + w_k e_{j_k}^T) B_k^{-1} \qquad (2.7.6)$$

e, aplicando recursivamente (2.7.5) - (2.7.6), temos:

$$B_{k+1}^{-1} = (I + w_k e_{j_k}^T)(I + w_{k-1} e_{j_{k-1}}^T) \ldots (I + w_0 e_{j_0}^T) B_0^{-1} \qquad (2.7.7)$$

Podemos notar que a cada iteração k efetuada é preciso armazenar um vetor adicional: w_k e um índice: j_k , e, portanto, este método é mais econômico, em

termos de memória, que o método de Broyden, que requer dois vetores adicionais por iteração.

De qualquer forma, o mínimo de iterações consecutivas é limitado pelo espaço de memória disponível para armazenar os vetores w_k.

Considerando que há espaço para armazenar m vetores w_k e m índices j_k, é possível efetuar uma iteração Newton e m iterações ACI consecutivas; neste caso, a expressão (2.7.7) deve ser escrita:

$$B_{k+1}^{-1} = (I + w_k e_{j_k}^T)\,(I + w_{k-1} e_{j_{k-1}}^T).....(I + w_\ell e_{j_\ell}^T) B_\ell^{-1} \qquad (2.7.8)$$

onde $\ell \equiv 0(\text{mod}.(m+1))$

Algoritmo 2.7.

Dados x^0, a aproximação inicial e os parâmetros $\beta > 0$, e Tolsing, e o número inteiro m, execute:

Passo 0: $k = 0,\ \ell = 0$

Passo 1: obtenha $B_k = J(x^k)$.

Passo 2: obtenha as matrizes P, L, U tais que:
$$PB_k = LU$$

Passo 3: resolva $LU\tilde{s}_k = P(-F(x^k))$

Passo 4: obtenha: $\theta = \min\{1,\ \beta/||s_k||_\infty\}$
faça $s_k = \theta\tilde{s}_k$

Passo 5: faça $x^{k+1} = x^k + s_k$

Passo 6: obter q tal que $k \equiv q(\text{mod}\ (m+1))$
Se $k \neq \ell$ e $q = 0$, faça: $k = k + 1$
$\ell = k$
volte ao passo 1.
caso contrário:

Passo 7: (cálculo do vetor w_k, de acordo com a fórmula 2.7.5)

execute os passos 7.1 à 7.4.

Passo 7.1: obtenha $j_k = \arg\max_j\{|e_j^T s_k|\}$

Passo 7.2: (cálculo de $t = -B_k^{-1}\, F(x^{k+1})$):

$$\left[\begin{array}{ll} \text{Passo 7.2.1 :} & \text{(resolução de } \quad B_\ell t = -F(x^{k+1}) \\ & \text{Resolva } \quad LUt = -PF(x^{k+1}) \\ \\ \text{Passo 7.2.2 :} & \text{para } \quad r = \ell, \ldots (k-1) \text{faça:} \\ & t \leftarrow (I + w_r e_{j_r}^T)t \end{array}\right.$$

Passo 7.3: (cálculo de $v_k = B_k^{-1} y_k = B_k^{-1}(F(x^{k+1})) - F(x^k)$:

$$v_k = \tilde{s}_k - t$$

Passo 7.4: faça: $w_k = (s_k - v_k)/e_{j_k}^T v_k$

Passo 8: (completa o cálculo do produto $-B_{k+1}F(x^{k+1})$)
faça $\tilde{s}_k = t + w_k e_{j_k}^T t$

Passo 9: $k = k + 1$
volte ao passo 4.

Através da fórmula (2.7.5) para w_k , podemos ver que B_{k+1} será inversível se $e_{j_k}^T B_k^{-1} y_k \neq 0$ o que equivale a pedir que $e_{j_k}^T v_k \neq 0$. A salvaguarda contra singularidade, na implementação do algoritmo, consiste em testar se:

$$|e_{j_k}^T v_k| < \text{Tolsing}||v_k||_\infty$$

Caso ocorra esta situação, colocamos $B_{k+1} = B_k$ e prosseguimos a execução.

CAPÍTULO 3

CONVERGÊNCIA LOCAL

Neste capítulo descrevemos inicialmente alguns conceitos e resultados importantes a respeito da convergência local dos métodos colocados no capítulo 2. Em seguida, apresentamos e discutimos o resultado de convergência de cada método, ou de um grupo de métodos de uma mesma família quase Newton.

Os resultados aqui colocados se referem sempre aos algoritmos sem salvaguardas contra singularidades e, sem controle sobre o tamanho do passo.

3.1. Definições, Hipóteses e Lemas Básicos

Definição 3.1.1. Dizemos que a função $F : D \subset \mathbb{R}^n \to \mathbb{R}^n$, D um conjunto aberto e convexo, é de classe C^1 sobre D, se, cada componente $f_i, i = 1, \ldots, n$ é de classe C^1 sobre D. Se

$$\nabla f_i(x) = (\frac{\partial f_i}{\partial x_1}(x), \ldots, \frac{\partial f_i}{\partial x_n}(x))^T, \text{ então :}$$

$J(x) = \nabla F(x)^T$ denota a matriz Jacobiana de F em x : $J : D \subset \mathbb{R}^n \to \mathbb{R}^{n \times n}$.

Definição 3.1.2. Dizemos que $J : D \subset \mathbb{R}^n \to \mathbb{R}^{n \times n}$ é Lipschitz contínua em x, se existe um conjunto aberto $D_1 \subset D$, $x \in D_1$ e uma constante $L > 0$, tal que, para todo $v \in D_1$:

$$||J(v) - J(x)|| \leq L||v - x|| \tag{3.1.1}$$

L é chamada constante de Lipschitz para J em x.
Se (3.1.1) vale para qualquer $x \in D_1$ então J é Lipschitz contínua em D_1.

Definição 3.1.3. Seja $\{x^k\} \subset \mathbb{R}^n$ uma sequência convergente a x^*. Então dizemos que $\{x^k\}$ converge:
i) linearmente a x^* se existe uma constante $r \in (0,1)$ e um inteiro $\overline{k} \geq 0$, tal que

para todo $k \geq \overline{k}$:

$$||x^{k+1} - x^*|| \leq r||x^k - x^*||$$

ii) superlinearmente à x^*, se para alguma sequência $\{r_k\}$ que converge a zero:

$$||x^{k+1} - x^*|| \leq r_k||x^k - x^*||$$

ou, equivalentemente, se:

$$\lim_{k\to\infty} \frac{||x^{k+1} - x^*||}{||x^k - x^*||} = 0$$

iii) com taxa quadrática à x^*, se existem constantes $r \geq 0$, e $\overline{k} \geq 0$ tais que para todo $k \geq \overline{k}$:

$$||x^{k+1} - x^*|| \leq r||x^k - x^*||^2$$

Neste trabalho, sempre que nos referirmos às taxas de convergência acima definidas diremos: taxa linear, superlinear e quadrática, respectivamente.

Algumas hipóteses são comuns a vários teoremas. Para simplificar o enunciado destes teoremas, nos referimos às estas hipóteses por: $H_1, H_2, \ldots, H_5$ onde:

$H_1 : F : D \subset I\!R^n \to I\!R^n$;

$H_2 : F \in C^1(D)$;

$H_3 : x^* \in D, F(x^*) = 0$;

$H_4 : J_* = J(x^*)$ não singular;

$H_5 : J(x)$ é Lipschitz contínua em x^*, no conjunto D; isto é, existe $L > 0$, tal que para qualquer $v \in D$:

$$||J(v) - J(x^*)|| \leq L||v - x^*||$$

Os lemas abaixo resultam em relações importantes, satisfeitas por $F(x)$ e $J(x)$ sob certas condições.

Lema 3.1.1. Supondo as hipóteses H_1 e H_2; para qualquer $x, x + s \in D$:

$$F(x + s) - F(x) = \int_0^1 J(x + ts)s\, dt$$

Dem.: [11]

Lema 3.1.2. Sob as hipóteses H_1, H_2 e H_5. Para quaisquer $u, v \in D$:

$$||F(v) - F(u) - J_*(v - u)|| \leq L||v - u||\text{máx}\{||v - x^*||,\ ||u - x^*||\}$$

Dem.: [23]

Lema 3.1.3. Sob as hipóteses $H_1, H_2, e H_5$. Para qualquer $v \in D$:

$$||F(v) - F(x^*) - J_*(v - x^*)|| \leq \frac{L}{2}||v - x^*||^2 \Leftrightarrow ||F(v) - J_*(v - x^*)|| \leq \frac{L}{2}||v - x^*||^2$$

Dem.: [11]

O lema seguinte reúne os resultados de dois lemas importantes: Lema de Newman e o Lema da Perturbação:

Lema 3.1.4. Supor que $||.||$ é qualquer norma de matrizes sobre $\mathbb{R}^{n\times n}$, tal que:
i) $\forall\, A$ e $B \in \mathbb{R}^{n\times n}$, $||AB|| \leq ||A||\,||B||$ e
ii) $||I|| = 1$
Seja a matriz $M \in \mathbb{R}^{n\times n}$. Se $||M|| < 1$, então $(I - M)^{-1}$ existe e :

$$||(I - M)^{-1}|| \leq \frac{1}{1 - ||M||}$$

Supor A não singular e $||A^{-1}(B - A)|| < 1$. Então, B é não singular e

$$||B^{-1}|| \leq \frac{||A^{-1}||}{1 - ||A^{-1}(B - A)||}$$

Dem.: [11]

3.2. Teorema das Vizinhanças

A condição central para se obter a taxa linear de convergência, é pedir que as aproximações x^k e B_k permaneçam em vizinhanças convenientes de x^* e J_*, pois:
i) supondo J_* não singular e $J(x)$ contínua em D, existe uma vizinhança em torno de J_* de matrizes não singulares;
ii) analisando o erro: $e^{k+1} = x^{k+1} - x^*$, temos que:

$$x^{k+1} - x^* = x^k - B_k^{-1}F(x^k) - x^* = (x^k - x^*) - B_k^{-1}[J_*(x^k - x^*) + O(e^2)]$$

sendo que a última igualdade vem da aproximação de $F(x)$, por Taylor em torno de x^*; assim:

$$x^{k+1} - x^* = (I - B_k^{-1}\, J_*)\,(x^k - x^*) + O(e^2)$$

e, então, uma das condições para se garantir a convergência é que:

$$B_k^{-1} J_* \sim I \quad \text{e, daí,} \quad B_k \sim J_*$$

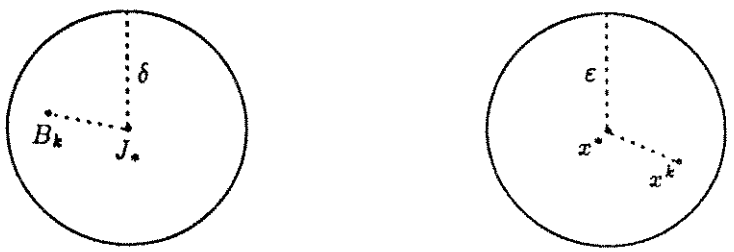

Figura 3.2.1

O teorema das vizinhanças, [23], estabelece a convergência de $\{x^k\}$ à x^* com taxa linear supondo que $\{x^k\}$ e $\{B_k\}$ permanecem em vizinhanças convenientes de x^* e J_*, respectivamente. As condições para se garantir esta suposição é uma questão específica de cada método.

No teorema seguinte, para simplificar, denotaremos x^k por x e B_k por B.

Teorema 3.2.1. Sob as hipóteses: H_1 à H_4. Para cada $x \in D$ definir: $\varphi(x,\ B) = x - B^{-1}F(x)$. Seja $r \in (0,\ 1)$. Então existem $\varepsilon_1,\ \delta_1 > 0$, tais que, se $||x - x^*|| \leq \varepsilon_1$ e $||B - J_*|| \leq \delta_1$, a função $\varphi(x,\ B)$ está bem definida e satisfaz:

$$||\varphi(x,\ B) - x^*|| \leq r||x - x^*||$$

Dem.:
Seja $\delta_1' = \dfrac{1}{2||J_*^{-1}||}$

Daí, se $||B - J_*|| \leq \delta_1'$, vem que:

$$||J_*^{-1}(B - J_*)|| \leq ||J_*^{-1}||\ ||(B - J_*)|| \leq \frac{1}{2}.$$

Então pelo lema 3.1.4, B^{-1} existe e:

$$||B^{-1}|| \leq 2||J_*^{-1}||$$

Então, se $x \in D$ e $\delta_1 \leq \delta_1'$, $\varphi(x,\ B)$ está bem definida.
Agora,

$$\begin{aligned} ||\varphi(x,\ B) - x^*|| &= ||(x - B^{-1}F(x)) - x^*|| = \\ &= ||x - x^* - B^{-1}F(x) + B^{-1}J_*(x - x^*) - B^{-1}J_*(x - x^*)|| \\ &= ||x - x^* - B^{-1}J_*(x - x^*) - B^{-1}(F(x) - J_*(x - x^*))|| \\ &\leq \underbrace{||x - x^* - B^{-1}J_*(x - x^*)||}_{A_1} + \underbrace{||B^{-1}(F(x) - J_*(x - x^*))||}_{A_2} \end{aligned}$$

Analisando A_1:

$$\begin{aligned} A_1 &= ||x - x^* - B^{-1}J_*(x - x^*)|| \\ &= ||B^{-1}(B - J_*)(x - x^*)|| \\ &\leq ||B^{-1}||||B - J_*||||x - x^*|| \\ &\leq 2||J_*^{-1}||\delta_1||x - x^*|| \end{aligned}$$

Analisando A_2:

$$\begin{aligned} A_2 &= ||B^{-1}(F(x) - J_*(x - x^*))|| \leq ||B^{-1}||\ ||F(x) - J_*(x - x^*)|| \\ &\leq 2||J_*^{-1}||\ ||F(x) - J_*(x - x^*)|| \end{aligned}$$

(e, pela definição de derivada de Fréchet em x^*):

$$= 2||J_*^{-1}||\ \beta(x)$$

onde $\lim\limits_{x \to x^*} \dfrac{\beta(x)}{||x - x^*||} = 0$

Daí:

$$||\varphi(x, \beta) - x^*|| \leq 2||J_*^{-1}||\ \delta_1||x - x^*|| + 2||J_*^{-1}||\ \beta(x)$$

Escolhendo ε_1, δ_1, tais que:

$$2(\delta_1 + \sup_{||x - x^*|| \leq \varepsilon_1} \{\frac{\beta(x)}{||x - x^*||}\}) \leq \frac{r}{||J_*^{-1}||}$$

para $||B - J_*|| \leq \delta_1$ e $||x - x^*|| \leq \varepsilon_1$, temos:

$$||\varphi(x,\ B) - x^*|| \leq ||J_*^{-1}||(2\delta_1\ ||x - x^*|| + 2\beta(x))$$

$$
\begin{aligned}
&= \|J_*^{-1}\|(2\delta_1 + \frac{2\beta(x)}{\|x - x^*\|})\|x - x^*\| \\
&\leq \|J_*^{-1}\|\frac{r}{\|J_*^{-1}\|}\|x - x^*\| \\
&= r\|x - x^*\|
\end{aligned}
$$

□

A seguir, $V(x^*, \varepsilon)$ e $V(J_*, \delta)$ representam os conjuntos: $\{x \in D \subset I\!R^n \mid \|x - x^*\| < \varepsilon\}$ e $\{B \in I\!R^{n\times n} \mid \|B - J_*\| < \delta\}$, respectivamente.

Teorema 3.2.2. Considere a sequência $\{x^k\}$ definida por:

$$x^{k+1} = x^k - B_k^{-1}F(x^k)$$

Supor que $x^0 \in V(x^*, \varepsilon_1)$ e $B_k \in V(J_*, \delta_1)$ para todo $k = 0, 1, 2, \ldots$.
Então, a sequência $\{x^k\}$ está bem definida, converge a x^* e satisfaz:

$$\|x^{k+1} - x^*\| \leq r\|x^k - x^*\| \qquad k = 0, 1, 2, \ldots$$

Dem.: Mostramos por indução em k, que $x^k \in V(x^*, \varepsilon_1)$, $\forall k = 0, 1, \ldots$, e daí a tese decorre do teorema 3.2.1.

Para $k = 0$: $x^0 \in V(x^*, \varepsilon_1)$ e $B_0 \in V(J_*, \delta_1)$, então: $x^1 = x^0 - B_0^{-1}F(x^0)$ está bem definido.

Usando o teorema das vizinhanças: dado $r \in (0, 1)$, é possível escolher:

$$\varepsilon_1 = \varepsilon_1(r) \quad \text{e} \quad \delta_1(r) = \delta_1$$

tais que:

$$\|x^1 - x^*\| \leq r\|x^0 - x^*\|$$

e, portanto, $x^1 \in V(x^*, \varepsilon_1)$.

A demonstração do passo de indução é idêntica uma vez que por hipótese, $B_k \in V(J_*, \delta_1)$ para todo k.

3.3. Propriedade de Deterioração Limitada

O teorema das Vizinhanças assume que para $r \in (0,\ 1)$, as sequências $\{x^k\}$ e $\{B_k\}$ permanecem em vizinhanças convenientes de x^* e J_* : $V(x^*,\ \varepsilon(r))$ e $V(J_*,\ \delta(r))$, respectivamente.

A questão é: sob que condições as matrizes B_k permanecem em $V(J_*,\ \delta(r))$?

A condição mais forte que se pode esperar é a denominada "condição de consistência":

$$\lim_{k\to\infty} B_k = J_* \tag{3.3.1}$$

Esta condição além de garantir naturalmente que $\{B_k\} \subset V(J_*,\ \delta(r))$, é uma condição suficiente para se obter a taxa superlinear de convergência.

Os métodos quase Newton não são em geral consistentes, mas, com condições mais fracas sobre as matrizes B_k, demonstram-se resultados de convergência local.

Uma importante condição é a propriedade de Deterioração Limitada, que consiste em pedir que se houver deterioração nas aproximações B_k em relação à J_*, que esta deterioração ocorra de forma controlada de modo a não comprometer a convergência de $\{x^k\}$ à x^*. Em 1973, Broyden, Dennis e Moré, [4], mostraram que esta condição é suficiente para que o método seja localmente convergente.

O estudo da convergência local de vários métodos quase Newton incluem um teorema que mostra que a sequência $\{B_k\}$ obedece tal princípio; basicamente, deve -se conseguir uma relação entre os erros nas matrizes B_k, da forma:

$$||B_{k+1} - J_*|| \leq ||B_k - J_*|| + \gamma_k \tag{3.3.2}$$

Exemplos:

(1) Método de Broyden

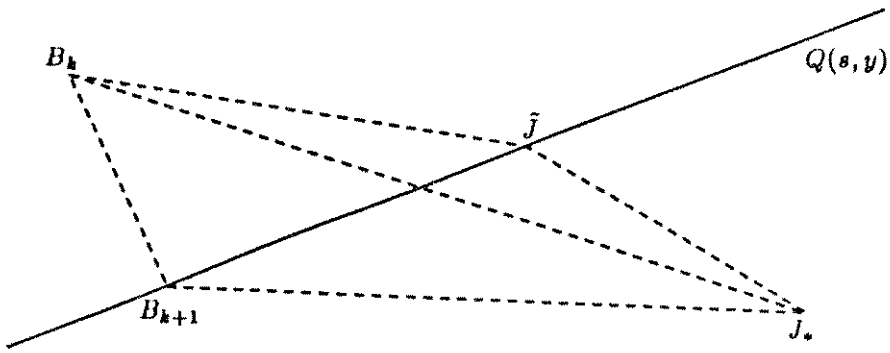

Figura 3.3.1

$\tilde{J} = [\nabla^T f_i(\xi_i)]$ onde para $i = 1, \ldots, n$, ξ_i está entre x^k e x^{k+1} e são tais que $\tilde{J}s = y$.
B_{k+1}: projeção ortogonal da matriz B_k no conjunto $Q(s, y)$.

Da figura 3.3.1 vem que:

$$\begin{aligned} ||B_{k+1} - J_*||_F &\leq ||B_{k+1} - \tilde{J}||_F + ||\tilde{J} - J_*||_F \\ &\leq ||B_k - \tilde{J}||_F + ||\tilde{J} - J_*||_F \\ &\leq ||B_k - J_*||_F + 2||\tilde{J} - J_*||_F \\ &\leq ||B_k - J_*||_F + 2L\text{máx}\{||x^{k+1} - x^*||,\ ||x^k - x^*||\} \quad (3.3.3) \end{aligned}$$

2) Método de Schubert :

No método de Schubert é considerado o conjunto S:
$S = \{B \in I\!R^{n\times n} | Bs = y$ e B conserva a estrutura constante de $J\}$.
Vimos na seção 2.4 que $S \neq \phi$, e, daí a relação 3.3.3 é também válida para o método de Schubert.

Da relação 3.3.3, vem que γ_k depende do máx $\{||x^{k+1} - x^*||,\ ||x^k - x^*||\}$, o que indica que o fator de deterioração nas aproximações B_k diminui a medida em

que x^k se aproxima de x^*. Daí, a possibilidade de se obter vizinhanças convenientes de x^* e J_* de modo a garantir a hipótese do teorema das vizinhanças.

3) Método de Dennis Marwil

Para $S = \{U \in \mathbb{R}^{n\times n} |$ i) $u_{ij} = 0$ quando $i < j$ ou $u^k_{ij} = 0$; ii) $Us = L^{-1}Py\}$

considerando o caso $S \neq \phi$:

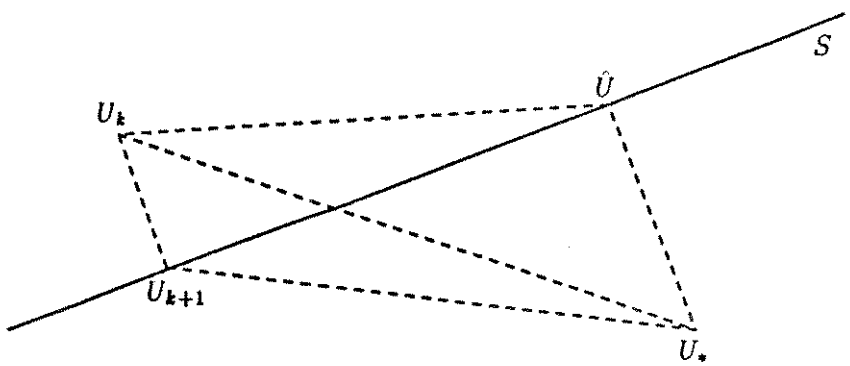

Figura 3.3.2

$\hat{U}$: projeção ortogonal de U_* no conjunto S
U_{k+1}: idem, em relação à U_k.

Como no exemplo 1, obtém - se que:

$$||U_{k+1} - U_*||_F \leq ||U_k - U_*||_F + 2||\hat{U} - U_*||_F \qquad (3.3.4)$$

Não é difícil mostrar que $||\hat{U} - U_*||_F$ é $O(||L_0 - L_*||_F)$ o que significa que a deterioração nas aproximações se dá a passos fixos. Por esta razão, a convergência local do método de Dennis - Marwil só é estabelecida ao se incorporar recomeços, isto é, $B_k = J(x^k)$ se $k \equiv 0$ mod (m) sendo que m depende de $||L_0 - L_*||$ e do raio δ de $V(J_*, \delta)$.

3.4. Taxa Superlinear: Condições - Propriedades

Em alguns métodos quase Newton, estabelecida a taxa linear, prova - se também que r, $r \in (0, 1)$, se acelera de modo que:

$$\lim_{k\to\infty} \frac{||x^{k+1} - x^*||}{||x^k - x^*||} = 0 \tag{3.4.1}$$

isto é, a taxa de convergência do método é superlinear.

Na secção anterior colocamos que uma condição suficiente para (3.4.1) é que $\lim B_k = J_*$, contudo, a taxa superlinear pode ser obtida sob hipóteses mais fracas.

No trabalho de Broyden, Dennis e Moré, [4], é feito um estudo sobre a convergência de uma classe de métodos onde as fórmulas para B_{k+1} são obtidas através de correções de posto um ou dois sobre B_k. A taxa superlinear é obtida sem verificar a condição de consistência. Um resultado apresentado neste trabalho, afirma que sob as hipóteses básicas H_1 à H_5 e, se as matrizes B_k satisfazem as propriedades de Deterioração Limitada, então, se alguma subsequência de $\{||B_k - J_*||\}$ converge a zero a sequência $\{x^k\}$ converge superlinearmente a x^*.

Este resultado é importante pois ao se incorporar recomeços com o Jacobiano verdadeiro aos algoritmos puros, os métodos com taxa linear passam a ter taxa superlinear de convergência.

A condição central sobre as matrizes B_k, necessária e suficiente na caracterização da taxa superlinear é a condição de Dennis e Moré [9], que consiste em pedir que a "ação de B_k sobre s_k", $B_k s_k$, se se aproxime da 'ação de J_* sobre s_k", $J_* s_k$:

Teorema 3.4.1. Sob as hipóteses H_1 à H_5. Considere a sequência $\{B_k\}$ de matrizes não singulares em $\mathbb{R}^{n\times n}$ e, suponha que para algum $x^o \in D \subset \mathbb{R}^n$, a sequência $\{x^k\}$ gerada por:

$$x^{k+1} = x^k - B_k^{-1} F(x^k)$$

permanece no conjunto D e converge à x^*.
Então, $\{x^k\}$ converge a x^*, com taxa superlinear, se e somente se:

$$\lim_{k\to\infty} \frac{||(B_k - J_*)s_k||}{||s_k||} = 0 \tag{3.4.2}$$

Dem.: [9]

□.

Uma última observação a respeito dos métodos com taxa superlinear é o fato de $||s_k||$ se aproximar de $||e^k||$ quando $k \to \infty$. Esta propriedade é importante computacionalmente, pois ao se verificar o teste de parada com $||s_k||$ pode-se ter a segurança de se ter uma boa aproximação para x^*.

Graficamente, a propriedade é facilmente verificada:

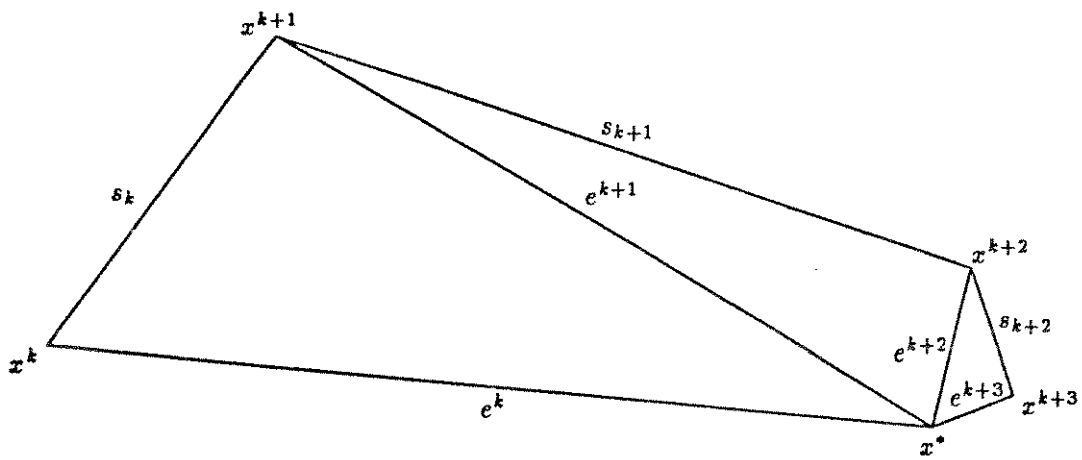

Figura 3.4.1

pois, se $\lim_{k\to\infty} \frac{||e^{k+1}||}{||e^k||} = 0$ então $\lim_{k\to\infty} \frac{||s^k||}{||e^k||} = 1$

Algebricamente:

$$0 = \lim_{k\to\infty} \frac{||e^{k+1}||}{||e^k||} = \lim_{k\to\infty} \frac{||x^{k+1} - x^* + x^k - x^k||}{||e^k||} =$$

$$= \lim_{k\to\infty} \frac{||s^k + e^k||}{||e^k||} \geq \lim_{k\to\infty} |\frac{||s^k|| - ||e^k||}{||e^k||}| = \lim_{k\to\infty} |\frac{||s^k||}{||e^k||} - 1| \geq 0$$

3.5. Convergência do Método de Newton

Nesta secção, demonstraremos a taxa de convergência quadrática do método de Newton usando o teorema das vizinhanças.

Teorema 3.5.1. Supor as hipóteses H1 a H5. Seja: $\varepsilon_1 > 0$, tal que $V(x^*, \varepsilon_1) \subset D$ e $\beta > 0$ tal que $\|J_*^{-1}\| \leq \beta$. Então, dado $r \in (0,1)$, existe $\varepsilon = \varepsilon(r)$, $0 < \varepsilon \leq \varepsilon_1$, tal que, se $x^0 \in V(x^*, \varepsilon)$, a sequência $\{x^k\}$ gerada por:

$$x^{k+1} = x^k - J(x^k)^{-1} F(x^k)$$

está bem definida e satisfaz:

$$\|x^{k+1} - x^*\| \leq r\|x^k - x^*\| \tag{3.5.1}$$

além disto, existe $c > 0$, tal que:

$$\|x^{k+1} - x^*\| \leq c\|x^k - x^*\|^2 \tag{3.5.2}$$

Dem: Seja $\varepsilon_2 = \min\{\varepsilon_1, \frac{1}{2\beta L}\}$.
Se $x^k \in V(x^*, \varepsilon_2)$:

$$\|J_*^{-1}(J(x^k) - J_*)\| \leq \|J_*^{-1}\|\|J(x^k) - J_*\| \leq \beta L\|x^k - x^*\| \leq \frac{1}{2}$$

assim, $\|J(x^k) - J_*\| \leq \dfrac{1}{2\|J_*^{-1}\|} = \delta_1'$ e pelo lema 3.1.4., $J(x^k)$ é não singular e,

$$\|J(x^k)^{-1}\| \leq \frac{\|J_*^{-1}\|}{1 - \|J_*^{-1}(J(x^k) - J_*\|} \leq 2\beta$$

e, portanto, x^{k+1} está bem definido.

Através do teorema das vizinhanças e do teorema 3.2.2, dado $r \in (0,1)$ é possível obter $\varepsilon = \varepsilon(r)$, $0 \leq \varepsilon \leq \varepsilon_2$ e $\delta_1 = \delta_1'(r)$, $0 \leq \delta_1 \leq \delta_1'$ tais que se $x^0 \in V(x^*, \varepsilon)$, (3.5.1) se verifica.

Agora, retomando a demonstração do teorema das vizinhanças, substituíndo a matriz B_k, por $J(x^k)$ e usando a hipótese que $J(x)$ é Lipschitz contínua em x^*,

temos:
se $x^k \in V(x^*, \varepsilon)$:
$\|x^{k+1} - x^*\| \leq \|J(x^k)^{-1}\| \, \|J(x^k) - J_*\| \, \|x^k - x^*\| + \|J(x^k)^{-1}\| \, \|F(x^k) - J_*(x^k - x^*)\|$

aplicando o lema 3.1.3:

$$\|x^{k+1} - x^*\| \leq 2\beta L \|x^k - x^*\|^2 + 2\beta \frac{L}{2} \|x^k - x^*\|^2 = 3\beta L \|x^k - x^*\|^2$$

Então, se $x^k \in V(x^*, \varepsilon)$, x^{k+1} está bem definido e satisfaz (3.5.2). Supondo que $x^0 \in V(x^*, \varepsilon)$ a demonstração da tese segue como no teorema 3.2.2.

□

3.6. Convergência do Método de Newton Modificado

A demonstração da taxa linear do método de Newton Modificado é uma aplicação quase direta do teorema das Vizinhanças:

Teorema 3.6.1. Supor as hipóteses H1 à H5. Seja $\varepsilon_1 > 0$, tal que $V(x^*, \varepsilon_1) \subset D$ e $\beta > 0$, tal que $\|J_*^{-1}\| \leq \beta$. Então, dado $r \in (0,1)$ existe $\varepsilon = \varepsilon(r)$, $0 < \varepsilon \leq \varepsilon_1$, tal que se $x^0 \in V(x^*, \varepsilon)$, a sequência $\{x^k\}$ gerada por:

$$x^{k+1} = x^k - J(x^0)^{-1} F(x^k)$$

está bem definida, e satisfaz:

$$\|x^{k+1} - x^*\| \leq r \|x^k - x^*\|$$

Dem.: Seja $\varepsilon_2 = \min\{\varepsilon_1, \frac{1}{2\beta L}\}$
Se $x^0 \in V(x^*, \varepsilon_2)$:

$$\|J_*^{-1}(J(x^0) - J_*)\| \leq \|J_*^{-1}\| \, \|J(x^0) - J_*\| \leq \beta L \|x^0 - x^*\| \leq \frac{1}{2}$$

daí, pelo lema 3.1.4, $J(x^0)$ é não singular e,

$$\|J(x^0)^{-1}\| \leq 2\|J_*^{-1}\| \leq 2\beta$$

Então, considerando que $B_k = J(x^0)$, para todo $k \geq 0$, temos que:

$$\|B_k - J_*\| \leq \frac{1}{2\|J_*^{-1}\|} = \delta \qquad k = 0, 1, 2, \ldots$$

e, a tese segue do teorema 3.2.2.

□

A taxa superlinear é obtida incorporando recomeços com o Jacobiano ao algoritmo puro, isto é, para $k \in \overline{K}$, $B_k = J(x^k)$, onde $\overline{K}$ é um conjunto infinito de índices.

3.7. Convergência do Método de Broyden e do Método de Schubert

Em 1988, Martínez [32], introduziu uma família de métodos Quase Newton, que apresentam resultados de convergência superlinear. Neste trabalho, denominaremos esta família por FQNSL.

Verificaremos, nesta secção, que os métodos de Broyden e Schubert pertencem a esta família, e na secção 3.8, que o método de Dennis Marwil é o caso limite de uma sub–família de FQNSL, denominada métodos quase Dennis Marwil.

Esta nova família, é assim definida:

considere: $F : D \subset \mathbb{R}^n \to \mathbb{R}^n$;
$0 < \alpha < 1$;
S_A e S_R variedades afim de $\mathbb{R}^{n \times n}$;
$x^0 \in D$;
$A_0 \in S_A$ e $R_0 \in S_R$;
A_0, R_0 matrizes não singulares;

a sequência $\{x^k\}$ é gerada, nos métodos pertencentes a esta família, por:

$$\begin{aligned} x^{k+1} &= x^k - B_k^{-1} F(x^k) \\ B_k &= A_k^{-1} R_k \end{aligned} \tag{3.7.1}$$

sendo que (A_{k+1}, R_{k+1}) é obtida à partir de (A_k, R_k) através da resolução do pro-

blema:

$$\left[\begin{array}{ll} \text{Min}: & \alpha\|A-A_k\|_a^2+(1-\alpha)\|R-R_k\|_b^2 \qquad (3.7.2) \\ \text{s.a.}: & \left[\begin{array}{ll} Rs=Ay & (3.7.3) \\ A\in S_A \text{ e } R\in S_R & (3.7.4) \\ \text{sendo que}: s=x^{k+1}-x^k \quad \text{e} \quad y=F(x^{k+1})-F(x^k) & \end{array}\right. \end{array}\right.$$

Consideraremos $\|\cdot\|_a=\|\cdot\|_b=\|\cdot\|_F$.

Os conjuntos S_A e S_R podem ser definidos por uma matriz θ^A , θ^R e n conjuntos de índices I_i^A , I_i^R, $i=1,\ldots,n$ tais que

$$\begin{aligned} S_A &= \{A\in I\!R^{n\times n} |\ a_{ij}=\theta_{ij}^A \text{ para todo } j\notin I_i^A\} \text{ e} \\ S_R &= \{R\in I\!R^{n\times n} |\ r_{ij}=\theta_{ij}^R \text{ para todo } j\notin I_i^R\} \end{aligned}$$

No método de Broyden, temos que:

$$\begin{array}{lll} A_k=I \quad \forall k=0,1,2\ldots & & \\ I_i^A=\phi & i=1,\ldots,n & \theta^A=I \\ I_i^R=\{1,\ldots,n\} & i=1,\ldots,n & S_R=I\!R^{n\times n} \end{array}$$

No método de Schubert, A_k, I_i^A e θ^A têm a mesma definição que no método de Broyden. Já o conjunto S_R é constituído pelas matrizes de ordem n que refletem a estrutura de esparsidade de $J(x)$, então, definimos:

$$I_i=\{j\in\{1,\ldots,n\} \ | \ \frac{\partial f_i}{\partial x_j}(x)=0, \ \forall x\in D\} \quad i=1,\ldots,n$$

daí:

$$\begin{aligned} I_i^R &= \{1,\ldots,n\}-I_i \\ S_R &= \{R\in I\!R^{n\times n} \ | \ r_{ij}=0 \ \text{ para todo } \ j\in I_i, \ i=1,\ldots,n\} \end{aligned}$$

Colocaremos a seguir, os principais teoremas, demonstrados em [32], que caracterizam a taxa superlinear para os métodos da família FQNSL.

A técnica para se chegar a este resultado, tem como etapas principais:
i) mostrar que a deterioração limitada é uma das propriedades das matrizes (A_k, R_k);

ii) obter a taxa linear de convergência; iii) verificar que as matrizes B_k satisfazem a condição de Dennis–Moré [9], o que conduz à taxa superlinear de convergência.

Sob as hipóteses: H1 à H4, e, considerando ainda:

i) $J(x)$ não singular, $\forall x \in D$

ii) $\mathcal{A}, \mathcal{R} : W \subset I\!R^{n\times n} \to I\!R^{n\times n}$ onde W é uma vizinhança aberta de J_*, tais que:

a) $\mathcal{A}(J_*)$, $\mathcal{R}(J_*)$ são não–singulares e, para todo $J \in W : J = \mathcal{A}(J)^{-1}\mathcal{R}(J)$
b) Existe $c > 0$, tal que:

$$\|\mathcal{A}(J) - \mathcal{A}(J_*)\| \leq c\|J - J_*\|$$

e

$$\|\mathcal{R}(J) - \mathcal{R}(J_*)\| \leq c\|J - J_*\|$$

para todo $J \in W$.

Daí, $\mathcal{A}$ e $\mathcal{R}$ são contínuas em J_* e, podemos supor que $\mathcal{A}$ e $\mathcal{R}$ são não singulares para todo $J \in W$.

c) Para todo $x, z \in D$, $\tilde{J} = \int_0^1 J(x + t(z - x))dt$

$$\tilde{J} \in W\ , \quad \mathcal{A}(\tilde{J}) \in S_A \quad \text{e} \quad \mathcal{R}(\tilde{J}) \in S_R$$

Para $\alpha \in (0, 1)$, define–se o produto escalar, no espaço $I\!R^{n\times n} \times I\!R^{n\times n}$:

$$\langle (A, R)\ ,\ (\overline{A}, \overline{R})\rangle_\alpha = \alpha\langle A, \overline{A}\rangle_F + (1 - \alpha)\langle R, \overline{R}\rangle_F$$

Considere: $S = S_A \times S_R$; e, para $\forall x \in D$ e $\forall z \in D : s = z - x$ e $y = F(z) - F(x)$
Defina: $V = V(x, z) = \{(A, R) \in S \mid Rs - Ay = 0\}$.

Com estas hipóteses e definições, podemos colocar os seguintes lemas e teoremas, demonstrados formalmente em [32].

Observando que para todo $x, z \in D$:
$y = F(z) - F(x) = [\int_0^1 J(x + t(z - x))dt]s = \tilde{J}s$

e, considerando a hipótese c, prova-se facilmente o lema abaixo, que estabelece que o conjunto solução (3.7.3) - (3.7.4) é não vazio:

Lema 3.7.1. Para todo $x, z \in D$, $V = V(x,z) \neq \phi$.

Desde que $V(x,z) \neq \phi$, podemos definir:
$(\widehat{A}, \widehat{R}) = \text{argmin } \{\|(A,R) - (A_*, R_*)\|_\alpha \mid (A,R) \in V\}$, onde $A_* = \mathcal{A}(J_*)$ e $R_* = \mathcal{R}(J_*)$.

$(\widehat{A}, \widehat{R})$ é a projeção ortogonal de (A_*, R_*) sobre V, relativa a norma α.

O lema 3.7.2 estabelece um limite superior para a α-distância entre (A_*, R_*) e V:

Lema 3.7.2. $\|(\widehat{A}, \widehat{R}) - (A_*, R_*)\|_\alpha \leq cL \max\{\|x - x^*\|\ ,\ \|z - x^*\}$

Esses dois lemas e as hipóteses iniciais permitem mostrar que as matrizes (A_k, R_k) têm a propriedade de deterioração limitada:

Lema 3.7.3. Supor que x^k e x^{k+1} são definidos por (3.7.1)-(3.7.4) e $x^k, x^{k+1} \in D$. Então:

$$\|(A_{k+1}, R_{k+1}) - (A_*, R_*)\|_\alpha \leq \|(A_k, R_k) - (A_*, R_*)\|_\alpha + 2cL \ \max\{\|x^k - x^*\|, \|x^{k+1} - x^*\|\}$$

O teorema abaixo estabelece a taxa linear de convergência para os métodos da família FQNSL.

Teorema 3.7.1. Existem vizinhanças $V(x^*, \varepsilon)$ e $V((A_*, R_*), \delta)$ tais que se $x^0 \in V(x^*, \varepsilon)$ e $(A_0, R_0) \in V((A_*, R_*), \delta)$ então, o algoritmo definido por (3.7.1)-(3.7.4) com as hipóteses introduzidas anteriormente, está bem definido e:

$$\|x^{k+1} - x^*\| \leq r\|x^k - x^*\| \quad \text{para todo} \quad k = 0, 1, \ldots$$

Assumindo as hipóteses do teorema 3.7.1, demonstra-se que os métodos da família FQNSL, possuem taxa superlinear de convergência. Este resultado é obtido

provando-se que as matrizes $\{B_k\}$ satisfazem a condição de Dennis–Moré. Uma ferramenta para se chegar a este resultado é:

Lema 3.7.4. i) $\lim \|(A_{k+1}, R_{k+1}) - (A_k, R_k)\|_\alpha = 0$ e, consequentemente,

ii) $\lim \|B_{k+1} - B_k\|_\alpha = 0$

Teorema 3.7.2. Sob as hipóteses do teorema 3.7.1, as matrizes B_k satisfazem

$$\lim_{k\to\infty} \frac{\|(B_k - J_*)s_k\|}{\|s_k\|} = 0 \quad \text{e, portanto}$$

x^k converge à x^* com taxa superlinear.

3.8. Convergência do Método de Dennis–Marwil

A principal dificuldade para se estabelecer a convergência local do método de Dennis Marwil, é que as matrizes U_k não obedecem o princípio de deterioração limitada.

Da secção 3.3, temos a relação:

$$\begin{aligned} & \|U_{k+1} - U_*\|_F \leq \|U_k - U_*\|_F + O(L_0 - L_*) \quad \Leftrightarrow \\ \Leftrightarrow \quad & \|U_{k+1} - U_*\|_F \leq \|U_0 - U_*\|_F + kO(L_0 - L_*) \end{aligned}$$

e, mesmo que $U_0 \in V(U_*, \delta)$ não há como provar que $U_k \in V(U_*, \delta)$, para todo k, pois o limite superior para a distância entre U_{k+1} e U_* aumenta a passos fixos.

Intuitivamente, este fato conduz à idéia de recomeços, isto é, a cada número fixo de iterações do algoritmo 2.5, a matriz deve ser a Jacobiana calculada em x^k, $J(x^k)$, para se garantir a permanência das matrizes U_k em $V(U_*, \delta)$.

A figura 3.8.1, mostra a necessidade de recomeços, considerando que a cada

iteração ocorre o pior caso:

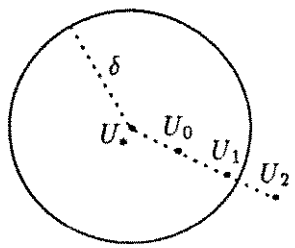

Figura 3.8.1

O teorema de convergência local do método de Dennis Marwil, tem como uma das hipóteses os recomeços com o Jacobiano a intervalos não maiores que um número fixo de iterações, o que implicará na taxa superlinear de convergência.

Teorema 3.8.1. Considerando: i) as hipóteses H1 à H4; ii) $\overline{K}$ um conjunto finito de índices, tais que, a diferença entre qualquer par de índices consecutivos é sempre menor ou igual a um inteiro fixo m; iii) a sequência $\{x^k\}$ gerada através do algoritmo 2.5, exceto que para $k \in \overline{K}, x^{k+1}$ é calculado pelo algoritmo 2.1.
Então, existe $\varepsilon > 0$, $\delta > 0$, tal que se $x^0 \in V(x^*, \varepsilon)$ e $B_0 \in V(J_*, \delta)$, x^k converge à x^* e,

$$\lim_{k\to\infty} \frac{\|x^{k+1} - x^*\|}{\|x^k - x^*\|}$$

Dem: [8] □

Na secção 3.7 descrevemos a família FQNSL e comentamos que o método de Dennis Marwil é o caso limite de uma sub família, denominada quase Dennis–Marwil.

Para se chegar a esta sub família, verificaremos inicialmente que o método de Johnson e Austria e o método de Chadee pertencem a família FQNSL.

No método de Johnson e Austria [27], $B_k = A_k^{-1} R_k$, onde R_k é triangular

superior e A_k é triangular inferior com diagonal unitária. A_{k+1} e R_{k+1} são matrizes triangulares como A_k e R_k respectivamente, e, são obtidas de modo a minimizar $\|M - M_k + U - U_k\|_F$.

Este método pertence à família FQNSL, se tomarmos:

$$\begin{aligned}
&\alpha = 1/2 \\
&I_i^A = \{1, \ldots, (n-1)\} \\
&\theta_{ij}^A = 0 \quad \text{se} \quad j > i \quad \text{e} \quad \theta_{ii}^A = 1 \quad i = 1, \ldots, n \\
&I_i^R = \{1, \ldots, n\} \\
&\theta_{ij}^R = 0 \quad \text{se} \quad j < i \quad \text{para} \quad i = 1, \ldots, n
\end{aligned}$$

O método de Chadee [5] é a adaptação do método de Johnson e Austria para o caso esparso, isto é, a estrutura de esparsidade de A_k e R_k devem ser preservadas.

Para $x \in D$, se $J(x) = \mathcal{L}(x)^{-1}\mathcal{U}(x)$, com $\mathcal{L}(x)$ triangular inferior com diagonal unitária e $\mathcal{U}(x)$ triangular superior.

$$\begin{aligned}
\text{Se } I_i^L &= \{j \in \{1, \ldots, n\} |\ \ell_{ij} = 0\} \quad i = 1, \ldots, n \\
I_i^U &= \{j \in \{1, \ldots, n\} |\ u_{ij} = 0\} \quad i = 1, \ldots, n,
\end{aligned}$$

verificamos que o método de Chadee pertence à família FQNSL se:

$$\begin{aligned}
\alpha &= 1/2 \\
I_i^A &= \{1, \ldots, (i-1)\} - I_i^L \quad i = 1, \ldots, n \\
\theta_{ij}^A &= 0 \text{ se } j > i \text{ e } j \in I_i^L \text{ e } \theta_{ii}^A = 1 \quad i = 1, \ldots, n \\
I_i^R &= \{1, \ldots, n\} - I_i^U \\
\theta_{ij}^R &= 0 \text{ se } j < i \text{ e } j \in I_i^R
\end{aligned}$$

Fixando os conjuntos I_i^A e I_i^R e as matrizes θ^A e θ^R como definidos acima teremos uma sub-família de métodos, gerada pela variação do parâmetro $\alpha \in (0, 1)$.

A partir das matrizes A_k e R_k, o cálculo das matrizes A_{k+1} e R_{k+1} envolve informações como x^k, x^{k+1}, $F(x^k)$, $F(x^{k+1})$ e α. Por esta razão, denotamos os elementos de A_{k+1} e R_{k+1} por $A_{k+1}(\alpha)$ e $R_{k+1}(\alpha)$.

O teorema enunciado abaixo e demonstrado em [32], estabelece que o método

de Dennis Marwil, é o caso limite desta sub-família, quando $\alpha \to 1$:

Teorema 3.8.2. Supor conhecidos x^k, x^{k+1}, $F(x^k)$, $F(x^{k+1})$, a matriz A_k, triangular inferior com diagonal unitária e a matriz R_k triangular superior. Sejam L_{k+1} e U_{k+1} as matrizes obtidas a partir destas informações através da fórmula de Dennis Marwil. Então:

$$\lim_{\alpha \to 1} A_{k+1}(\alpha) = L_{k+1}$$
$$\lim_{\alpha \to 1} R_{k+1}(\alpha) = U_{k+1}$$

Podemos então concluir que por ser o caso limite de uma família de métodos com taxa de convergência superlinear, o método de Dennis-Marwil deve incorporar algumas das boas propriedades destes métodos, e, este fato minimiza, do ponto de vista computacional, o efeito de não se conseguir um resultado de convergência local para o algoritmo puro.

3.9. Convergência dos Métodos com Escalamento da Fatoração

No capítulo 2, definimos esta família de métodos quase Newton e, descrevemos três de seus métodos.

O resultado obtido para os métodos da família FQNEF é a taxa linear de convergência para o algoritmo puro, e, a taxa superlinear quando se incorpora recomeços com o Jacobiano.

As suposições básicas para se estabelecer tais resultados são:
i) as hipóteses iniciais H1 à H5;
e, para cada método desta família, as condições sobre os fatores C, D, E são:
ii) existem as matrizes: C_*, D_*, E_*, não singulares, D_* diagonal, tais que: $J_* = C_* D_* E_*$ e
iii) existe $\delta_1 > 0$, tal que para qualquer matriz $B \in V(J_*, \delta_1)$, existem as matrizes: $C(B)$, $D(B)$, $E(B)$, tais que: $D(B)$ é diagonal, e, C, D, E, são funções contínuas de B. Sem perda de generalidade, supomos que B, $C(B)$, $D(B)$, $E(B)$ são não singulares se $B \in V(J_*, \delta_1)$.

No método de Atualização do Fator Diagonal, definimos C, D, E como sendo os fatores L, D, U; se B estiver suficientemente próxima de J_*, então B será não

singular, daí, a fatoração LDU de B existe e, as matrizes $C(B)$, $D(B)$, $E(B)$ são funções contínuas de B.

Para os outros dois métodos: Escalamento de Linhas e Escalamento de Colunas, temos:

$$C(B) = I\ ,\ E(B) = B$$
$$\text{e} \quad C(B) = B\ ,\ E(B) = I$$

respectivamente, que obviamente, são funções contínuas de B.

Enunciamos os lemas e teorema que conduzem à taxa linear de convergência. O primeiro lema estabelece um limitante superior para o erro nas aproximações d_i^k em relação à d_i^*. É interessante observar que este limitante depende apenas das informações da iteração zero, devido ao fato de ser única a matriz solução: D_{k+1}, para a equação: $(C_0 D_{k+1} E_0) s_k = y_k$.

O último teorema formaliza o resultado de convergência com taxa superlinear quando se incorpora recomeços com o Jacobiano verdadeiro ao algoritmo original.

Todos estes resultados encontram-se demonstrados em [28].

Considere a função $M(t)$, contínua e não decrescente, para $t \in (0, \delta_1)$, definida por:

$$M(t) = \max\{\|C(B) - C_*\|\ ,\ \|E(B) - E_*\|\ ,\ \|C^{-1}(B) - C_*^{-1}\|\ ,\ E^{-1}(B) - E_*\|,$$
$$\|D(B) - D_*\|,\ \text{para}\ \|B - J_*\| \leq t\}$$

Lema 3.9.1. Sob as hipóteses:
i) ε_2 e δ_2, tais que $\delta_2 \leq \delta_1$, e:

$$M(\delta_2)[1 + \|J_*\| + \max_i |d_i^*|] \ + \ L\varepsilon_2 \frac{\max\{C^{-1}(B),\ \text{tal que}\ B \in V(J_*, \delta_1)\}}{\alpha}$$
$$\leq \min_i \frac{|d_i^*|}{2}$$

ii) B_0, tal que $\|B_0 - J_*\| < t \leq \delta_2$

iii) $\{x^k\}$ está bem definida e $x^k \in V(x^*, u), u \leq \varepsilon_2$ para todo $k = 0, 1, \ldots$

Então:

a) $|d_i^k - d_i^*| \leq \max\{(M(t)[\|J_*\| + d_i^*] + L\|C_0^{-1}\|\frac{u}{\alpha}\ ,\ |d_i^0 - d_i^*|\} \quad i = 1, \ldots, n$

b) $|d_j^k| \geq \min_i \frac{|d_i^*|}{2}$ para todo $j = 1, \ldots, n$

Lema 3.9.2. Existe $c > 0$, tal que, se $B_0 \in V(J_*, \delta_2)$ e $x^0, x^1, \ldots, x^k \in V(x^*, \varepsilon_2)$, então:

$$\max_j\{\|(C_0 D_j E_0)^{-1}\|\ ,\ \|C_0 D_j\| + \|D_j E_*\|\ ,\quad j = 0, 1, \ldots, k\} \leq c$$

Teorema 3.9.1. Para $r \in (0, 1)$, existem $\varepsilon = \varepsilon(r)$ e $\delta = \delta(r)$, tais que, se $x^0 \in V(x^*, \varepsilon)$, $B_0 \in V(J_*, \delta)$ e $F(x^k) \neq 0$, para todo $k = 0, 1, \ldots$, então, as sequências $\{x^k\}$ e $\{D_k\}$ estão bem definidas e

$$\|x^{k+1} - x^*\| \ \leq\ r\ \|x^k - x^*\| \qquad \text{para todo} \quad k = 0, 1, \ldots$$

Teorema 3.9.2. Se, para $k \in \overline{K}$, $\overline{K} \subsetneq I\!N$, $B_k = J(x^k)$ então, existe $\varepsilon > 0$, tal que se $||x^0 - x^*|| < \varepsilon$, $\{x^k\}$ converge a x^* com taxa superlinear.

3.10. Convergência do Método de Atualização de uma Coluna por Iteração

As matrizes B_k geradas pelo método de atualização de uma coluna por iteração não apresentam a propriedade de deterioração limitada e por esta razão deve-se incluir recomeços com matrizes escolhidas em vizinhanças convenientes de $J(x^*)$ de modo a se obter a taxa linear de convergência.

Teorema 3.10.1. Seja $r \in (0, 1)$. Então, existem ε e δ tais que se $||x^0 - x^*|| < \varepsilon$ e $||B_k - J(x^*)|| < \delta$ sempre que $k \equiv 0(\text{mod. m})$, então as sequências $\{x^k\}$ e $\{B_k\}$ estão bem definidas e

$$||x^{k+1} - x^*|| \leq r||x^k - x^*|| \quad \text{para todo} \quad k = 0, 1, \ldots$$

Dem.: [30]

Teorema 3.10.2. Se, com as hipóteses do teorema 3.10.1 existe uma subsequência $\{B_{mk_j}\}$ tal que $\lim_{j\to\infty} B_{mk_j} = J(x^*)$ então a convergência é superlinear. Em particular isso acontece se $B_k = J(x^k)$ sempre que $k \equiv 0$ (mod. m)

Dem.: [30]

Os resultados de convergência local para ACI são semelhantes aos resultados obtidos para o método de Dennis-Marwil, no entanto, os resultados computacionais obtidos em alguns testes com problemas de pequeno e grande porte, mostram que o desempenho de ACI é comparável ao desempenho dos métodos de Broyden e Schubert que apresentam taxa superlinear.

É fácil verificar que ACI, quando aplicado a um sistema linear, converge em 2n iterações, usando os resultados de Gay [17]. Recentemente, Martínez conjecturou que, usando este resultado pode ser provada a convergência 2n–quadrática do método com recomeços do tipo $B_k \equiv B_0$ se $k \equiv 0$ (mod. 2n). Este resultado é válido para o método de Broyden, mas, não é válido para o método de Newton Modificado e para os métodos da família FQNEF.

Por outro lado, em um trabalho recente, [23], analisamos as propriedades de ACI no caso de dimensão infinita. O interesse direto desta análise é a aplicação do método à resolução de equações diferenciais por diferenças finitas, tomando discretizações cada vez mais finas. Além disto, estudos do comportamento do método de Broyden em espaços de dimensão infinita, [39], mostraram que a taxa superlinear só é obtida sob hipóteses muito restritivas, o que sugere que os recomeços precisam ser incorporados em aplicações de grande porte. O estudo completo da convergência de ACI em espaços de Hilbert encontra-se em [23].

CAPÍTULO 4

ESTRATÉGIA DE CONVERGÊNCIA GLOBAL

4.1. Introdução

A incorporação de uma estratégia de convergência global a métodos que apresentam resultados de convergência local, tem por objetivo reduzir a possibilidade de divergência do processo caso a aproximação inicial esteja distante da solução.

Estas estratégias são naturalmente incorporadas aos métodos para resolução de problemas de minimização tais como:

$$\min : \quad f : I\!R^n \to I\!R \tag{4.1.1}$$

Para o problema de resolução de sistemas não lineares:

$$F(x) = 0, \quad F : D \subset I\!R^n \to I\!R^n \tag{4.1.2}$$

podemos assumir a função de mérito:

$$f(x) = \frac{1}{2} F(x)^T F(x) \tag{4.1.3}$$

uma vez que qualquer solução para $F(x) = 0$ será ponto de mínimo para $\frac{1}{2} F(x)^T F(x)$.

A dificuldade é que os algoritmos globais garantem a convergência a pontos estacionários de $f(x)$, isto é, aqueles em que $\nabla f(x) = 0$; porém, $\nabla f(x) = J(x)^T F(x)$ e, portanto um ponto estacionário de $f(x)$ não é necessariamente uma solução para $F(x) = 0$.

Ainda assim, as estratégias de globalização para resolução de sistemas não lineares, são baseadas nas estratégias colocadas para o problema 4.1.1.

Optamos por incorporar aos métodos locais, uma estratégia de globalização tolerante no sentido que as iterações especiais para obter a convergência global só

serão efetuadas após um ciclo de iterações do método local, no caso de não se verificar um decréscimo satisfatório no valor de $f(x)$.

O objetivo dessa escolha é controlar o comportamento da sequência $\{x^k\}$ sem se colocar um controle rígido como um decréscimo satisfatório em $f(x)$ a cada iteração, principalmente porque uma característica numérica dos quase Newton é gerar uma sequência $\{x^k\}$ convergente à x^* sem que $||F(x^k)||$ decresça monoticamente.

A estratégia de globalização incorporada aos métodos locais consiste essencialmente em se efetuar buscas unidimensionais ao longo de direções de descida para $f(x)$.

Este processo de busca é detalhado no algoritmo 4.1.1.

Algoritmo 4.1.1.

Considere a função $f(x) = \frac{1}{2}F(x)^T F(x)$ e, denote $g(x) = \nabla f(x)$.

Dados: x^0 uma aproximação inicial e os parâmetros: $\alpha \in (0,\ 1)$ e $\theta,\ m > 0$, execute:

Passo 1: obtenha $s_k = -J(x^k)^{-1}F(x^k)$ executando os passos 1 a 5 do algoritmo 2.1.

Passo 2: se
i) $||s_k|| \geq m||g(x^k)||$ e,
ii) $\langle g(x^k),\ s_k\rangle \leq -\theta||g(x^k)||\ ||s_k||$, vá ao passo 3
caso contrário faça $s_k = -g(x^k)$

Passo 3: faça $\lambda = 1$
enquanto $f(x^k + \lambda s_k) > f(x^k) + \alpha\lambda\langle g(x^k),\ s_k\rangle$, execute:
Passo 3.1: obtenha $\overline{\lambda}$, tal que $\overline{\lambda} \in (0.1\lambda,\ 0.9\lambda)$
Passo 3.2: $\lambda \leftarrow \overline{\lambda}$

Passo 4: $\lambda_k = \lambda$

Passo 5: $x^{k+1} = x^k + \lambda_k s_k$
$k = k + 1$
volte ao passo 1.

Observações:
i) as direções geradas pelos métodos quase Newton não são necessariamente direções de descida para $f(x) = \frac{1}{2}F(x)^T F(x)$. Já para a direção de Newton temos:

$$\langle g(x^k),\ s_k \rangle = -(J(x^k)^T F(x^k))^T (J(x^k)^{-1} F(x^k)) = -F(x^k)^T F(x^k) < 0.$$

e, por este motivo esta é a direção escolhida no passo 1;
ii) As condições testadas no passo 2, são necessárias para se obter o resultado de convergência global; conforme veremos na seção 4.2, estas condições são satisfeitas pela direção de Newton, a menos de uma possível singularidade em $J(x^k)$;
iii) o parâmetro λ_k é obtido por um processo de interpolação quadrática-cúbica com salvaguardas, descrito em [11, pgs. 126 - 129]. Na seção 4.2, demonstramos a existência de tal parâmetro.

Considerando o problema 4.1.1 e a função 4.1.3, o algoritmo para os métodos locais com estratégia de globalização tolerante pode ser assim colocado:

Algoritmo 4.1.2.

Dados $x^0 \in D$, a aproximação inicial, o parâmetro $\delta \in (0,\ 1)$ e o número inteiro q; e, definindo: $y^j = \arg\min\{f(x^0), \ldots, f(x^j)\}$, execute:

Passo 1: obtenha x^1 através do método de Newton.

Passo 2: para $j = 1, \ldots, q$, execute:
Passo 2.1: obtenha x^{k+1} através de uma iteração do método local.
Passo 2.2: $k = k + 1$

Passo 3: enquanto $f(x^k) > \delta f(y^{k-1})$, execute os passos 3.1 a 3.3
Passo 3.1: $x^k = y^k$
Passo 3.2: obtenha x^{k+1} através de uma iteração do algoritmo 4.1.1.
Passo 3.3: $k = k + 1$

Passo 4: volte ao passo 2.

4.2. Teorema de Convergência Global

Lema 4.2.1. Assumindo que: i) $J(x^k)$ é não singular para todo $k = 0,\ 1,\ 2, \ldots$; ii) $\{x^k\}$ está contida num conjunto $\overline{D} \underset{\infty}{\subset} D$, compacto. Considerando $s_k = -J(x^k)^{-1}F(x^k)$, existe $m > 0$, tal que: $||s_k|| \geq m||g(x^k)||$ para todo $k = 0,\ 1,\ 2, \ldots$.

Dem.: para todo $k = 0,\ 1,\ 2, \ldots$.

$$(J(x^k)^T J(x^k))s_k = (J(x^k)^T J(x^k))(-J(x^k)^{-1}F(x^k)) = -g(x^k)$$

então:

$$||g(x^k)|| = ||J(x^k)^T J(x^k)s_k|| \leq ||J(x^k)^T J(x^k)||\ ||s_k||$$

$$\Rightarrow ||s_k|| \geq \frac{1}{||J(x^k)^T J(x^k)||}\ ||g(x^k)|| \qquad \forall\ k = 0,\ 1,\ 2, \ldots$$

Da suposição que $\overline{D}$ é compacto, segue que $||J(x)||$ e $||J(x)^T)||$ são uniformemente limitadas em $\overline{D}$, então, é possível obter $m > 0$ tal que:

$$||s_k|| \geq m||g(x^k)|| \quad \forall\ k = 0,\ 1,\ 2, \ldots \quad \square.$$

Lema 4.2.2. Assumindo que: i) $J(x^k)$ é não singular para todo $k = 0,\ 1,\ 2, \ldots$; ii) $\{x^k\}$ está contida num conjunto compacto $\overline{D} \underset{\infty}{\subset} D$ e, considerando $s_k = -J(x^k)^{-1}F(x^k)$, existe $\theta > 0$, tal que:

$$\langle g(x^k),\ s_k\rangle \leq -\theta||g(x^k)||\ ||s^k|| \quad \text{para todo} \quad k = 0,\ 1, \ldots$$

Dem.: denotando $J(x^k) = J_k$ e $F(x^k) = F_k$, temos:

$$\frac{\langle J_k^T F_k,\ (J_k^T J_k)^{-1}(J_k^T F_k)\rangle}{||J_k^T F_k||\ ||(J_k^T J_k)^{-1}(J_k^T F_k)||} = \frac{(J_k^T F_k)^T[(J_k^T J_k)^{-1}(J_k^T F_k)]}{||J_k^T F_k||\ ||(J_k^T J_k)^{-1}(J_k^T F_k)||} \geq$$

$$\frac{(J_k^T F_k)^T[(J_k^T J_k)^{-1}(J_k^T F_k)]}{||(J_k^T J_k)^{-1}||\ ||J_k^T F_k||^2} \tag{4.2.1}$$

seja $H_k = (J_k^T J_k)^{-1}$, e as matrizes:
$Q_k : n \times n$, ortogonal;
$D_k : n \times n$, diagonal com $d_i^k = \lambda_i$, λ_i é autovalor de H_k e , $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n$ tais que:

$$H_k = Q_k^T D_k Q_k$$

Definindo $w_k = J_k^T F_k$, teremos:

$$(J_k^T F_k)^T[(J_k^T J_k)^{-1}(J_k^T F_k)] = w_k^T H_k w_k = w_k^T (Q_k^T D_k Q_k) w_k =$$

$$= (Q_k w_k)^T D_k (Q_k w_k) = v_k^T D_k v_k, \quad \text{onde} \quad v_k = Q_k w_k$$

Agora,

$$v_k^T D_k v_k = \lambda_1 (v_1^k)^2 + \lambda_2 (v_2^k)^2 + \ldots + \lambda_n (v_n^k)^2 \geq \lambda_n \, ||v_k||^2 =$$

$$= \lambda_n ||w_k||^2 = \frac{1}{||H_k^{-1}||} ||w_k||^2 = \frac{1}{||J_k^T J_k||} \, ||J_k^T F_k||^2$$

Então:

$$(J_k^T F_k)^T[(J_k^T J_k)^{-1}(J_k^T F_k)] \geq \frac{1}{||J_k^T J_k||} \, ||J_k^T F_k||^2$$

Daí:

$$\frac{\langle J_k^T F_k, \; (J_k^T J_k)^{-1}(J_k^T F_k) \rangle}{||J_k^T F_k|| \; ||(J_k^T J_k)^{-1}(J_k^T F_k)||} \geq \frac{1}{\text{cond} \; (J_k^T J_k)} \Leftrightarrow$$

$$\frac{\langle g(x^k), \; s_k \rangle}{||g(x^k)|| \; ||s_k||} \leq \frac{-1}{\text{cond} \; (J_k^T J_k)} \Rightarrow$$

$$\langle g(x^k), \; s_k \rangle \leq \frac{-1}{\text{cond} \; (J_k^T J_k)} ||g(x^k)|| \; ||s_k|| \leq \frac{-1}{||J_k^T J_k|| \; ||(J_k^T J_k)^{-1}||} ||g(x^k)|| \; ||s_k||$$

Da suposição que $\overline{D}$ é compacto, segue que $||J(x)||$, $||J^T(x)||$, $||J^{-T}(x)||$ são uniformemente limitadas em $\overline{D}$, então, é possível obter $\theta > 0$, tal que

$$\langle g(x^k), \; s_k \rangle \leq -\theta ||g(x^k)|| \; ||s_k|| \qquad k = 0, \; 1, \; 2, \ldots \square.$$

Lema 4.2.3. Dados: o parâmetro $\alpha \in (0, \; 1)$; a aproximação x^k e a direção de descida s_k para $f(x)$ em x^k, existe $\widehat{\lambda} > 0$, tal que:

$$f(x^k + \widehat{\lambda} s_k) \leq f(x^k) + \alpha \widehat{\lambda} \langle g(x^k), \; s_k \rangle$$

Dem.: da definição de derivada direcional e, da hipótese que s_k é direção de descida para $f(x)$ em x^k, vem que:

$$\langle g(x^k), \; s_k \rangle = \lim_{\lambda \to 0} \frac{f(x^k + \lambda s_k) - f(x^k)}{\lambda} < 0$$

então

$$\lim_{\lambda \to 0} \frac{f(x^k + \lambda s_k) - f(x^k)}{\lambda \langle g(x^k),\ s_k \rangle} = 1$$

Portanto, dado $\alpha \in (0,\ 1)$, existe $\overline{\lambda} > 0$, suficientemente pequeno, tal que:

$$\frac{f(x^k + \widehat{\lambda} s_k) - f(x^k)}{\widehat{\lambda} \langle g(x^k),\ s_k \rangle} \geq \alpha, \qquad \forall \widehat{\lambda} \in (0,\ \overline{\lambda})$$

$$\Rightarrow \quad f(x^k + \widehat{\lambda} s_k) - f(x^k) \leq \alpha \widehat{\lambda} \langle g(x^k),\ s_k \rangle$$

$$\Rightarrow \quad f(x^k + \widehat{\lambda} s_k) \leq f(x^k) + \alpha \widehat{\lambda} \langle g(x^k),\ s_k \rangle, \qquad \forall \widehat{\lambda} \in (0,\ \overline{\lambda})$$

□.

O teorema 4.2.1 estabelece a convergência de uma sequência $\{x^k\}$ gerada pelo algoritmo 4.1.1 a um ponto estacionário de $f(x)$. O resultado é mais geral, no sentido que vale para toda função $f : \mathbb{R}^n \to \mathbb{R}$, $f \in C^1(\mathbb{R}^n)$ limitada inferiormente e, qualquer direção que satisfaça as condições do passo 2 do algoritmo 4.1.1.

Teorema 4.2.1. Considere a sequência $\{x^k\}$ gerada pelo algoritmo 4.1.1. Então,

$$\inf_k \ ||g(x^k)|| = 0$$

Dem.: para qualquer $k \in \mathbb{N}$, temos que:

$$f(x^k + \lambda_k s_k) \leq f(x^k) - \theta\alpha ||g(x^k)||\ ||x^{k+1} - x^k||$$

então,

$$\begin{aligned}
f(x^1) &\leq f(x^0) - \theta\alpha ||g(x^0)||\ ||x^1 - x^0|| \\
f(x^2) &\leq f(x^1) - \theta\alpha ||g(x^1)||\ ||x^2 - x^1|| \\
\vdots \qquad &\qquad \vdots \qquad\qquad \vdots \\
f(x^j) &\leq f(x^{j-1}) - \theta\alpha ||g(x^{j-1})||\ ||x^j - x^{j-1}||
\end{aligned}$$

e daí:

$$f(x^j) - f(x^0) \leq -\theta\alpha \sum_{k=0}^{j-1} ||g(x^k)||\ ||x^{k+1} - x^k|| \tag{4.2.2}$$

Definindo,

$$v_k = x^{k+1} - x^k = \lambda_k s_k \tag{4.2.3}$$

e considerando a condição i) do algoritmo 4.1.1:

$$||s_k|| \geq m||g(x^k)||$$

temos dois casos a analisar:

Caso 1: $\exists\ K1 \subsetneq_{\infty} I\!N$ tal que

$$||v_k|| \geq m||g(x^k)|| \quad \text{para todo} \quad k \in K1$$

Seja $\{x^{k_t}\}$ a subsequência de $\{x^k\}$ formada pelas aproximações x^k, $k \in K1$. De (4.2.2):

$$\begin{aligned} f(x^{k_t}) - f(x^0) &\leq -\theta\alpha \sum_{i=0}^{t-1} ||g(x^{k_i})||\ ||v_{k_i}|| \\ &\leq -\theta\alpha m \sum_{i=0}^{t-1} ||g(x^{k_i})||^2 \end{aligned}$$

e, tomando o limite para $t \to \infty$:

$$\lim_{t\to\infty}(f(x^{k_t}) - f(x^0)) \leq -\theta\alpha m \sum_{i=0}^{\infty} ||g(x^{k_i})||^2$$

como $f(x)$ é limitada inferiormente:

$$\begin{aligned} &\sum_{i=0}^{\infty} ||g(x^{k_i})||^2 \quad \text{converge, e, daí :} \\ &\lim_{i\to\infty} ||g(x^{k_i})|| = 0 \end{aligned} \tag{4.2.4}$$

Caso 2: $\exists \overline{K}$ tal que

$$||v_k|| < m||g(x^k)|| \quad \text{para todo} \quad k \geq \overline{K}$$

De (4.2.2), temos:

$$\begin{aligned} f(x^j) - f(x^0) &\leq -\theta\alpha \sum_{k=0}^{j-1} ||g(x^k)||\ ||v_k|| \\ &< -\frac{\theta\alpha}{m} \sum_{k=0}^{j-1} ||v_k||^2 \end{aligned}$$

Tomando o limite para $j \to \infty$:

$$\lim_{j\to\infty}(f(x^j) - f(x^0)) \leq -\frac{\theta\alpha}{m}\sum_{k=0}^{\infty} ||v_k||^2$$

como $f(x)$ é limitada inferiormente:

$$\sum_{k=0}^{\infty} ||v_k||^2 \quad \text{converge, e, daí :}$$
$$\lim_{k\to\infty} ||v_k|| = 0 \tag{4.2.5}$$

Então, $\lim_{k\to\infty}\sum_{j=k}^{\infty}||v_k|| = 0$. Deste resultado, e, da definição de v_k, segue que:
existem $k, j \geq 0$, tais que:

$$\begin{aligned} ||x^{k+j} - x^k|| &\leq ||x^{k+j} - x^{k+j-1}|| + \ldots + ||x^{k+2} - x^{k+1}|| + ||x^{k+1} - x^k|| \\ &\leq \sum_{i=k}^{\infty} ||x^{i+1} - x^i|| \to 0 \quad \text{quando} \quad k \to \infty \end{aligned}$$

e neste caso, $\{x^k\}$ é uma sequência de Cauchy e, portanto $\{x^k\}$ tem ponto limite x^*.

Agora, para todo $k \geq \overline{K}$ e da condição i) do passo 2 do algoritmo 4.1.1

$$||v_k|| = ||\lambda_k s_k|| < m||g(x^k)|| \Leftrightarrow \lambda_k < 1$$

e, considerando o passo 3 do algoritmo 4.1.1, existe $p_k = \overline{\lambda}_k s_k$, tal que:

$$||p_k|| \leq 10||x^{k+1} - x^k|| \tag{4.2.6}$$

$$\langle g(x^k), p_k\rangle \leq -\theta||g(x^k)||\ ||p_k|| \tag{4.2.7}$$

$$f(x^k + p_k) > f(x^k) + \alpha\langle g(x^k),\ p_k\rangle \tag{4.2.8}$$

De (4.2.5) e (4.2.6) segue que:

$$\lim_{k\to\infty} ||p_k|| = 0 \tag{4.2.9}$$

Considerando o conjunto: $\{\frac{p_k}{||p_k||},\ k \geq \overline{K}\}$
A sequência $\{\frac{p_k}{||p_k||}\}$ está contida na esfera unitária, e, portanto, possui uma subsequência convergente.

Existe então $K2 \subset \{k \geq \overline{K}\}$ tal que:

$$\lim_{k \in K2} \frac{p_k}{||p_k||} = d \quad \text{e} \quad ||d|| = 1 \tag{4.2.10}$$

De (4.2.7), segue que:

$$\langle g(x^k), \frac{p_k}{||p_k||} \rangle \leq -\theta ||g(x^k)||$$

e, tomando o limite para $k \in K2$:

$$\langle g(x^*), d \rangle \leq -\theta ||g(x^*)|| \tag{4.2.11}$$

De (4.2.8) temos:

$$f(x^k + p_k) - f(x^k) > \alpha \langle g(x^k), p_k \rangle$$

Do teorema do Valor Médio, existe $\xi_k \in (0,\ 1)$ tal que:

$$\langle g(x^k + \xi_k p_k),\ p_k \rangle > \alpha \langle g(x^k),\ p_k \rangle$$

aplicando o limite para $k \in K2$, e, de (4.2.9) e (4.2.10), segue que:

$$\langle g(x^*), d \rangle \geq \alpha \langle g(x^*),\ d \rangle$$

como $\alpha \in (0,\ 1)$, esta relação implica que:

$$\langle g(x^*), d \rangle \geq 0 \tag{4.2.12}$$

Portanto, (4.2.11) e (4.2.12) só serão satisfeitas simultaneamente se:

$$||g(x^*)|| = 0 \qquad \square.$$

Teorema 4.2.2. Seja $\{x^k\}$ a sequência gerada pelo algoritmo 4.1.2. Então,

i) $\inf_k ||F(x^k)|| = 0$ ou

ii) $\inf_k ||g(x^k)|| = 0$

Dem.:

caso 1) existe $K1 \subsetneq I\!N$ tal que:

$$f(x^k) \leq \delta f(y^{k-1}) \qquad \forall\, k \in K1 \tag{4.2.13}$$

Seja $\{x^{kj}\}$ a subsequência de $\{x^k\}$ formada por estas aproximações.
Então, de (4.2.13), e como $y^j = \arg\min\{f(x^0), \ldots f(x^j)\}$ temos a seguinte relação entre os elementos de $\{x^{kj}\}$:

$$f(x^{k_{j+1}}) \leq \delta f(x^{k_j})$$

e, portanto

$$f(x^{k_r}) \leq \delta^{(r-1)} f(x^{k_1})$$

Tomando o limite:

$$\lim_{r\to\infty} f(x^{k_r}) \leq f(x^{k_1}) \lim_{r\to\infty} \delta^{(r-1)} = 0 \quad \text{porque} \quad \delta \in (0,\ 1)$$

Daí, $\lim_{r\to\infty} ||F(x^{k_r})|| = 0$ e, portanto: $\inf_k ||F(x^k)|| = 0$

Caso 2: existe $K2$, tal que para qualquer $k \geq K2$, as aproximações x^k serão obtidas através do algoritmo 4.1.1, e, neste caso, a convergência a um ponto estacionário de $f(x)$ está assegurada pelo teorema 4.2.1.

□.

CAPÍTULO 5

FATORAÇÃO SIMBÓLICA PARA A FATORAÇÃO LU COM PIVOTEAMENTO PARCIAL

5.1. Introdução

No início deste trabalho ressaltamos a importância da estabilidade numérica na resolução do sistema linear $B_k s = -F(x^k)$ e, por esta razão, optamos pela fatoração LU com estratégia de pivoteamento parcial.

Denotaremos B_k por B e, representaremos o processo de fatoração por:

$$B = P_1 L_1 P_2 L_2 \ldots P_{n-1} L_{n-1} U \tag{5.1.1}$$

onde:
P_j: matriz de permutação, $n \times n$, que corresponde à troca de linhas na etapa j do processo;
L_j: matriz triangular inferior, $n \times n$, com diagonal unitária, na qual a coluna j contém os multiplicadores da etapa j;
U: matriz triangular superior, $n \times n$.

Sabe-se que a fatoração LU afeta a estrutura de esparsidade da matriz B uma vez que novos elementos não nulos são criados durante o processo de eliminação, provocando preenchimentos na estrutura original.

Por esta razão, a dificuldade em se implementar a fatoração LU no caso esparso está centrada na forma de se incorporar os preenchimentos à estrutura de dados que armazena a matriz B.

A estrutura de esparsidade das matrizes L_j e da matriz U depende da estrutura de B e da estrutura da linha pivotal em cada etapa. Por sua vez, a escolha da linha pivotal na etapa j depende dos valores numéricos dos elementos da coluna j e linhas $j,\ j+1, \ldots, n$, e, portanto, não há como se prever a sequência de linhas pivotais

no início do processo de fatoração e, consequentemente, não há como prever onde ocorrerão os preenchimenos, sem se efetuar os cálculos numéricos, conforme mostra o exemplo:

Exemplo 5.1.1. Considerando a estrutura de esparsidade de uma matriz B, 5×5, dada por:

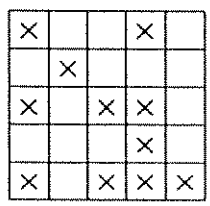

Figura 5.1.1

As figuras abaixo representam a estrutura resultante após a etapa 1, no caso em que a linha 1, 3 ou 5 for escolhida como linha pivotal.

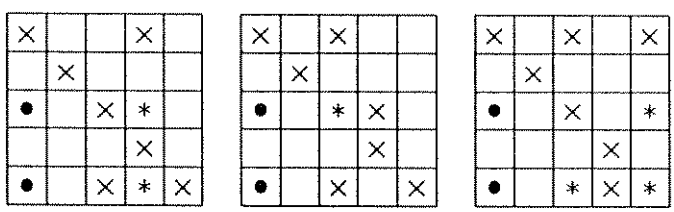

Figura 5.1.2

As posições marcadas com • correspondem aos multiplicadores e as posições marcadas com $*$ representam os preenchimentos. Cada um dos casos leva a uma matriz com estrutura de esparsidade e valores numéricos diferentes e, somente no final da etapa 1 é possível estabelecer qual será a linha pivotal para a etapa 2.

O fato de não se poder prever a sequência de linhas pivotais é um dos motivos que leva a maioria das implementações da fatoração LU com pivoteamento parcial a adotar uma estrutura de dados dinâmica [12, 14, 37, 44].

Neste tipo de estrutura os novos elementos não nulos são alocados durante a

fase de eliminação. A desvantagem é que as operações para alocar ou acessar um elemento são caras no sentido que exigem: buscas, comparações, atualizações de apontadores.

Diante da impossibilidade de se fixar a priori uma estrutura de dados exata para acomodar as estruturas dos fatores L_j, $j = 1, \ldots n-1$, e de U, o processo denominado Fatoração Simbólica tem por objetivo gerar, a partir da estrutura original de B, uma estrutura de dados grande o suficiente para armazenar as estruturas de L_j, $j = 1, \ldots n-1$, e de U, para qualquer seqüência possível de matrizes $P_1, \ldots, P_{n-1}$.

A vantagem deste processo é que realizada a Fatoração Simbólica, a partir da estrutura de esparsidade de B, a fatoração numérica de B pode ser efetuada usando uma estrutura de dados estática.

Em 1985, George e Ng, [20], mostraram que a estrutura dos fatores de Choleski de B^TB contêm a estrutura dos fatores L_j, $j = 1, \ldots n-1$, e de U, e, este resultado independe da sequência de linhas pivotais.

A desvantagem deste algoritmo é que B^TB pode não ser esparsa ainda que B o seja, e, além disso, a esparsidade do fator de Choleski depende fortemente de um bom esquema de ordenação das colunas de B.

As experiências computacionais com este algoritmo mostram que a estrutura de dados resultante sobreestima largamente a memória necessária para os fatores L_j, $j = 1, \ldots n-1$, e U em qualquer sequência pivotal.

Em 1987, George e Ng, [21], apresentaram um algoritmo para a Fatoração Simbólica que gera uma estrutura de dados capaz de armazenar os fatores L_j, $j = 1, \ldots n-1$, e U porém, com uma sobreestimativa de memória menor que o esquema baseado na fatoração de Choleski.

Neste trabalho optamos pela estrutura de dados estática, considerando nesta opção que para o caso da resolução de sistemas não lineares, em que a fatoração LU deve ser efetuada a cada iteração como nos métodos de Newton e Schubert, ou pelo menos na iteração inicial como nos métodos quase Newton, o custo computacional por se realizar a fatoração simbólica inicialmente, pode ser compensado durante a fase numérica da fatoração.

5.2. Fatoração Simbólica

Na descrição que segue, estamos supondo que os elementos da diagonal da matriz B são não nulos; podemos assumir esta hipótese, uma vez que se a matriz B é não singular é possível permutar suas linhas ou colunas de modo que os elementos da diagonal de B sejam não nulos [13].

O algoritmo proposto por George e Ng [21], tem como argumentos básicos:
a) em cada etapa k do processo de eliminação, as linhas que podem ter sua estrutura alterada são as linhas candidatas a linha pivotal e a estrutura de cada uma delas só será alterada a partir da coluna k;
b) sob a suposição que não ocorrem cancelamentos estruturais ou numéricos a nova estrutura de cada uma destas linhas deverá estar contida na união das estruturas de todas as linhas candidatas a pivotal.

Exemplo 5.2.1. Considerando o exemplo 5.1.1 e a figura 5.1.2, temos que uma estrutura capaz de acomodar quaisquer preenchimentos que possam ocorrer durante a etapa 1, é dada por:

×		×	×	×
	×			
×		×	×	×
			×	
×		×	×	×

Figura 5.2.1.

Discutiremos a validade dos argumentos a) e b), considerando o caso geral $n \times n$.

Particionando a matriz B original em:

$$B = \begin{pmatrix} \widehat{\alpha} & \widehat{u}_1^T \\ \widehat{v}_1 & \widehat{G}_1 \end{pmatrix}$$

ond $\widehat{u}_1 : (n-1) \times 1$, $\widehat{v}_1 : (n-1) \times 1 : \widehat{G}_1 : (n-1) \times (n-1)$ e $\widehat{\alpha} \in I\!R$;

e, realizando a etapa 1 do processo de eliminação, com pivoteamento parcial, teremos:

$$\begin{aligned} P_1 B &= P_1 \begin{pmatrix} \hat{\alpha} & \hat{u}_1^T \\ \hat{v}_1 & \hat{G}_1 \end{pmatrix} = \begin{pmatrix} \alpha & u_1^T \\ v_1 & G_1 \end{pmatrix} \\ &= \begin{pmatrix} 1 & 0 \\ v_1/\alpha & I \end{pmatrix} \begin{pmatrix} \alpha & u_1^T \\ 0 & G_1 - \frac{1}{\alpha}(v_1 u_1^T) \end{pmatrix} \\ &= \begin{pmatrix} 1 & 0 \\ l_1 & I \end{pmatrix} \begin{pmatrix} \alpha & u_1^T \\ 0 & \overline{B}_1 \end{pmatrix} \end{aligned}$$

Ao final desta etapa somente a primeira linha de B e as linhas j para as quais $\hat{v}_{j1} \neq 0$ poderão ter suas estruturas alteradas.

Sob a suposição que não ocorrem cancelamentos, a união das estruturas das linhas candidatas a pivotal no início desta etapa, é dada pela estrutura do vetor $\overline{u}_1$:

$$\overline{u}_1^T = \hat{u}_1^T + \hat{v}_1^T \, \hat{G}_1$$

Estamos interessados em obter uma estrutura capaz de acomodar todos os preenchimentos ao final da etapa 1, sem importar neste momento os valores numéricos dos elementos. Uma vez que este preenchimento depende da linha pivotal, podemos escrever a matriz $\overline{B}$:

$$\overline{B} = \begin{pmatrix} \hat{\alpha} & \overline{u}_1^T \\ \hat{v}_1 & \hat{G}_1 \end{pmatrix}$$

que é a matriz B com o vetor $\hat{u}_1^T$ substituído por $\overline{u}_1^T$.

Efetuando a eliminação (sem pivoteamento) teremos:

$$\overline{B} = \begin{pmatrix} \hat{\alpha} & \overline{u}_1^T \\ \hat{v}_1 & \hat{G}_1 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ \hat{v}_1/\hat{\alpha} & I \end{pmatrix} \begin{pmatrix} \hat{\alpha} & \overline{u}_1^T \\ 0 & \hat{G}_1 - \frac{1}{\hat{\alpha}}(\hat{v}_1 \overline{u}_1^T) \end{pmatrix}$$

Denotando:

$$\overline{l}_1 \equiv \frac{\hat{v}_1}{\hat{\alpha}} \quad \text{e} \quad \overline{G}_1 \equiv \hat{G}_1 - \frac{1}{\hat{\alpha}} (\hat{v}_1 \overline{u}_1^T)$$

teremos:

$$\overline{B} = \begin{pmatrix} 1 & 0 \\ \overline{l}_1 & I \end{pmatrix} \begin{pmatrix} \hat{\alpha} & \overline{u}_1^T \\ 0 & \overline{G}_1 \end{pmatrix}$$

onde: as estruturas de $\overline{l}_1$, $\overline{u}_1$ e $\overline{G}_1$ são limitantes superiores para as estruturas de l_1, u_1 e $\overline{B}_1$ qualquer que seja a matriz de permutação P_1. Isto é: se $Nel\,(Z)$ representa o número de elementos não nulos no vetor ou matriz Z, então:
$Nel\,(l_1) = Nel\,(\overline{l}_1)$
$Nel\,(u_1^T) \subseteq Nel(\overline{u}_1^T)$
$Nel(\overline{B}_1) \subseteq Nel(\overline{G}_1)$.

O mesmo raciocínio pode ser repetido para a etapa 2, uma vez que esta etapa é equivalente à etapa 1 aplicada sobre a matriz $\overline{G}_1$, e, sob a suposição que não ocorrem cancelamentos numéricos ou estruturais, $\overline{G}_1$ tem elementos não nulos na diagonal.

Este argumento pode ser aplicado recursivamente, isto é, na etapa k, $k = 1, \ldots, (n-1)$ realizamos o processo de eliminação descrito acima sobre uma matriz de ordem $(n-k+1)$, e obtém-se ao final de cada etapa o vetor coluna $\overline{l}_k$ e o vetor linha $\overline{u}_k$, ambos de dimensão $(n-k)$. Ao final do processo o conjunto de vetores $\overline{l}_j$ e $\overline{u}_j$, $j = 1, \ldots, (n-1)$ são tais que: a estrutura de $\overline{l}_j$ será usada como limitante superior para a estrutura do fator L_j, $j = 1, \ldots, n-1$ e a estrutura de $\overline{u}_j$ será usada como limitante e superior para a estrutura da linha j do fator U, $j = 1, \ldots, (n-1)$.

O bom desempenho do processo da fatoração simbólica está no fato que em cada etapa k, $k = 1, \ldots, (n-1)$ é preciso atualizar apenas as estruturas do vetor linha $\overline{u}_k$ e do vetor coluna $\overline{l}_k$. Com relação às etapas posteriores à etapa k, a informação essencial é: "entre as etapas j, $j = (k+1), \ldots, (n-1)$ qual será a primeira a ser afetada pelo resultado da etapa k?"

Exemplo 5.2.2. Considerando a estrutura de uma matriz 8×8 dada por:

×		×					
	×	×					
	×	×		×			
			×				
				×			
×				×	×		
		×			×	×	
×							×

Figura 5.2.2

Após a etapa 1, o vetor $\overline{u}_1$ guarda a união das estruturas das candidatas a pivotal

e $\bar{l}_1$ os índices das linhas candidatas a pivotal.

Teremos então a matriz estrutural:

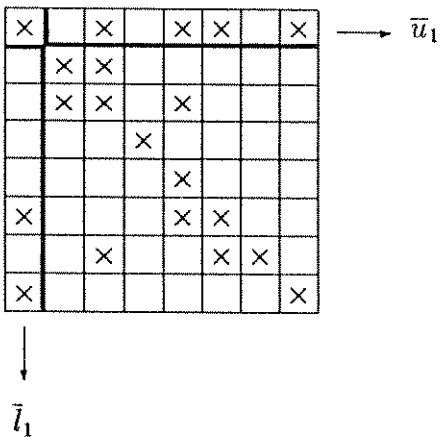

Figura 5.2.3

Combinando as informações de $\bar{l}_1$ e $\bar{u}_1$ é possível concluir que a etapa 3 será a primeira que será influenciada pela etapa 1. Isto porque as linhas 6 e 8 são candidatas a pivotais na etapa 1 (informação obtida através de $\bar{l}_1$) e, na estrutura atualizada destas linhas o primeiro elemento não nulo ocorre na coluna 3 (informação obtida através de $\bar{u}_1$). Portanto, as linhas 6 e 8 serão candidatas a pivotais na etapa 3 e a estrutura de $\bar{u}_1$ estará contida na estrutura de $\bar{u}_3$.

Definindo os conjuntos de índices:

$$\tau_k = \{j |\ \bar{u}_{kj} \neq 0\}$$

$$\sigma_k = \{i |\ \bar{l}_{ik} \neq 0\}$$

temos que $\sigma_k \subset \tau_k$ desde que os elementos da diagonal são não nulos.

Se $\sigma_k \neq \phi$ existem candidatas a pivotal além da linha k e a primeira etapa que será influenciada pela etapa k será a etapa ρ_k, onde:

$$\rho_k = \min_t \{t \in \tau_k\}.$$

No exemplo 5.2.2, $\rho_1 = 3$.

Como processar a etapa ρ_k?

Sabe-se que a estrutura final do vetor $\overline{u}_{\rho_k}$ depende da estrutura das linhas candidatas a pivotal nesta etapa. Mas, se a coluna ρ_k não foi atualizada como obter o índice das linhas que se "tornaram candidatas devido aos preenchimentos que ocorreram nas etapas anteriores?"

Para se ter esta informação constrói-se no desenvolvimento do processo, o vetor de índices S_{ρ_k} definido por:

$$S_{\rho_k} = \{r |\ \sigma_r \neq \phi \quad \text{e} \quad \rho_k = \min_t\{t \in \tau_r\}\}$$

isto é, S_{ρ_k} é o conjunto dos índices das etapas r, $1 \leq r < \rho_k$, tais que a etapa ρ_k é a primeira a ser influenciada pelos resultados da etapa r.

A etapa ρ_k será processada por fases:
na fase i) é feito um levantamento das linhas candidatas a pivotais na estrutura original; considera-se somente os preenchimentos que ocorrem nestas linhas, a partir da coluna ρ_k; estes preenchimentos são incorporados à linha ρ_k;
na fase ii) utiliza-se o vetor S_{ρ_k}; a cada $r \in S_{\rho_k}$, incorpora-se: a estrutura de $\overline{l}_r$ à $\overline{l}_{\rho_k}$ e a estrutura de $\overline{u}_r$ à estrutura da linha ρ_k.

Aplicando este processo ao exemplo 5.2.2 teremos:
os resultados da etapa 1, conforme a figura 5.2.3, estão armazenados nos vetores $\overline{l}_1$ e $\overline{u}_1$; $\rho_1 = 3$ e $S_3 = \{1\}$;
na etapa 2 são candidatas a pivotal as linhas 2 e 3 e $S_2 = \phi$, então, após esta etapa o vetor $\overline{u}_2$ guarda a união das estruturas das linhas 2 e 3 e o vetor $\overline{l}_2$ o índice $i = 3$; $\rho_2 = 3$ e $S_3 = \{1, 2\}$;
na etapa 3, inicialmente pesquisamos a coluna 3 (na figura 5.2.3) que mostra as linhas 3 e 7 como candidatas a pivotal. Analisando o vetor S_3, temos que as etapas 1 e 2 influenciam a etapa 3. O vetor $\overline{l}_3$ é obtido através da união das estruturas da coluna 3 original, e das estruturas dos vetores $\overline{l}_1$ e $\overline{l}_2$; e o vetor $\overline{u}_3$ é obtido pela união das estruturas das linhas 3 e 7 e dos vetores $\overline{u}_1$ e $\overline{u}_2$.

A figura 5.4.3 apresenta a estrutura resultante após as etapas 1, 2 e 3:

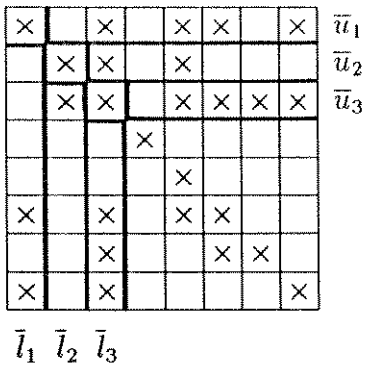

Figura 5.2.4

As informações dos vetores S_j, $j = 1, \ldots, n-1$ podem ser armazenadas num único vetor S, de dimensão n, isto porque:
i) a cada etapa k está associado um único índice, ρ_k, e, então:

$$S_i \cap S_j = \phi \quad \text{se} \quad i \neq j, \;\; i, \; j = 1, \ldots, n-1$$

ii) na etapa k, $\rho_k \geq k+1$.
então, ao final da etapa k, obtido ρ_k, o vetor S pode ser atualizado do seguinte modo:

$$S(k) = S(\rho_k)$$
$$S(\rho_k) = k$$

Inicialmente $S(i) = 0, \;\; i = 1, \ldots, n$.

Para o exemplo 5.2.2:

ao final da etapa 1: $\rho_1 = 3$ e : $S(1) = 0$
$S(3) = 1$
ao final da etapa 2: $\rho_2 = 3$ e : $S(2) = 1$
$S(3) = 2$

Ao se processar a fase 2 da etapa 3, obtém-se os índices das etapas que a influenciam, através do vetor S, partindo da posição 3:

$$S(3) = 2$$
$$S(2) = 1$$
$$S(1) = 0$$

e portanto, as etapas 1 e 2 influenciaram a etapa 3.

Algoritmo.

Definindo os vetores:

$C_j = \{i |\ b_{ij} \neq 0\} \quad j = 1, \ldots, n$

$R_i = \{j |\ b_{ij} \neq 0\} \quad i = 1, \ldots, n$

M: vetor $n \times 1$ onde:

$M(i) = 1$ se a linha i já foi candidata a pivotal em alguma etapa

$= 0$ caso contrário;

execute:

Passo 1: (inicialização dos vetores M e S)

para $i = 1, \ldots, n$, faça:

$M(i) = 0$

$S(i) = 0$

Passo 2: para $k = 1, \ldots, n-1$ execute os passos 2.1 a 2.5

passo 2.1: $\overline{L}_k = \phi$ e $\overline{U}_k = \phi$

passo 2.2: para $i \in C_k$, faça:

se $M(i) = 0$ faça :

$\overline{L}_k \leftarrow \overline{L}_k \cup \{i\}$

$\overline{U}_k \leftarrow \overline{U}_k \cup R_i - \{1,\ 2, \ldots, k-1\}$

$M(i) = 1$

passo 2.3: $r = S(k)$

$S(k) = 0$

passo 2.4: enquanto $r \neq 0$, faça:

$\overline{L}_k \leftarrow \overline{L}_k \cup \overline{L}_r - \{1,\ 2, \ldots, k-1\}$

$\overline{U}_k \leftarrow \overline{U}_k \cup \overline{U}_r - \{1,\ 2, \ldots, k-1\}$

$\overline{r} = r$

$r \leftarrow S(r)$

$S(\overline{r}) = 0$

passo 2.5: se $\{\overline{L}_k - \{k\}\} \neq \phi$ então
obtenha $\rho = \min\{t|\ t \in \overline{U}_k - \{k\}\}$
$S(k) \leftarrow S(\rho)$
$S(\rho) \leftarrow k$

Analisando os passos 2.2 e 2.4 do algoritmo, concluímos que:
a) os elementos da estrutura original, guardados em R_i e C_i, $1 \leq i \leq n$, são considerados em uma única etapa do processo de fatoração simbólica, pois uma vez que uma linha i, $1 \leq i \leq n-1$, tenha sido candidata a pivotal em alguma etapa j, $j \leq i$, sua estrutura original estará contida nos vetores $\overline{L}_j$ e $\overline{U}_j$;
b) os elementos dos vetores $\overline{L}_j$ e $\overline{U}_j$, $1 \leq j \leq n-1$, (correspondentes a uma etapa j, já processada) serão considerados no máximo uma vez em alguma etapa futura, mais precisamente na etapa ρ_j.

De a) e b) podemos afirmar que os elementos dos vetores R_i e C_i, $1 \leq i \leq n-1$, serão "acessados" uma única vez e os elementos dos vetores $\overline{L}_j$ e $\overline{U}_j$, $1 \leq j \leq n-1$, serão "acessados" no máximo uma vez, de modo que a complexidade do algoritmo pode ser estimada por:

$$O(\max(|B|, \sum_{k=1}^{n-1} |\overline{L}_k| + |\overline{U}_k|))$$

onde $|B|, |\overline{L}_k|$ e $|\overline{U}_k|$ denotam o número de elementos em B, $\overline{L}_k$ e $\overline{U}_k$ respectivamente.

CAPÍTULO 6

IMPLEMENTAÇÃO COMPUTACIONAL

Com o objetivo de comparar o desempenho computacional dos métodos descritos no capítulo 2, desenvolvemos o pacote Rouxinol, escrito em Fortran 77 e implementado no VAX 11/785 - Sistema Operacional VMS, da Unicamp.

6.1. Características de Rouxinol

a) Métodos implementados

N: método de Newton (algoritmo 2.1)
NM: método de Newton Modificado (algoritmo 2.2)
B: método de Broyden (algoritmo 2.3)
S: método de Schubert (algoritmo 2.4)
DM: método de Dennis Marwil (algoritmo 2.5)
ED: método de Atualização do Fator Diagonal (algoritmo 2.6.1)
EC: método de Escalamento de Colunas (algoritmo 2.6.2)
EL: método de Escalamento de Linhas (algoritmo 2.6.3)
ACI: método de Atualização de uma Coluna por Iteração (algoritmo 2.7)

b) Formas de Execução

b.1. Método de Newton

Em Rouxinol, o método de Newton pode ser executado conforme o algoritmo 2.1 (algoritmo local) ou com estratégia de convergência global (algoritmo 4.1.2)

b.2. Métodos quase Newton

A forma básica de execução de cada método quase Newton é a que corresponde à implementação dos algoritmos 2.2 a 2.7 que é a forma denominada "sem restarts" uma vez que somente a iteração inicial é uma iteração Newton.

Todos os métodos quase Newton podem ser executados com estratégia de convergência global, algoritmo 4.1.2.

Localmente, o desempenho dos quase Newton pode ser melhorado se iterações Newton forem intercaladas sob algum critério, entre as iterações quase Newton, que é a forma de execução "com restarts".

O critério mais simples para o restart é aquele em que uma iteração Newton é efetuada após um número fixo de iterações quase Newton:
se $k \equiv 0 \pmod{q}$, x^{k+1} é obtido por uma iteração Newton, onde q é um número inteiro fornecido pelo usuário.

O índice de eficiência de Ostrowski, [38], fornece um valor ótimo para q em função da dimensão do problema; porém, este valor ótimo é estabelecido levando-se em conta apenas o esforço na avaliação das funções. O esforço computacional na resolução dos sistemas lineares deve também ser considerado, bem como a atualização das matrizes B_k, daí, um bom índice de eficiência de uma iteração deve relacionar o tempo de execução com a taxa de decréscimo no valor de $||F(x)||$:
se t_k denota o tempo de execução na iteração k e considerando o fator r_k dado por:

$$r_k = ||F(x^{k+1})||/||F(x^k)|| \tag{6.1.1}$$

se, $r_k < 1$ e, se a relação (6.1.1) se mantém nas iterações seguintes, podemos estimar que o tempo de execução necessário para se obter a solução, será proporcional a:

$$-t_k/\log r_k$$

e, sob estas considerações, a eficiência de uma iteração pode ser definida por:

$$\begin{aligned} E_k &= (-\log\ r_k)/t_k \quad \text{se} \quad r_k < 1 \\ &= 0 \qquad\qquad\quad \text{caso contrário} \end{aligned}$$

O algoritmo abaixo corresponde à execução de um método quase Newton usando o critério da eficiência para decidir o restart:

Algoritmo 6.1.1.

Passo 0: flag = 1

Passo 1: $k = k + 1$
se flag = 1 execute o passo 2 caso contrário, execute o passo 3.

Passo 2: passo 2.1: obtenha x^k através de uma iteração do algoritmo 2.1
passo 2.2: se $||F(x^k)|| < ||F(x^{k-1})||$ faça:
$r_k = ||F(x^k)||/||F(x^{k-1})||$
$EN = -\log r_k/t_k$
flag = 0
passo 2.3: volte ao passo 1.

Passo 3: passo 3.1: obtenha x^k através de uma iteração do algoritmo quase Newton escolhido.
passo 3.2: se $||F(x^k)|| \geq ||F(x^{k-1})||$ faça flag = 1, volte ao passo 1
passo 3.3: calcule
$r_k = ||F(x^k)||/||F(x^{k-1})||$
$EQ = -\log r_k/t_k$
passo 3.4: se $EQ < EN$ faça flag = 1
passo 3.5: volte ao passo 1.

c) Execução de um teste

A resolução de um sistema não linear, em Rouxinol, é separada em duas fases:
fase Simbólica: nesta fase é executado o algoritmo da fatoração simbólica e a estrutura de dados para a fatoração LU é fixada;
fase Numérica: nesta fase, o sistema não linear é resolvido pelo método e forma de execução escolhidos; no caso de comparação entre os vários métodos na resolução de um mesmo sistema não linear, poderão ser realizados vários testes consecutivos, aproveitando a estrutura de dados fixada na fase simbólica.

c) Singularidade

Os algoritmos descritos no capítulo 2 consideram a possibilidade da matriz B_k ser singular e incluem um passo para se detectar a singularidade através de um teste relativo.

Em Rouxinol o usuário pode optar se a execução deve ou não prosseguir em caso de singularidade; com relação ao parâmetro de tolerância, Tolsing, o usuário pode fornecer este valor ou pode optar que seja igual a: $epsmaq$ ou $(epsmaq)^{1/2}$ onde $epsmaq$ é a precisão da máquina; neste caso, uma rotina interna em Rouxinol é executada para se obter $epsmaq$.

d) Critério de Parada

Um critério de parada natural para algoritmos para sistemas não lineares é impor que x^k seja aceito como solução se

$$||F(x^k)||_\infty < \varepsilon_1 \tag{6.1.2}$$

onde ε_1 é um parâmetro real, próximo de zero, fornecido pelo usuário. Neste trabalho, declaramos convergência do tipo 0, quando ocorre (6.1.2).

O teste (6.1.2) pode ser difícil ou impossível de ser satisfeito em casos onde a matriz Jacobiana é grande em x^*; por esta razão, incluímos outro teste de parada, que consiste em se aceitar x^{k+1} como solução se:

$$||s_k||_\infty = ||x^{k+1} - x^k||_\infty < \varepsilon_2||x^{k+1}||_\infty + 10^{-25} \tag{6.1.3}$$

e, neste caso, declaramos a convergência do tipo 1.

Dado que os métodos implementados apresentam resultados de convergência local, incorporamos um teste para detectar divergência; esta situação ocorre se for gerada uma aproximação x^k para a qual se verifica:

$$||F(x^k)|| > Fmax\ ||F(x^0)|| \tag{6.1.4}$$

onde $Fmax$ é um número positivo, grande, fornecido pelo usuário.

Desde que testes relativos são mais aconselháveis que testes absolutos, os critérios (6.1.2) e (6.1.4) podem ser trocados respectivamente por:

$$||F(x^k)||_\infty < \varepsilon_1||F(x^0)||_\infty \quad \text{e} \tag{6.1.5}$$

$$||F(x^k)||_\infty > Fmax||F(x^0)||_\infty \tag{6.1.6}$$

Deixamos como opção para o usuário a escolha entre testes absolutos (6.1.2 e 6.1.4) e testes relativos (6.1.5 e 6.1.6).

A execução do programa é interrompida se for excedido o número máximo de iterações ou o tempo máximo de execução sendo que estes limitantes são definidos pelo usuário.

Observamos que todos estes critérios são adotados na implementação dos algoritmos globais, exceto para a convergência tipo 0 (teste com o valor de $F(x)$). Devido ao fato de usarmos $f(x) = \frac{1}{2}F(x)^T F(x)$ como função de mérito, usamos o teste:

$$\frac{||F(x)||_2}{(n)^{1/2}} < \varepsilon_1 \tag{6.1.7}$$

para detectar a convergência do tipo 0.

6.2. Análise do Desempenho Computacional

Para o estudo do desempenho computacional dos algoritmos locais e globais, escolhemos os seguintes problemas:

Problema 1 (Broyden Tridiagonal) [2, 3]

$$\begin{aligned} f_1(x) &= (3 - 2x_1)x_1 - 2x_2 + 1 \\ f_i(x) &= (3 - 2x_i)x_i - x_{i-1} - 2x_{i+1} + 1 \quad i = 2, \ldots, n-1 \\ f_n(x) &= (3 - 2x_n)x_n - x_{n-1} + 1 \end{aligned}$$

Problema 2 (Broyden Banda) [3].

$$f_i(x) = (3 + 5x_i^2)x_i + 1 - \sum_{j \in I_i}(x_j + x_j^2) \quad i = 1, \ldots, n$$

onde:

$$\begin{aligned} I_i &= \{i_1, \ldots, i_2\} - \{i\} \\ i_1 &= max\ \{1,\ i-5\} \quad i_2 = min\{n,\ i+5\} \end{aligned}$$

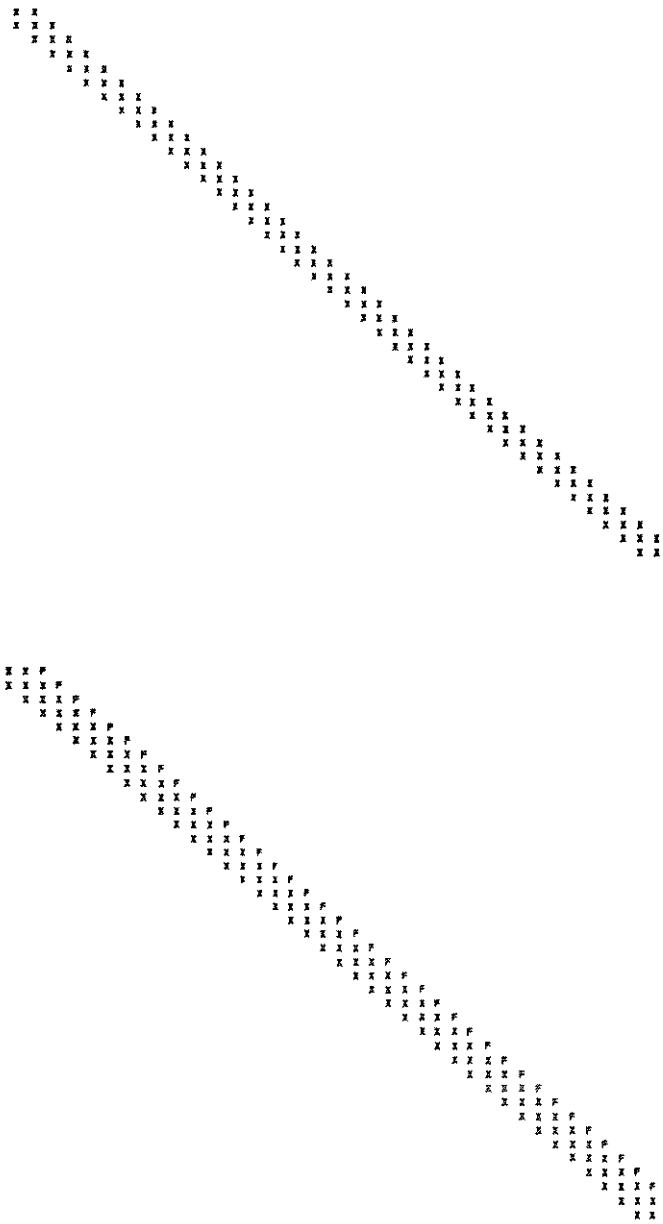

Figura 6.2.1. Estrutura da matriz Jacobiana e estrutura de dados para a fatoração LU(problema 1, $n = 40$)

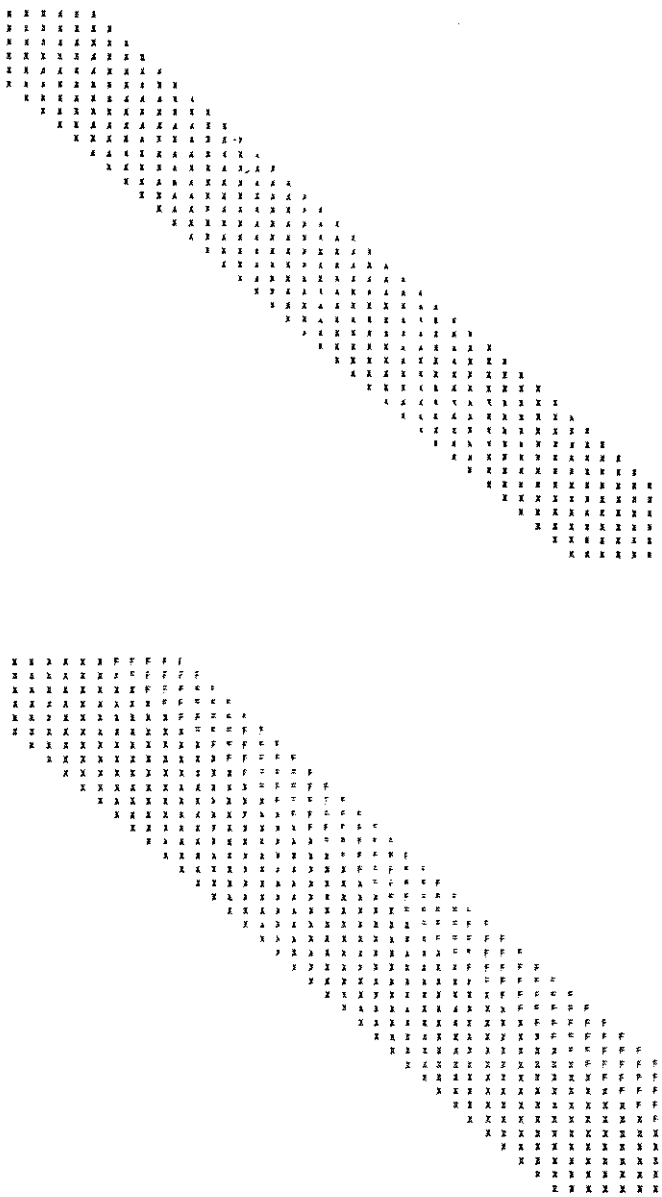

Figura 6.2.2. Estrutura da matriz Jacobiana e estrutura de dados para a fatoração LU(problema 2, $n = 40$)

Problema 3. (Trigexp - Toint) [42].

$$\begin{aligned} f_1(x) &= 3x_1^3 + 2x_2 - 5 + sen(x_1 - x_2)\, sen(x_1 + x_2) \\ f_i(x) &= -x_{i-1}e^{(x_{i-1}-x_i)} + x_i(4 + 3x_i^2) + 2x_{i+1} + \\ &\quad + sen(x_i - x_{i+1})\, sen(x_i + x_{i+1}) - 8 \quad i = 2, \ldots, n-1 \\ f_n(x) &= -x_{n-1}e^{(x_{n-1}-x_n)} + 4x_n - 3 \end{aligned}$$

A matriz Jacobiana deste sistema é tridiagonal como para o problema 1.

Problema 4. (Problema de Poisson) [41]

Este problema é o sistema de equações não lineares que surge da discretização por diferenças finitas do problema de contorno de Poisson:

$$\begin{aligned} &\Delta u = \frac{u^3}{1 + s^2 + t^2} \quad 0 \leq s \leq 1, \quad 0 \leq t \leq 1 \\ &u(0,\ t) = 1 \\ &u(1,\ t) = 2 - e^t \quad t \in [0,\ 1] \\ &u(s,\ 0) = 1 \\ &u(s,\ 1) = 2 - e^s \quad s \in [0,\ 1] \end{aligned}$$

Efetuamos testes usando uma malha de L^2 com $L = 15$ e $L = 31$, que resultaram em problemas de dimensão 225 e 961 respectivamente.

Problema 5.

$$\begin{aligned} f_1(x) &= -2x_1^2 + 3x_1 - 2x_2 + 0.5x_{\alpha_1} + 1.0 \\ f_i(x) &= -2x_i^2 + 3x_i - x_{i-1} - 2x_{i+1} + 0.5x_{\alpha_i} + 1.0 \quad i = 2, \ldots, n-1 \\ f_n(x) &= -2x_n^2 + 3x_n - x_{n-1} + 0.5x_{\alpha_n} + 1.0 \end{aligned}$$

para α_i, $i = 1, \ldots, n$, escolhido aleatóriamente nos intervalos: $\alpha_i \in \{\alpha_{i\,\min},\ \alpha_{i\,\max}\}$ onde $\alpha_{i\,\min} = \max\{1,\ i-b\}$ e $\alpha_{imax} = \min\{n,\ i+b\}$ para um parâmetro b que define a largura da banda.

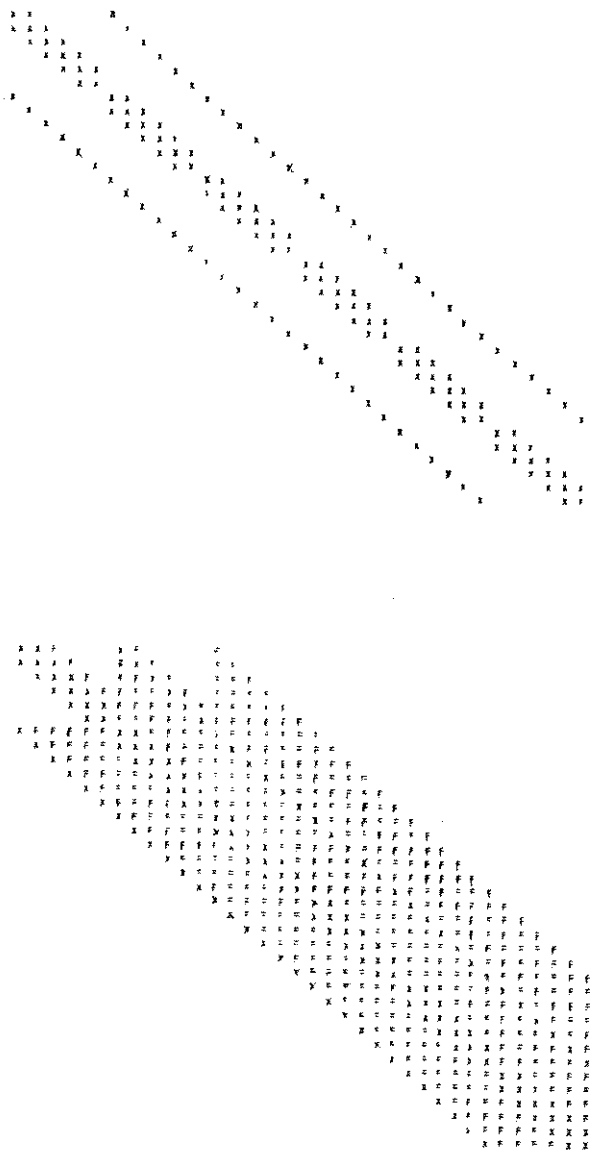

Figura 6.2.3. Estrutura da matriz Jacobiana e estrutura de dados para a fatoração LU(problema 4, $n = 36$)

Figura 6.2.4. Estrutura da matriz Jacobiana e estrutura de dados para a fatoração LU(problema 5, $n = 40$, $b = 15$)

Problema 6 (Broyden Faixa)

$$
\begin{aligned}
f_1(x) &= -2x_1^2 + 3x_1 + 3x_{n-4} - x_{n-3} - x_{n-2} + 0.5x_{n-1} - x_n + 1 \\
f_i(x) &= -2x_i^2 + 3x_i - x_{i-1} - 2x_{i+1} + 3x_{n-4} - x_{n-3} - x_{n-2} + \\
&\quad +0.5x_{n-1} - x_n + 1 \qquad i = 2, \ldots, n-1 \\
f_n(x) &= -2x_n^2 + 3x_n - x_{n-1} + 3x_{n-4} - x_{n-3} - x_{n-2} + 0.5x_{n-1} - x_n + 1
\end{aligned}
$$

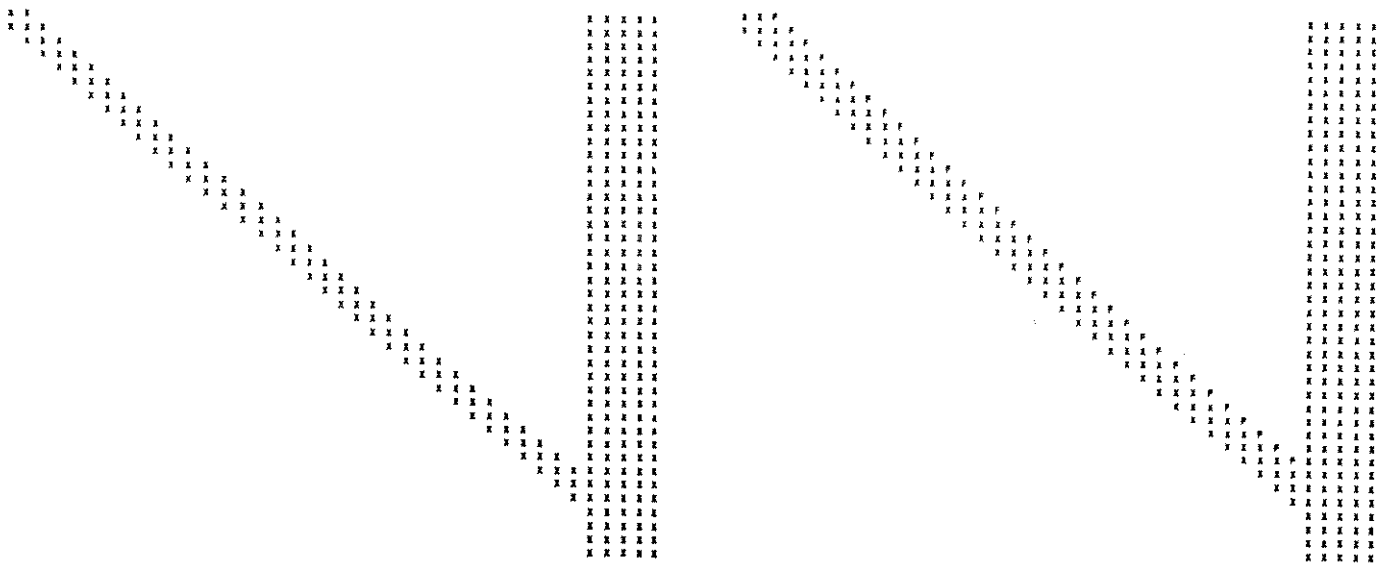

Figura 6.2.5: Estrutura da matriz Jacobiana e estrutura de dados para a fatoração LU(problema 6, $n = 40$)

Problema 7 (Broyden Singular)

$$
\begin{aligned}
f_1(x) &= ((3 - 2x_1)x_1 - 2x_2 + 1)^2 \\
f_i(x) &= ((3 - 2x_i)x_i - x_{i-1} - 2x_{i-1} + 1)^2 \quad i = 2, \ldots, n-1 \\
f_n(x) &= ((3 - 2x_n)x_n - x_{n-1} + 1)^2
\end{aligned}
$$

Este problema é equivalente ao problema 1, mas, neste caso a matriz Jacobiana é singular na solução; nosso objetivo, neste teste, é observar o comportamento dos diferentes algoritmos na ocorrência de Jacobianos singulares.

Cada algoritmo local foi executado sem restart e com restart baseado no critério da eficiência.

Para podermos avaliar o desempenho dos algoritmos globais, usamos pontos não-clássicos como chutes iniciais, para os quais o método de Newton não convergiu ou teve dificuldades para convergir.

Os chutes iniciais e parâmetro β usado para efetuar o controle sobre o tamanho do passo foram os seguintes:

i) testes efetuados com os algoritmos locais:
para os problemas 1, 2, 5, 6 e 7:

$$x^0 = (-1, \ldots, -1)^T \quad \text{e} \quad \beta = 10$$

para o problema 3, realizamos testes com dois chutes iniciais:

$$x^0 = (0, \ldots, 0)^T \quad \text{e} \quad \beta = 10 \quad \text{e}$$
$$x^0 = (0.3, \ldots, 0.3)^T \quad \text{e} \quad \beta = 10$$

para problema 4, usamos:

$$x^0 = (-1, \ldots, -1)^T \quad \text{e} \quad \beta = 5$$

ii) testes efetuados com os algoritmos globais:
problema 1: $x^0 = (10^{-3}, \ldots, 10^{-3})^T$ e $\beta = 5\ 000$;
problema 2: $x^0 = (0, \ldots, 0)^T$ e $\beta = 10$;
problema 3: $x^0 = (-1, \ldots, -1)^T$ e $\beta = 10$;
problema 4: $x^0 = (-3, \ldots, -3)^T$ e $\beta = 100$;
problema 5: $x^0 = (-0.008, \ldots, -0.008)^T$ e $\beta = 10^4$;
problema 6: $x^0 = (0.8, \ldots, 0.8)^T$ e $\beta = 10$;
problema 7: $x^0 = (5, \ldots, 5)^T$ e $\beta = 20$;

Os demais parâmetros e tolerâncias usados, foram os seguintes:

$\alpha = 10^{-4}$;
$\varepsilon_1 = \varepsilon_2 = 10^{-4}$ para os testes com os algoritmos locais e globais;
$Fmax = 10^{10}$ para os algoritmos locais e
$Fmax = 10^{30}$ para os algoritmos globais;
$Tolsing = (epsmaq)^{1/2}$ para os algoritmos locais e
$Tolsing = 10^{-7}$ para os algoritmos globais;
$\delta = 0.9$, usado apenas nos algoritmos globais.

As tabelas (6.2.1 - 6.2.5) resumem o desempenho da fase simbólica e da fase numérica nos vários testes efetuados.

Na tabela 6.2.1 apresentamos o tempo de execução na fatoração simbólica. Observamos que para os problemas 1, 2, 3, 6, 7 este tempo é proporcional à dimensão do problema. Para estes problemas o tempo é representado em milisegundos. Para os problemas 4 e 5 o tempo é dado em segundos.

Problema	Tempo de Execução
1	(0.33n) ms
2	(1.22n) ms
3	(0.33n) ms
4 (n = 225)	1.31 s
4 (n = 961)	16.63 s
5 (n = 1000 b = 100)	12.37 s
6	(0.69n) ms
7	(0.33n) ms

Tabela 6.2.1. Tempo de Execução da Fatoração Simbólica

A tabela 6.2.2 apresenta o tempo de execução de uma iteração típica de cada método implementado em cada teste. Da mesma forma que na fase simbólica, observamos uma proporcionalidade entre o tempo e a dimensão do problema nos testes 1, 2, 3, 6 e 7. No problema 4, os tempos de execução de uma iteração Newton e de uma iteração de Schubert são "quase" proporcionais à $L^2n(= L^4)$, enquanto que para os outros métodos são proporcionais à $L^3(= Ln)$.

O tempo é dado em milisegundos nos problemas 1, 2, 3, 6 e em segundos nos problemas 4 e 5. Os resultados para o problema 7 são os mesmos que para o problema 1.

Os tempos de uma iteração típica dos métodos de Broyden e Atualização de uma Coluna por Iteração dependem da iteração k, considerando o caso em que só a primeira iteração é uma iteração Newton. Isto se deve à implementação com memória limitada que resulta num trabalho computacional maior a cada iteração

efetuada. Basicamente o número de operações em uma iteração de Broyden é da ordem de $(k+1)(2n)$ e para ACI este número é da ordem de $(k+1)n$, se contar a resolução dos dois sistemas triangulares.

PROB / MET.	1	2	3	4 n = 225	4 n = 961	5	6
N	0.26n	1.05n	0.42n	0.66	10.6	10.5	0.51n
NM	0.10n	0.26n	0.18n	0.09	0.75	0.8	0.13n
S	0.32n	1.13n	0.43n	0.69	10.6	10.1	0.66n
DM	0.18n	0.43n	0.26n	0.16	1.43	1.39	0.32n
ED	0.12n	0.29n	0.19n	0.11	0.68	0.82	0.16n
EC	0.13n	0.28n	0.19n	0.10	0.65	0.75	0.15n
EL	0.13n	0.29n	0.19n	0.09	0.60	0.75	0.16n
B	(0.089 + 0.015k)n	(0.243 + 0.015k)n	(0.186 + 0.008k)n	0.11 (k = 1)	0.76 (k = 1)	0.78 (k = 1)	(0.137 + 0.014k)n
ACI	(0.091 + 0.005k)n	(0.25 + 0.004k)n	0.15 + 0.006k)n	0.09 (k = 1)	0.71 (k = 1)	0.78 (k = 1)	(0.145 + 0.005k)n

Tabela 6.2.2. Tempo de execução de uma iteração típica de cada método em cada teste.

Os resultados numéricos dos algoritmos locais são apresentados na tabela 6.2.3. O desempenho computacional de cada algoritmo é representado por: $(CP,\ k,\ k_1,\ k_2,\ t)$ que indica que a execução do algoritmo terminou com critério de parada de código CP, após um total de k iterações, sendo k_1 iterações Newton e k_2 iterações quase Newton e usando t segundos de tempo de execução (CPU). O código CP pode assumir 5 valores:
$CP = 0$ para convergência do tipo 0;
$CP = 1$ para convergência do tipo 1;

$CP = 2$ para divergência ;
$CP = 3$ se for excedido o número máximo de iterações e
$CP = 4$ se for excedido o tempo máximo de execução.

A tabela 6.2.4 apresenta o desempenho computacional dos algoritmos globais. Para cada método, o desempenho do algoritmo local sem restart é representado de acordo com a convenção da tabela 6.2.3.

Os algoritmos globais foram executados com $q = 3$ e o desempenho de cada método é representado por: (CP, $k1$, $k2$, NAF, MF, t) onde:
CP: (pode assumir os valores definidos para a tabela 6.2.3);
$k1$: número total de iterações;
$k2$: número de iterações especiais;
NAF: número de avaliações da função;
MF: menor valor de $||F(x^k)||_2/(n)^{1/2}$ obtido no processo;
t: tempo de execução (em segundos).

A tabela 6.2.5 apresenta um levantamento entre:
o total de elementos não nulos na estrutura original do problema;
a previsão de preenchimento para os fatores L e U após a execução do algoritmo da fatoração simbólica e,
o número de elementos não nulos em cada fator L e U, na iteração em que houve o maior preenchimento na fatoração LU durante a execução do método de Newton.

Na coluna referente ao problema resolvido, a letra L corresponde à execução do algoritmo 2.1, e a letra G corresponde à execução do método de Newton com estratégia de convergência global, algoritmo 4.1.2.

PROB.	n	RESTART	N	NM	S	DM	ED	EC	EL	B	ACI
1	5000	Não	0,3,3,0 3.97	1,9,1,8 5.25	0,6,1,5 9.44	0,5,1,4 4.94	1,5,1,4 3.75	1,5,1,4 3.85	0,6,1,5 4.47	0,6,1,5 4.97	0,6,1,5 4.38
1	5000	Sim	–	1,9,1,8 5.33	0,4,2,2 6.63	0,5,1,4 5.13	1,5,1,4 3.88	1,5,1,4 3.82	0,6,2,4 5.02	0,6,2,4 5.41	0,6,1,5 4.36
2	5000	Não	0,4,4,0 21.1	1,17,1,16 25.9	0,9,1,8 50.3	1,11,1,10 26.6	0,6,1,5 12.5	0,6,1,5 12.4	0,6,1,5 12.5	0,9,1,8 18.8	1,8,1,7 15.3
2	5000	Sim	–	1,17,1,16 25.8	0,5,3,2 28.8	0,8,2,6 23.0	0,6,1,5 12.2	0,6,1,5 12.1	0,6,1,5 12.3	0,9,1,8 17.9	1,8,1,7 15.0
3 $x^0 = 0.0$	5000	Não	0,8,8,0 16.7	3,100,1,99 97.8	overflow	2,46,1,45 57.4	3,100,1,99 101.	3,100,1,99 102.	2,13,1,12 14.8	2,11,1,10 15.6	2,50,1,49 91.2
3 $x^0 = 0.0$	5000	Sim	–	0,15,5,10 19.7	0,10,5,5 20.6	0,11,4,7 17.2	0,12,3,9 15.1	0,12,3,9 15.4	1,18,2,16 20.4	0,13,3,10 17.8	0,11,4,7 14.9
3 $x^0 = 0.3$	5000	Não	0,6,6,0 12.5	3,100,1,99 91.3	0,11,1,10 23.4	0,12,1,11 16.5	1,19,1,18 19.1	1,13,1,12 13.5	1,36,1,35 36.2	2,6,1,5 7.77	1,21,1,20 26.2
3 $x^0 = 0.3$	5000	Sim	–	1,11,3,8 13.0	0,7,4,3 14.5	1,8,2,6 11.5	0,9,3,6 12.3	0,9,3, 6 12.2	0,8,3,5 11.5	0,8,3,5 11.5	1,10,2,8 12.6
4	225	Não	0,3,3,0 1.98	0,5,1,4 1.03	0,4,1,3 2.73	0,5,1,4 1.29	1,7,1,6 1.29	1,6,1,5 1.16	1,6,1,5 1.12	1,4,1,3 0.92	0,5,1,4 0.98
4	225	Sim	–	0,5,1,4 1.01	0,4,2,2 2.67	0,5,1,4 1.37	0,5,2,3 1.55	0,5,2,3 1.59	1,5,2,3 1.59	1,4,1,3 1.0	0,5,1,4 1.05
4	961	Não	1,4,4,0 42.3	1,5,1,4 13.6	1,5,1,4 53.0	1,5,1,4 16.3	1,8,1,7 15.4	1,6,1,5 13.8	1,5,1,4 13.0	1,4,1,3 12.2	1,5,1,4 12.8
4	961	Sim	–	1,5,1,4 12.8	1,4,2,2 39.8	1,5,1,4 15.6	1,4,2,2 22.0	1,6,2,4 22.6	1,5,2,3 22.6	1,4,1,3 12.1	1,5,1,4 12.9
5 $b = 100$	1000	Não	0,4,4,0 42.5	1,11,1,10 18.2	0,6,1,5 62.1	0,7,1,6 19.0	1,6,1,5 14.8	1,6,1,5 14.2	0,6,1,5 14.9	0,7,1,6 15.2	0,7,1,6 15.2
5 $b = 100$	1000	Sim	–	1,11,1,10 18.7	0,4,2,2 41.6	0,7,1,6 18.4	1,6,1,5 14.6	1,6,1,5 14.8	0,6,1,5 14.2	0,7,1,6 15.4	0,7,1,6 15.1
6	5000	Não	0,4,4,0 10.2	0,14,1,13 11.3	0,7,1,6 22.5	0,8,1,7 13.7	0,10,1,9 9.87	1,8,1,7 7.88	1,7,1,6 7.45	0,8,1,7 9.4	0,8,1,7 8.26
6	5000	Sim	–	0,14,1,13 11.2	0,5,3,2 14.9	0,6,2,4 11.4	0,6,2,4 9.1	0,6,2,4 8.25	0,6,2,4 8.63	0,8,1,7 9.57	0,8,1,7 8.24
7	5000	Não	0,9,9,0 12.9	3,100,1,99 56.1	3,100,1,99 148.	overflow	0,12,1,11 8.76	0,15,1,14 11.2	0,15,1,14 11.4	1,34,1,33 56.3	1,33,1,32 36.5
7	5000	Sim	–	0,16,6,10 13.5	0,11,6,5 17.8	0,11,6,5 13.3	0,12,1,11 8.29	0,12,2,10 9.74	0,12,2,10 9.17	0,13,2,11 11.1	0,13,2,11 10.6

Tabela 6.2.3. Resultados Numéricos dos Algoritmos Locais

PROB.	n	VERSÃO	N	NM	S	DM	ED	EC	EL	B	ACI
1	1000	Local	0,17,17,0 4.49	3,100,1,99 10.9	3,100,1,99 36.9	3,100,1,99 18.0	3,100,1,99 12.2	3,100,1,99 12.7	3,100,1,99 12.4	3,100,1,99 57.9	3,100,1,99 36.9
		Global	0,13,2,25 .6E-6,3.77	0,24,5,37 .5E-4,4.02	0,28,7,41 .2E-6,10.1	0,19,3,32 .1E-4,4.12	overflow	3,100,55,275 .6E-1,27.2	0,31,7,44 .2E-6,5.6	0,28,7,41 .2E-6,7.0	0,27,6,40 .1E-4,5.77
2	1000	Local	0,83,83,0 81.1	3,100,1,99 24.6	3,100,1,99 116.	overflow	3,100,1,99 26.1	3,100,1,99 26.5	3,100,1,99 26.4	3,100,1,99 70.2	3,100,1,99 51.1
		Global	0,19,9,42 .2E-4,21.5	0,23,1,23 .9E-4,7.21	0,13,2,13 .9E-4,13.7	overflow	overflow	0,36,15,59 .2E-4,24.0	0,36,15,59 .2E-4,24.2	0,36,15,59 .3E-5,25.6	0,35,8,41 .9E-4,18.6
3	1000	Local	0,14,14,0 5.14	3,100,1,99 16.5	0,12,1,11 4.08	0,12,1,11 2.92	1,29,1,28 5.81	1,29,1,28 5.43	1,20,1,19 3.78	1,12,1,11 2.92	1,14,1,13 3.21
		Global	0,9,1,10 .6E-6,3.7	0,8,1,8 .2E-4,1.95	0,10,0,10 .3E-4,3.62	0,10,0,10 .3E-4,2.61	0,18,0,18 .7E-4,3.52	0,17,3,17 .9E-4.4.34	0,18,0,18 .7E-4,3.56	0,11,0,11 .2E-4,2.92	0,16,4,17 .5E-5,4.85
4	225	Local	3,100,100,0 71.9	3,100,1,99 10.8	3,100,1,99 76.7	overflow	3,100,1,99 11.4	3,100,1,99 11.6	overflow	3,100,1,99 21.4	1,19,1,18 2.52
		Global	0,7,1,8 .3E-4,4.98	1,20,3,21 .3E-2,4.39	0,16,4,17 .4E-4,11.3	overflow	overflow	1,19,3,20 .7E-2,4.34	1,27,5,28 .1E-2,6.31	1,18,2,18 .2E-2,3.95	1,20,3,20 .3E-1,4.26
5	100	Local	3,100,100,0 47.9	3,100,1,99 5.19	3,100,1,99 47.1	3,100,1,99 8.65	3,100,1,99 5.24	3,100,1,99 5.29	3,100,1,99 5.33	3,100,1,99 9.73	3,100,1,99 9.22
		Global	0,12,2,14 .3E-5,5.91	0,21,4,23 .5E-4,3.54	0,16,1,17 .2E-4,7.9	0,20,4,22 .3E-4,3.97	0,24,6,26 .3E-5,4.56	0,26,5,28 .5E-4,4.18	0,24,6,26 .4E-6,4.57	0,29,5,31 .2E-4,4.61	0,24,6,26 .2E-5,4.39
6	100	Local	3,100,100,0 4.22	3,100,1,99 1.33	3,100,1,99 5.66	3,100,1,99 2.49	3,100,1,99 1.55	3,100,1,99 1.49	3,100,1,99 1.52	3,100,1,99 2.05	3,100,1,99 1.42
		Global	1,20,8,74 0.2,1.12	1,20,8,74 0.2,0.72	1,20,8,74 0.2,1.23	1,20,8,74 0.2,0.93	1,20,8,74 0.2,0.77	1,20,8,74 0.2.0.76	1,20,8,74 0.2,0.78	1,21,9,78 0.2,0.85	1,20,8,74 0.2,0.8
7	100	Local	3,100,100,0 2.97	3,100,1,99 1.08	3,100,1,99 3.83	3,100,1,99 1.88	3,100,1,99 1.27	3,100,1,99 1.4	3,100,1,99 1.25	3,100,1,99 5.74	3,100,1,99 5.44
		Global	1,72,51,307 0.4,3.92	1,90,48,299 .5E-1,4.15	1,42,21,121 .9E-1,2.56	1,21,3,35 0.2,0.82	overflow	0,59,11,87 .6E-4,2.1	0,51,7,67 .9E-4,1.86	1,18,2,23 0.1,0.69	1,23,5,45 0.1,0.95

Tabela 6.2.4. Resultados Numéricos dos Algoritmos Globais

PROB.	DIMENSÃO	ELEM. NÃO NULOS	FATORAÇÃO SIMBÓLICA Nº DE ELEM. NÃO NULOS			% de UTILIZAÇÃO (FAT. LU x FAT. SIMB.)		
			L	U	TOTAL	L	U	TOTAL
1(L)	5000	14998	4999	14997	19996	100 %	66.7 %	75 %
1(G)	1000	2998	999	2997	3996	100 %	66.7 %	75 %
2(L)	5000	54970	24985	54945	79930	100 %	54.6 %	68.8 %
2(G)	1000	10970	4985	10945	15930	100 %	57.3 %	70.6 %
3(L)	5000	14998	4999	14997	19996	100 %	66.7 %	75.0 %
3(G)	1000	2998	999	2997	3996	100 %	66.7 %	75.1 %
4(L)	225	1065	3164	6341	9505	100 %	53.4 %	68.9 %
4(G)	225	1065	3164	6341	9505	100 %	56.5 %	71 %
4(L)	961	4681	28860	57749	86609	100 %	51.6 %	67.8 %
5(L)	1000 (b=100)	3998	25328	59356	84684	100 %	37.1 %	55.9 %
5(G)	100 (b=100)	398	1730	3498	5228	99.1 %	44.5 %	62.5 %
6(L)	5000	39984	5005	39972	44977	100 %	87.5 %	88.9 %
6(G)	1000	784	105	772	877	100 %	92.2 %	93.2 %
7(L)	5000	14998	4999	14997	19996	100 %	66.7 %	75 %
7(G)	100	298	99	297	396	100 %	82.2 %	87 %

Tabela 6.2.5. Porcentagem de utilização da estrutura de dados para a fatoração LU

6.3. Comentários e Conclusões

Considerando os resultados dos algoritmos locais apresentados na tabela 6.2.3, concluímos:

a) o método de Newton consegue convergência em todos os testes e com menor número de iterações que qualquer quase Newton ($q-N$), exceto para o problema 4, em que Newton e Broyden efetuaram 4 iterações. O fato de Newton ser o único a não apresentar falhas, não é surpreendente, pois a região de convergência da versão local de Newton é maior que as regiões dos $q-N$;

b) a situação de não-convergência ocorre nos quase Newton somente na opção sem restart e, este fato mostra que a principal propriedade do restart pela eficiência é corrigir a situação de uma trajetória errada dos $q-N$; esta situação é melhor observada no problema 3 com $x^0=(0,\ldots,0)^T$;

c) com relação ao tempo de execução, em todos os testes o menor tempo coube sempre a um quase Newton, exceto para o problema 3 com $x^0=(0,\ldots,0)^T$;

d)o método de Newton Modificado não foi superior a qualquer outro $q-N$; já que os demais $q-N$ satisfazem a equação secante, podemos dizer que NM não foi superior a nenhum método secante. Este resultado mostra a importância da condição secante sobre a matriz B_k, uma vez que teoricamente, ED, EC, EL possuem o mesmo resultado de convergência que NM, conforme colocado no capítulo 3;

e) verificamos que o bom desempenho dos algoritmos DM, ED, EC e EL está associado com as situações em que a condição secante foi satisfeita em todas as iterações; num estudo mais detalhado dos casos em que estes métodos tiveram mau desempenho detectamos que o teste com o parâmetro α, falhou em muitas iterações (entre 20% e 90%);

f) o método de Schubert é caro computacionalmente, conforme era esperado, porque a matriz B_{k+1} é atualizada a partir de B_k e portanto, requer a resolução completa de um sistema linear a cada iteração; o requerimento de memória também é alto porque a matriz B_k precisa ser guardada na sua forma original. O fato mais grave contra Schubert, é que somente no teste com o problema 3 e $x^0=(0.3,\ldots,0.3)^T$ seu número de iterações foi menor que em qualquer outro $q-N$;

g) observa-se também uma certa uniformidade no desempenho dos métodos secantes, com relação ao número de iterações, em todos os testes, exceto no problema 3. Este é um fato interessante, se considerarmos que apenas Broyden e Schubert apresentam taxa superlinear de convergência. Com relação a DM, voltamos a afirmar que este desempenho pode ser explicado pelo fato de ser o caso limite de uma família com taxa superlinear de convergência. Conforme colocado na seção 3.10, era de se esperar que o desempenho de ACI fosse próximo ao de Broyden; esta expectativa não apenas se confirmou, como também foi superada, já que em alguns testes ACI conseguiu convergência num número de iterações inferior ao de Broyden. Já o bom desempenho dos métodos ED, EC e EL foi uma surpresa, notadamente no problema 7; neste teste, $J(x^*)$ é singular, e este fato particular pode ter uma explicação teórica para o desempenho superior de ED, EC e EL sobre os demais métodos secantes. Exceto no problema 4, o menor número de iterações entre os $q-N$ sempre coube a um dos três métodos: ED, EC e EL, o que é certamente uma boa informação sobre a família de métodos $FQNEF$ mas, torna difícil uma conclusão individual para cada um de seus métodos aqui estudados.

Os gráficos 6.3.1 e 6.3.2 representam o desempenho dos vários algoritmos locais com relação a iterações efetuadas e tempo de execução na resolução dos problemas 4 e 5 respectivamente com os mesmos dados usados nos testes apresentados na tabela 6.2.3 e com opções de execução sem restart e com restart pela eficiência. Estes gráficos evidenciam algumas conclusões, destacando:
- tempo alto de execução no método de Schubert; basicamente, o esforço computacional por iteração é equivalente ao de Newton, nestes testes, mas o número de iterações é maior o que resulta num tempo total de execução em Schubert consideravelmente maior que em Newton.

- a opção do restart pela eficiência parece ser vantajosa somente somente quando não há convergência nos $q-N$. Testes com os métodos ED, EC e EL na resolução do problema 4, confirmam a importância em se permitir que os $q-N$ efetuem um número de iterações consecutivas antes de se avaliar a melhora em $||F(x)||$; nota-se que na versão sem restart ED efetuou 8 iterações (uma iteração Newton) em 15s de execução, enquanto que na versão com restart pela eficiência, o número de iterações efetuadas caiu para 4 (das quais, duas são Newton, sendo que a segunda iteração Newton foi efetuada devido ao não-decréscimo em $||F(x)||$ na iteração ED anterior), mas, o tempo de execução aumentou para 22s. Os testes com EC e EL com restart pela eficiência não resultaram em redução no número de iterações efetuadas mas, o tempo de execução aumentou uma vez que foi executada uma iteração Newton a

mais que a execução sem restart.

Com relação aos algoritmos globais, nosso objetivo foi testar o comportamento dos $q - N$ em testes onde o método de Newton local teve desempenho ruim ou não convergiu. Analisando os resultados apresentados na tabela 6.2.4 concluímos:

a) comparando com a execução sem restart pode-se concluir que a estratégia tolerante teve sucesso sobre o desempenho dos métodos $q-N$ pois em aproximadamente 50% dos casos os algoritmos globais $q - N$ superaram o desempenho de Newton global, e, este resultado ocorreu porque poucas iterações especiais foram necessárias para colocar o método $q - N$ na trajetória correta;

b) no problema 6 todos os algoritmos globais convergiram para o mesmo mínimo local de $||F(x)||$; no problema 7, somente EL e EC encontraram a solução do sistema, os demais métodos (exceto ED) convergiram para diferentes minimizadores locais;

c) os testes com os algoritmos globais não apresentam diferenças significativas entre o desempenho dos métodos secantes e Newton Modificado. Comparando com o tempo de execução em Newton, temos que: NM foi superior em 4 testes, EC, EL, B e ACI foram superiores em 3 testes e, S, DM e ED foram superiores em 2 testes.

O gráfico 6.3.3 mostra o desempenho dos algoritmos globais aplicadas à resolução do problema 5 ($n = 100$, $b = 100$). Este teste é idêntico ao apresentado na tabela 6.2.4.

Os resultados com os algoritmos globais aplicados à resolução do problema 5 foram apresentados no gráfico 6.3.2 porque é um exemplo onde todos os algoritmos locais falharam, isto é, o número máximo de 100 iterações foi atingido sem obter uma aproximação em que se verificasse a convergência do tipo 0 ou tipo 1. Já os algoritmos globais obtiveram sucesso e no gráfico fica evidente que poucas iterações especiais foram necessárias para colocar os algoritmos locais na trajetória correta.

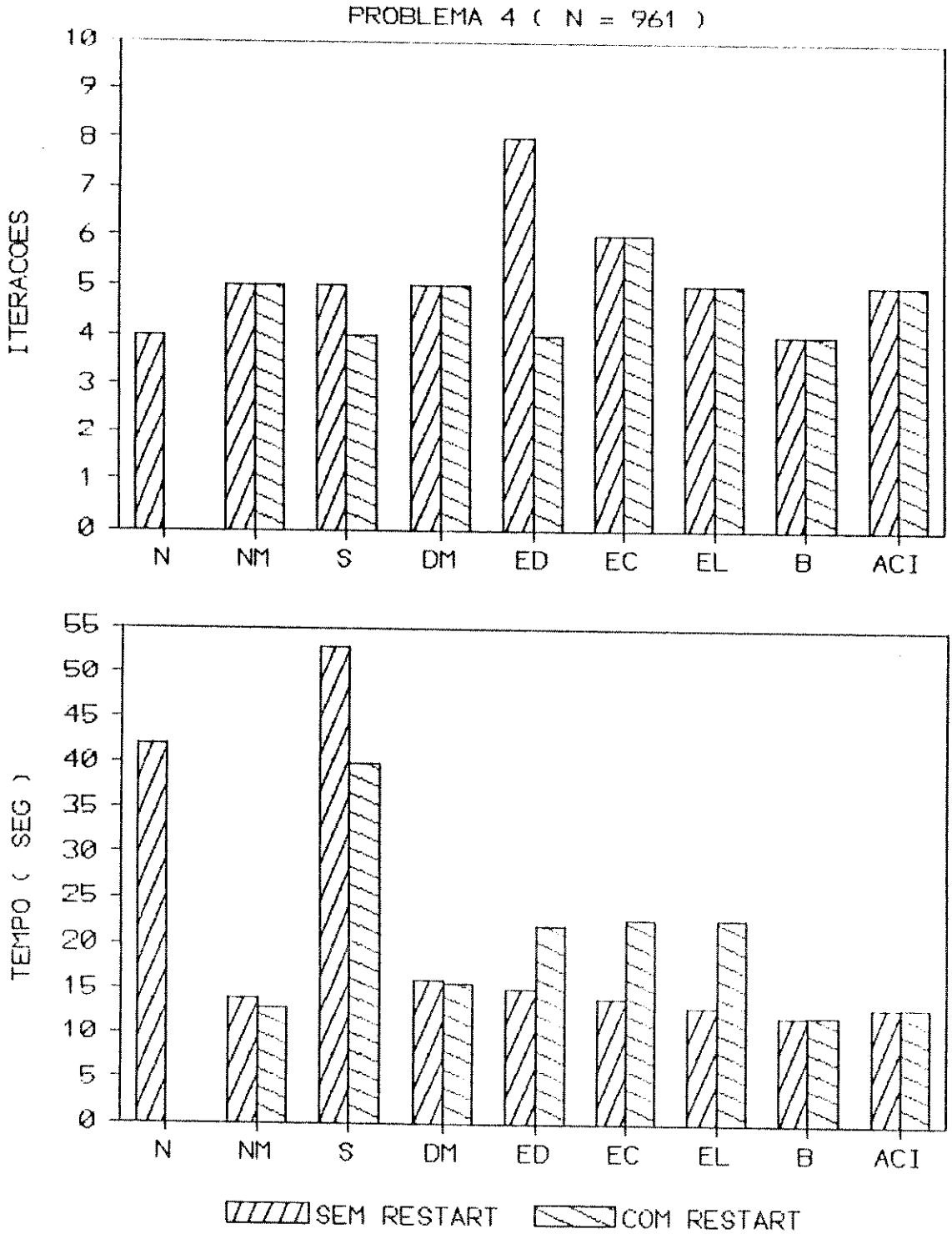

Gráfico 6.3.1. Algoritmos Locais Aplicados à resolução do problema 4.

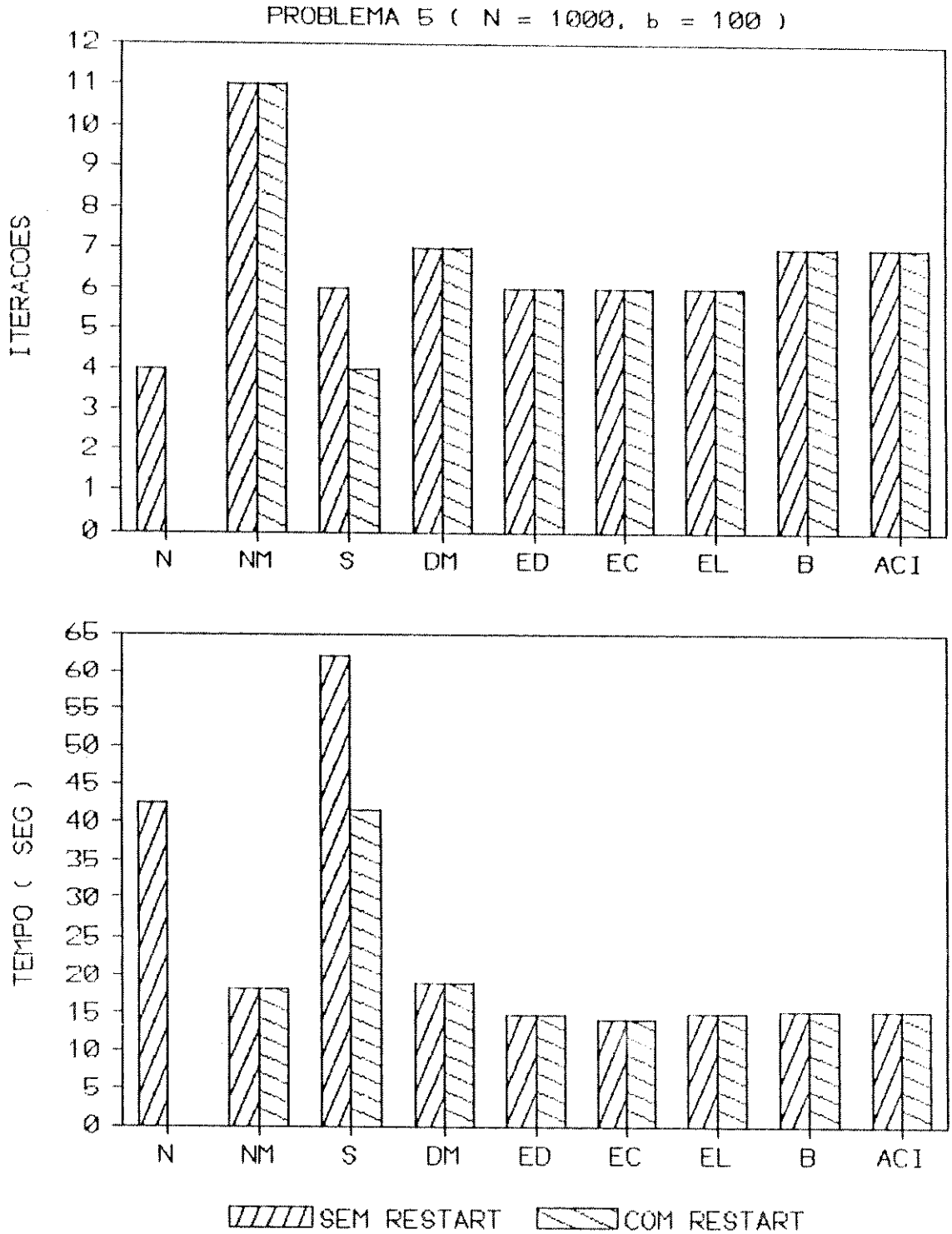

Gráfico 6.3.2. Algoritmos Locais Aplicados à resolução do problema 5.

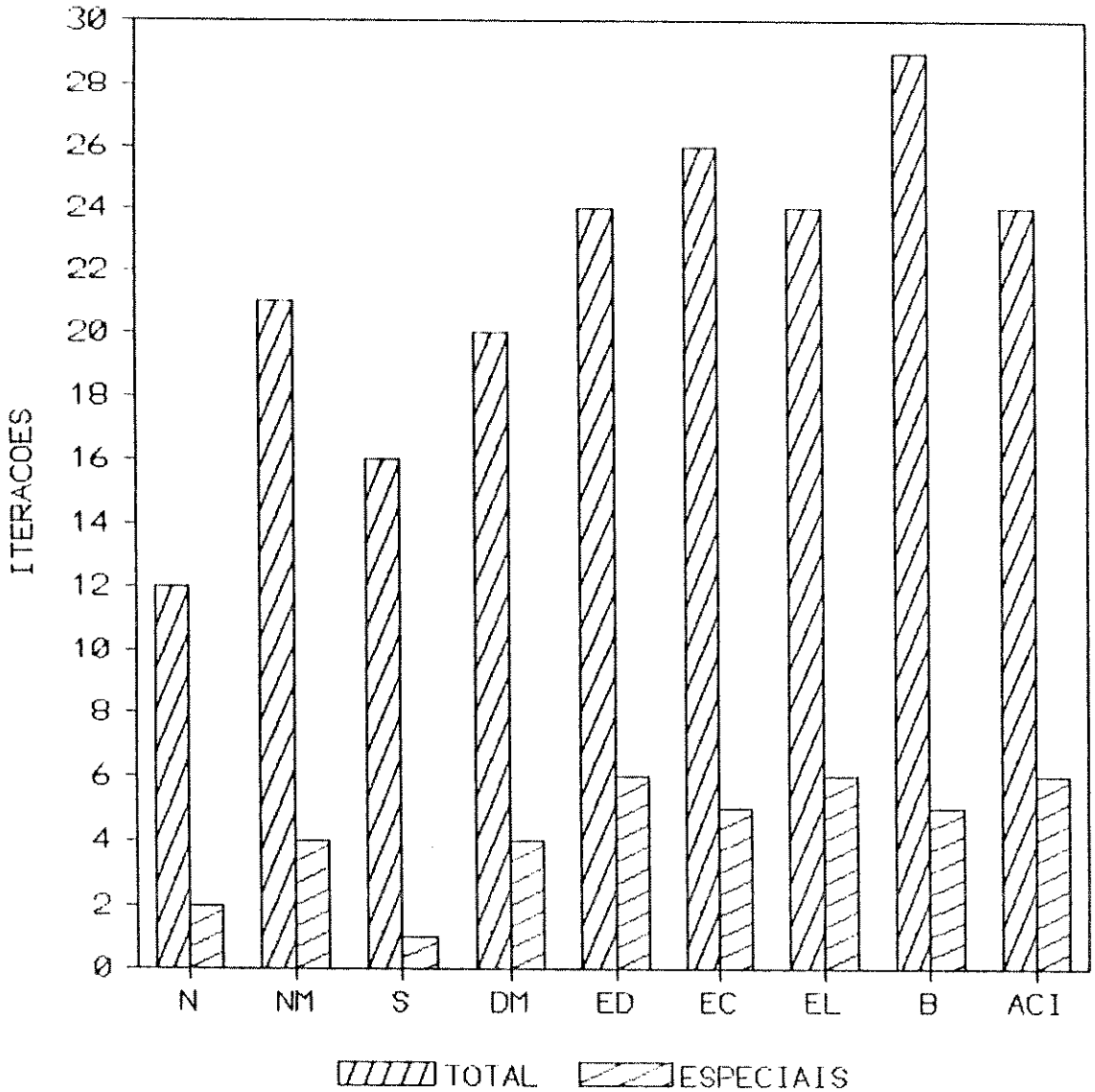

Gráfico 6.3.3. Algoritmos Globais Aplicados à resolução do problema 5 ($n = 100$, $b = 100$)

Finalmente, os resultados apresentados na tabela 6.2.5 mostram que a estrutura de dados fixada pela fatoração simbólica não apresentou um "overhead" de memória muito alto neste conjunto de problemas. A porcentagem de utilização efetiva da estrutura de dados variou no intervalo 55.9 % (problema 5, $n = 1000$) e 93.2 % (problema 6). A estrutura do Jacobiano do problema 5 é a que apresenta elementos com maior distância do elemento da diagonal, o que resulta num grande preenchimento na fase simbólica.

Observamos ainda o aproveitamento total da estrutura fixada para o fator L em todos os problemas exceto para o problema 5. Este fato se deve em parte à estrutura especial dos problemas: tridiagonal, banda 5, tridiagonal com faixa (laplaciano) e ao fato que a estrutura fixada para o fator L tende a apresentar um "overhead" menor que a estrutura fixada para o fator U.

6.4. Aplicação: Problema de fluxo de carga [15]

Fluxo de carga ou fluxo de potência é a solução para a condição de operação estática de um sistema de transmissão de potência elétrica.

O objetivo fundamental do cálculo de fluxo de carga é a determinação das tensões e das injeções de potência em todos os nós do sistema de transmissão (rede), sob determinadas condições de geração de carga.

As equações que modelam o comportamento dos principais componentes da rede são dadas por:

$$\begin{cases} P_k(\theta,\ v) = v_k \sum_{m \in K_k} v_m (G_{km}\ \cos\ \theta_{km} + B_{km}\ \text{sen}\ \theta_{km}) \\ \\ Q_k(\theta,\ v) = v_k \sum_{m \in K_k} v_m (G_{km}\ \text{sen}\ \theta_{km} - B_{km}\ \cos\ \theta_{km}) \end{cases}$$

onde: $P_k(\theta,\ v)$ e $Q_k(\theta,\ v)$ representam as injeções líquidas de potência ativa e reativa respectivamente;
v_k: representa a magnitude da tensão no nó k;
G_{km} e B_{km}: são as componentes da matriz de admitância: $Y_{km} = G_{km} + iB_{km}$;
$\theta_{km} = \theta_k - \theta_m$ é a abertura angular no ramo km e
K_k é o conjunto dos nós vizinhos ao nó k.

Na formulação mais simples do problema a cada nó são associadas quatro variáveis: P_k, Q_k, θ_k e v_k. Os nós do sistema são classificados em três tipos,

dependendo das quantidades que entram no problema como dados e das que entram como incógnitas:

nós folga: nós k onde v_k e θ_k são dados;
nós do tipo A: nós k onde P_k e Q_k são dados;
nós do tipo B: nós k onde P_k e v_k são dados.

Se N representa o total de nós, o sistema de fluxo de potência é formado por $2N$ equações para as quais as variáveis são precisamente as incógnitas em cada nó.

Após algumas simplificações algébricas, as equações correspondentes aos nós folga podem ser eliminadas, assim como pode ser eliminada uma equação para cada nó do tipo B. Sabendo que existe um nó folga e supondo que existem S nós do tipo B, teremos um sistema com $n \equiv 2N - S - 2$ equações e n variáveis.

A estrutura da matriz Jacobiana do sistema é determinada pela estrutura das matriz de admitância Y. Uma estrutura típica de uma matriz Jacobiana para um sistema com $n = 30$, assim como a estrutura fixada pela fatoração simbólica é dada na figura 6.4.1.

O problema de fluxo de carga foi resolvido para sistemas com:
30 nós e $n = 54$;
118 nós e $n = 182$;
1138 nós e $n = 2190$.

A tabela 6.4.2 apresenta o desempenho do método de Newton local e métodos quase Newton com opção de execução sem restart, na resolução dos sistemas com 30 e 118 nós. O desempenho de cada método é representado por $(CP,\ k,\ t)$ que significa que o processo parou com código CP, após k iterações e t segundos de tempo de execução. CP tem o mesmo significado que na secção 6.3.

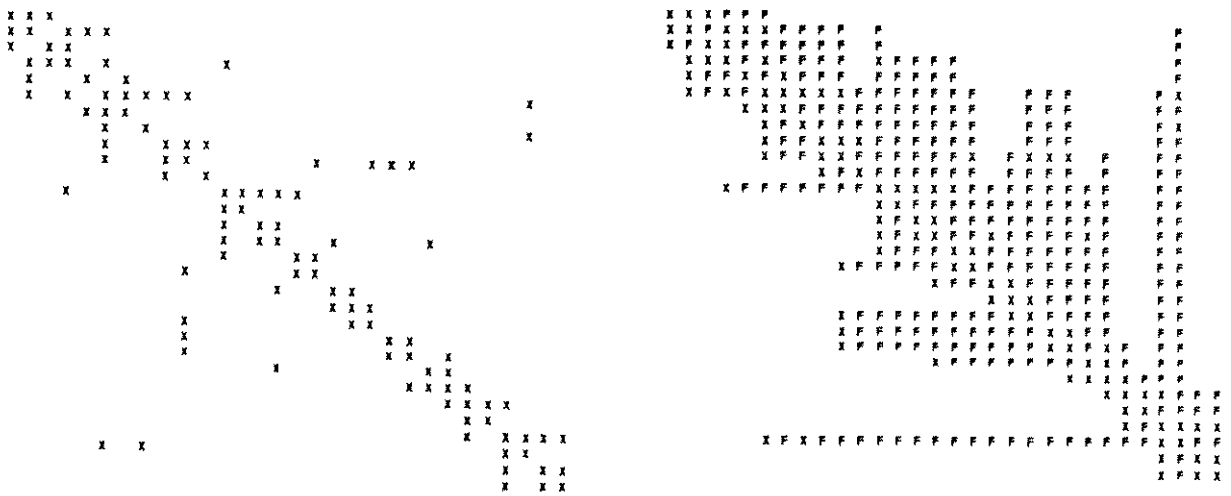

Figura 6.4.1. Estrutura original e estrutura de dados para a fatoração LU $(n = 30)$

O tempo total na fase simbólica, em cada caso, foi:

Problema	Fase simbólica
N = 30	0.20 s
N = 118	1.08 s
N = 1138	138.4 s

Tabela 6.4.1. Tempo na fase simbólica

MÉTODO	N = 30 (n = 54)	N = 118 (n = 182)
N	0,3, 0.61	0,3,5.08
NM	0,5,0.44	0,6,3.71
S	0,5,0.82	0,6,6.91
DM	0,6,0.54	1,6,4.01
ED	0,14,0.96	1,29,13.4
EC	1,26,1.68	3,30,14.0
EL	0,9,0.65	1,12,6.27
B	0,5,0.44	0,5,3.32
ACI	0,4,0.34	0,6,3.85

6.4.2. Desempenho dos métodos para $N = 30$ e $N = 118$

As tolerâncias usadas nestes testes foram: $\varepsilon_1 = \varepsilon_2 = 10^{-4}$; $\beta = 10$; e $Tolsing = (espmaq)^{1/2}$.

Entre os $q-N$ os melhores desempenhos ficam para os métodos: ACI, B, NM e DM, sendo que ACI tem o menor tempo para $N = 30$ e Broyden o menor tempo para $N = 118$.

Os métodos ED, EC e EL, não conseguem bons resultados nestes testes; a ocorrência de fator singular é registrada em EC e EL nos dois problemas e em ED para o problema com $N = 118$.

No teste com $N = 1138$ ($n = 2190$) apenas Newton, Broyden e ACI obtêm a solução; nos demais métodos a execução foi interrompida após a execução de 15 iterações.

Neste teste, as tolerâncias usadas foram: $\varepsilon_1 = 10^{-2}$; $\varepsilon_2 = 10^{-4}$; $\beta = 10$ e $Tolsing = 10^{-14}$.

O desempenho de cada método é representado por (CP, k, t):

Método	Desempenho	Iteração Típica
N	0,4,2349.	587.4 s
B	0,12,1092.	42.5 s
ACI	0,12,1066.	42.5 s

Tabela 6.4.3. Desempenho dos métodos para $N = 1138$

O tempo alto de execução de uma iteração Newton (aproximadamente 10 minutos) se deve essencialmente ao tempo gasto na fatoração LU. A estrutura original

da Jacobiana é tal que resulta num grande preenchimento na fase simbólica: a estrutura original é 0.32% densa e a estrutura fixada após a fase simbólica é 15% densa.

Estas posições fixadas na fase simbólica não são totalmente utilizadas, conforme mostra a tabela 6.4.4, mas, as posições previstas fazem com que a fase numérica opere com "elementos nulos" o que implica num esforço computacional alto para a fatoração LU.

A tabela 6.4.4 mostra a porcentagem de utilização da estrutura de dados para a fatoração LU. Os resultados apresentados se referem à iteração Newton em que a fatoração LU gerou o maior número de elementos não nulos.

		FATORAÇÃO SIMBÓLICA Nº DE ELEM. NÃO NULOS			% de UTILIZAÇÃO (FAT. LU x FAT. SIMB.)		
PROB.	ELEM. NÃO NULOS	L	U	TOTAL	L	U	TOTAL
N = 30	380	461	917	1378	98.7 %	55.5 %	69.9 %
N = 118	1052	2451	4437	6888	98.5 %	58.5 %	72.7 %
N = 1138	15280	228042	492297	720249	66.7 %	35.5 %	45.4 %

Tabela 6.4.4. Porcentagem de utilização da estrutura de dados para a fatoração LU

6.5. Comparação com MA28

Conforme esclarecemos no capítulo 5, a opção pela fatoração LU com estratégia de pivoteamento parcial nos conduziu à escolha de uma estrutura de dados estática.

O pacote MA28 de Harwell [12] é um conjunto de rotinas Fortran para resolução de sistemas lineares escrito por I. S. Duff. Em MA28 a estrutura de dados é dinâmica e, a escolha do pivô é feita de modo a manter ao máximo a esparsidade da matriz de coeficientes. Para que o pivô não seja pequeno a ponto de provocar instabilidade numérica, o elemento não nulo escolhido como pivô satisfaz a desigualdade:

$$|a_{kk}^{(k)}| \geq u \max_j |a_{kj}^{(k)}| \quad \text{ou} \qquad (6.5.1)$$

$$|a_{kk}^{(k)}| \geq u \max_i |a_{ik}^{(k)}|$$

onde $a_{kj}^{(k)}$ é o elemento da matriz de coeficientes no início da etapa k do processo de eliminação, e, o parâmetro u, $u \in (0, 1)$, é fixado pelo usuário.

Em MA28 existe a possibilidade de fatorar uma matriz empregando a sequência pivotal previamente estabelecida pela fatoração LU de uma outra matriz com estrutura idêntica. Esta opção pode ser escolhida no caso dos métodos de Newton e Schubert que requerem o cálculo da fatoração LU a cada iteração; na primeira iteração, a sequência pivotal é obtida e fixada de forma que em todas as iterações subsequentes a sequência pivotal será a mesma. Este processo tem a vantagem de reduzir consideravelmente o tempo de execução, mas, pode provocar instabilidades e até mesmo interrupção na execução se na sequência fixada surgir um pivô muito pequeno.

O trabalho de mestrado de M. C. Zambaldi, [43], consiste essencialmente em analisar o desempenho dos métodos quase Newton e Newton utilizando MA28 para resolução dos sistemas lineares.

O pacote SNLDIN, [43], reúne as rotinas de Rouxinol, para os métodos Newton e quase Newton, adaptados à estrutura de dados de MA28.

As tabelas 6.5.1-6.5.3 apresentam os principais resultados da comparação entre SNLDIN e Rouxinol.

O desempenho de cada método é representado por (k, t) que significa que o algoritmo efetuou k iterações até conseguir convergência (do tipo 0 ou tipo 1) em t segundos de execução. Em SNLDIN, o parâmetro u, necessário para MA28, foi fixado em 0.1 em todos os testes; para os métodos de Newton e Schubert apresentamos os resultados onde a sequência pivotal foi fixada após a iteração inicial e, abaixo, entre parênteses o resultado para o caso em que a sequência pivotal é calculada em cada iteração. Os tempos, t, nos testes com Rouxinol incluem o tempo na fase simbólica.

A tabela 6.5.1 apresenta os resultados para a resolução dos problemas 1, 2 e 5 (descritos na secção 6.2). O problema 4 foi resolvido com diferentes dimensões e os resultados são mostrados na tabela 6.5.2. Finalmente, a tabela 6.5.3 apresenta os

resultados dos testes com o problema de fluxo de carga descrito na secção 6.4.

Os parâmetros e chutes iniciais são os mesmos usados para os testes com os algoritmos locais, no caso dos problemas 1, 2, 4 e 5. Para o problema de fluxo de carga, usamos os mesmos parâmetros colocados na secção 6.4.

MÉTODO	PROBLEMA 1 (n=5000)		PROBLEMA 2 (n=5000)		PROBLEMA 5 (n=1000, b=100)	
	SNLDIN	ROUXINOL	SNLDIN	ROUXINOL	SNLDIN	ROUXINOL
N	3, 19.3 (3, 29.6)	3, 5.82	4, 94.6 (4, 139.)	4, 26.2	4, 95.5 (4, 253.)	4, 46.7
NM	9, 13.4	9, 6.93	17, 58.	17, 30.	11, 65.8	11, 24.5
S	6, 32.5 (6, 60.3)	6, 11.	9, 165. (9, 312.)	9, 56.1	6, 117. (6, 376.)	6, 64.9
DM	5, 13.1	5, 6.52	11, 55.4	11, 31.4	7, 66.4	7, 25.1
ED	5, 11.6	5, 5.63	6, 41.3	6, 17.8	6, 62.9	6, 21.5
EC	5, 11.7	5, 5.51	6, 41.3	6, 17.8	6, 64.5	6, 21.5
EL	5, 11.6	6,6.08	6, 41.3	6, 17.8	6, 64.4	6, 21.4
B	5, 12.4	5, 7.09	9, 47.6	9, 22.9	7, 64.9	7, 22.4
ACI	5, 12.2	5, 6.64	9, 46.9	9, 24.4	7, 64.8	7, 21.2

Tabela 6.5.1. Desempenho dos métodos Newton e quase Newton em SNLDIN e Rouxinol (problemas 1, 2 e 5)

MÉTODO	PROBLEMA 4 (n=225)		PROBLEMA 4 (n=961)		PROBLEMA 4 (n=1600)	
	SNLDIN	ROUXINOL	SNLDIN	ROUXINOL	SNLDIN	ROUXINOL
N	3, 3.81 (3, 6.87)	3, 2.87	4, 35.4 (4, 74.8)	4, 47.5	4, 81.0 (4, 169.)	4, 137.
NM	5, 2.45	5, 2.05	5, 19.5	5, 20.9	5, 43.9	5, 58.5
S	5, 5.21 (5, 10.9)	4, 3.59	5, 40.9 (5, 96.5)	5, 56.6	5, 93.9 (5, 215.)	5, 167.2
DM	5, 2.61	5, 2.08	4, 20.	5, 23.6	4, 44.9	5, 66.1
ED	6, 2.54	6, 2.21	5, 19.7	8, 23.5	7, 44.9	9, 62.8
EC	8, 2.57	6, 2.1	6, 19.8	6, 21.6	6, 44.4	6, 60.7
EL	6, 2.54	6, 2.01	5, 19.6	5, 21.3	5, 43.9	5, 58.4
B	4, 2.44	4, 2.12	4, 19.5	4, 20.6	4, 43.6	4, 56.4
ACI	5, 2.45	5, 1.95	5, 19.6	5, 20.	5, 43.9	5, 55.8

Tabela 6.5.2. Desempenho dos métodos Newton e quase Newton em SNLDIN e Rouxinol no problema 4

MÉTODO	$N = 30 (n = 54)$		$N = 118 (n = 182)$		$N = 1138 (n = 2190)$	
	SNLDIN	ROUXINOL	SNLDIN	ROUXINOL	SNLDIN	ROUXINOL
N	3, 0.85 (3, 1.41)	3, 0.79	3, 4.35 (3, 5.37)	3, 6.23	4, 420. (4, 443.)	4, 2471.
NM	5, 0.59	5, 0.62	6, 3.49	6, 4.89	-	-
S	5, 1.07 (5, 1.8)	5, 1.0	6, 5.14 (5, 6.6)	6, 8.1	-	-
DM	5, 0.59	6, 0.72	5, 3.27	6, 5.19	-	-
ED	9, 0.71	14, 1.14	9, 4.51	29, 14.6	-	-
EC	21, 1.07	26, 1.86	-	-	-	-
EL	7, 0.65	9, 0.83	23, 9.05	12, 7.45	-	-
B	5, 0.59	5, 0.62	5, 3.31	5, 4.5	12, 440.	12, 1213.
ACI	4, 0.56	4, 0.52	6, 3.49	6, 5.0	12, 430.5	12, 1186.7

Tabela 6.5.3. Desempenho dos métodos Newton e quase Newton em SNLDIN e Rouxinol no problema de Fluxo de Carga

Em todos os testes apresentados na tabela 6.5.1 o número de iterações foi o mesmo em SNLDIN e Rouxinol, mesmo nos métodos de Newton e Schubert quando foi fixada a sequência pivotal em SNLDIN. Isto ocorreu porque os chutes iniciais estavam em vizinhanças convenientes da solução. Com relação ao tempo de execução, Rouxinol foi superior em todos os testes mesmo nos casos em que a sequência pivotal for fixada na iteração inicial.

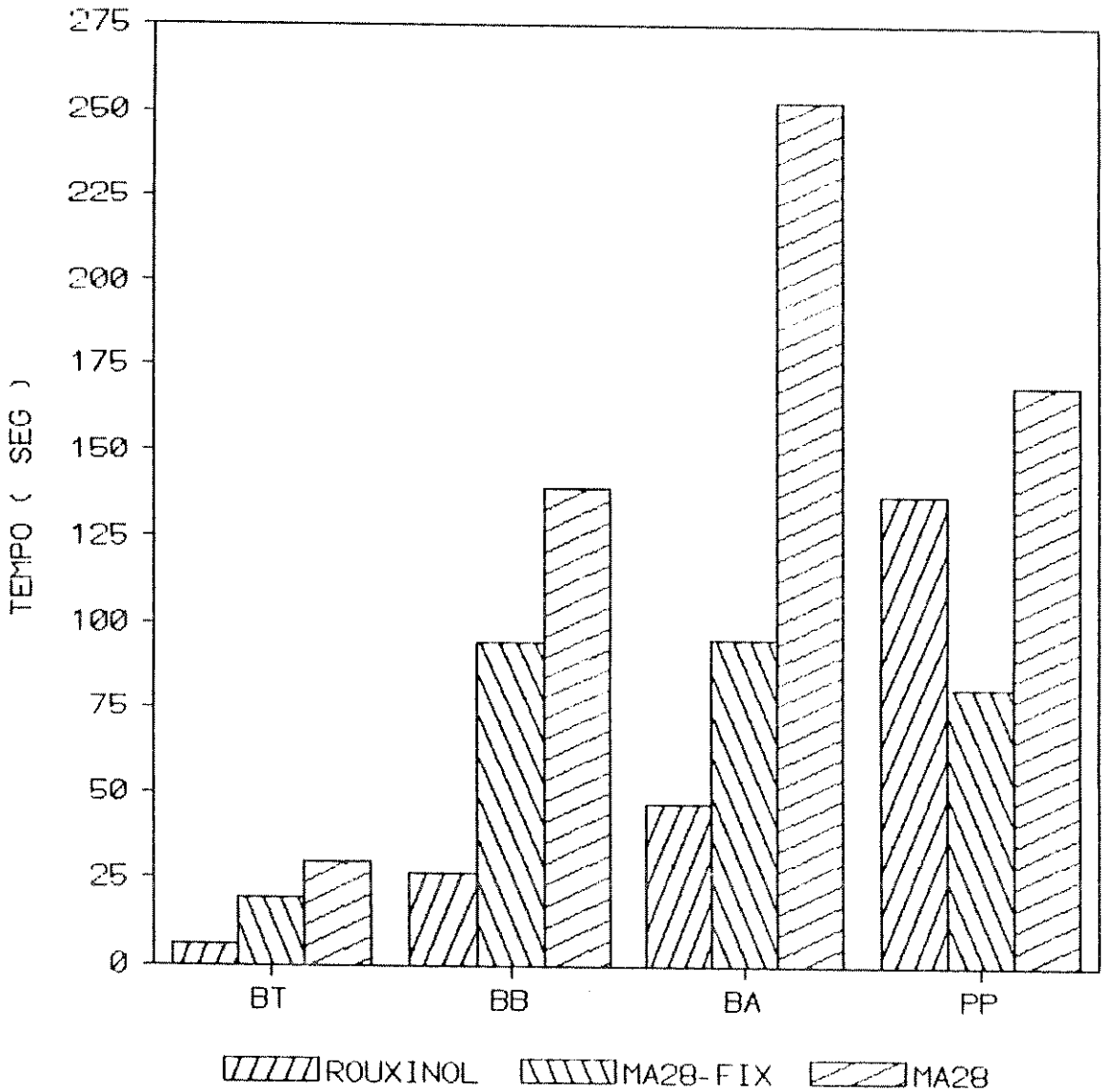

Gráfico 6.5.1. Comparação entre Newton/Rouxinol e Newton/SNLDIN

Para os testes com o problema 4, problema de Poisson, a medida em que a malha se torna mais fina, aumentam as distâncias das faixas em relação a diagonal (ver figura 6.2.3) o que provoca preenchimentos cada vez maiores na fase simbólica de Rouxinol; por esta razão os tempos de execução em SNLDIN são menores que em Rouxinol para $n = 961$ e $n = 1600$. Observamos contudo, que para os métodos de Newton e Schubert o tempo de execução em Rouxinol é menor que em SNLDIN quando a sequência pivotal é calculada em cada iteração.

O gráfico 6.5.1 apresenta a comparação entre Rouxinol e SNLDIN quando é escolhido o método de Newton para a resolução dos problemas 1 (BT) e 2 (BB) com dimensão $n = 5000$;problema 4 (PP) com dimensão $n = 1600$ e problema 5 (BA) com $n = 1000$ e $b = 100$.

Em SNLDIN o método de Newton foi executado com as duas opções: sequência pivotal fixada ($MA28 - FIX$) e sequência pivotal calculada em cada iteração ($MA28$).

No gráfico é mais evidente a superioridade de Rouxinol sobre SNLDIN notadamente na versão SNLDIN sem fixar a sequência pivotal.

Finalmente, no problema de fluxo de carga as diferenças entre os resultados com SNLDIN e Rouxinol se torna mais sensível à medida em que aumenta a dimensão do problema e este fato se justifica pelas mesmas razões colocadas no problema 4.

A nível de comparação, consideramos válidos os resultados de SNLDIN sem fixar a sequência pivotal uma vez que este esquema pode resultar em instabilidades desastrosas no decorrer da execução. Sob este ponto de vista, o desempenho de Rouxinol é superior em todos os testes com os métodos de Newton e Schubert nos problemas 1, 2, 4 e 5. Já para os demais $q - N$ o desempenho é superior nos problemas 1, 2 e 5 e, no problema 4, SNLDIN e Rouxinol têm quase o mesmo desempenho com vantagem para SNLDIN.

6.6. Conclusões e Trabalhos Futuros.

a) ordenação de linhas e colunas em B:

a opção pela estrutura estática de dados trouxe bons resultados nos casos em que a matriz Jacobiana apresenta estruturas especiais; já nos problemas em que esta estrutura está originalmente espalhada, os preenchimentos provocados na fase

simbólica prejudicam a fase numérica. No primeiro algoritmo proposto por George e Ng, [20] o esquema de ordenação "minimum degree" de George e Liu [18, 19] é aplicado sobre B^TB de modo que os fatores de Choleski sejam o mais esparso possível. Já para o segundo algoritmo de George e Ng [21] que foi implementado em Rouxinol, a fatoração é realizada sobre a matriz B (e não sobre B^TB) e seria então essencial obter um bom esquema de ordenação de linhas e colunas de modo a se obter o menor preenchimento na fase simbólica e consequentemente em qualquer fatoração LU. George e Ng [21] apresentam uma sugestão para tal estratégia e um possível trabalho seria explorar a idéia apresentada no sentido de se obter um algoritmo eficiente, e, incorporar este algoritmo em Rouxinol;

b) aperfeiçoamento da estratégia de globalização tolerante:

em alguns testes com os algoritmos globais o processo gerou uma sequência que convergiu a um ponto estacionário de $f(x) = \frac{1}{2}||F(x)||_2^2$ que não era solução para $F(x) = 0$; trabalhamos atualmente no aperfeiçoamento desta estratégia. A modificação consiste basicamente em exigir que uma iteração especial gere uma aproximação x^{k+1} tal que:

$$f(x^{k+1}) = \min_{\alpha} f(x^k + \alpha\, s_k) \tag{6.6.1}$$

A filosofia por trás da exigência é que uma vez que uma iteração especial é executada e consequentemente uma iteração Newton é efetuada para obter a direção s_k, deve-se explorar ao máximo esta direção para evitar a ocorrência de mínimos locais de $f(x)$;

c)métodos Newton - Inexatos:

tanto a estrutura de dados estática quanto a estrutura de dados dinâmica impõem limites sobre a resolução dos sitemas lineares, em termos de memória e tempo de execução; isto é, se n é muito grande e $J(x)$ não apresenta uma estrutura de esparsidade favorável, é impossível resolver o sistema linear de Newton:

$$J(x^k)s = -F(x^k) \tag{6.6.2}$$

usando métodos diretos, qualquer que seja o aproveitamento da esparsidade de $J(x)$. Para estes casos a única opção são os metodos iterativos, o que conduz aos métodos Newton - Inexatos.

Os métodos iterativos só são eficientes quando implementados com pré-condicionadores adequados.

Martínez, [31] desenvolveu recentemente uma teoria sobre a técnica de precondicionadores secantes e propõe um algoritmo ao qual se conjetura taxa superlinear de convergência. A proposta atual é desenvolver e incorporar em Rouxinol, um "software" baseado nesta teoria.

Referências

[1] Brameler, A., Allan, R. N., Haman, Y. M. [1976]: *Sparsity*, New York Pitman.

[2] Broyden, C. G. [1965]: A class of methods for solving nonlinear simultaneous equations, Math. Comput. 19, pp. 577-593.

[3] Broyden, C. G. [1971]: The convergence of an algorithm for solving sparse nonlinear systems, Mathematics of Computation 25, pp. 285-294.

[4] Broyden, C. G.; Dennis, J. E., Jr.; Moré, J. J. [1973]: On the local and superlinear convergence of quasi-Newton methods, J. Inst. Math. Appl. 12, pp. 223-245.

[5] Chadee, F. F. [1985]: Sparse quasi-Newton methods and the continuation problem, TR SOL nº 85-8, Dept. of Operations Research, Stanford University.

[6] Coleman, T. C. [1984]: *Large Sparse Numerical Optimization*, Lectures Notes in Computer Science, nº 165, Springer-Verlag.

[7] Dennis, J. E.; Martínez, J. M. [1990]: Numerical methods for solving nonlinear systems, em Handbook of Numerical Analysis, P. G. Ciarlet and J. L. Lions (editors), Elsevier-North-Holland (em preparação).

[8] Dennis, J. E., Jr.; Marwil, E. S. [1982]: Direct secant updates of matrix factorizations, Mathematics of Computation 38, pp. 459-476.

[9] Dennis, J. E., Jr.; Moré, J. J. [1974]: A characterization of superlinear convergence and its application to quasi-Newton methods, Math. Comp. 28, pp. 549-560.

[10] Dennis, J. E. Jr.; Moré, J. J. [1977]: Quasi-Newton methods, motivation and theory, SIAM Review 19, pp. 46-89.

[11] Dennis, J. E., Jr.; Schnabel, R. B. [1983]: *Numerical Methods for Unconstrained Optimization and Nonlinear Equations,* Prentice Hall, Englewood Cliffs, N. J.

[12] Duff, I. S. [1977]: MA28 - a set of Fortran subroutines for sparse unsymmetric linear equations. AERE R8730, HMSO, London.

[13] Duff, I. S. [1981]: On algorithms for obtaining a maximum transversal, ACM Transaction on Mathematical Software, 7, pp. 315-330.

[14] Duff, I. S.; Erisman, A. M.; Reid, J. K. [1986]: Direct Methods for Sparse Matrices, Clarendon Press, Oxford.

[15] Duran, A. C. [1990]: Resolução de sistemas não lineares esparsos: sua aplicação na resolução do problema de fluxo de carga em redes de energia elétrica, Tese de Mestrado,Depto de Matemática Aplicada, IMECC-UNICAMP, Campinas, Brasil.

[16] Forsythe, G.; Moler, C. B. [1967]: *Computer Solution of Linear Algebric Equations*, Prentice-Hall, New Jersey.

[17] Gay, D. M. [1979]: Some convergence properties of Broyden's method, SIAM J. Numer. Anal. 16, pp. 623-630.

[18] George, A.; Liu J. W. H. [1980]: A minimal storage implementation of the minimum degree algorithm, SIAM J. Numer. Anal. 17, pp. 282-299.

[19] George, A.; Liu, J. W. H. [1981]: *Computer Solution of Large Sparse Positive Definite Systems*, Prentice-Hall, Englewood Cliffs, N. J.

[20] George, A.; Ng, E. [1985]: An implementation of Gaussian elimination with partial pivoting for sparse systems, SIAM J. Sci. Stat. Comput. 6, pp. 390-409.

[21] George, A.; Ng, E. [1987]: Symbolic factorization for sparse Gaussian elimination with partial pivoting, SIAM J. Sci. Stat. Comput. 8, pp. 877-898.

[22] Golub, G. H.; Van Loan, Ch. F. [1983]: *Matrix Computations*, The Johns Hopkins Univesrity Press, Baltimore.

[23] Gomes-Ruggiero, M. A.; Martínez, J. M.: The column updating method for solving non linear equations in Hilbert space, RAIRO Mathematical Modeling and Numerical Analysis (por aparecer)

[24] Gomes-Ruggiero, M. A.; Martínez, J. M.; Moretti, A. C.: Comparing algorithms for solving sparse nonlinear systems of equations, SIAM J. Sci. Stat. Comput. (por aparecer)

[25] Griewank, A. [1986]: The "Global" convergence of Broyden - like methods with a suitable line search, J. Australian Mathematical Society Ser. B 28, pp. 75-92.

[26] Jennings, A. [1977]: *Matriz computations for Engineers and Scientists*, John Wiley & Sons.

[27] Johnson, G. W.; Austria, N. H. [1983]: A quasi-Newton method employing direct secant updates of matrix factorizations, SIAM J. Numer. Anal. 20, pp. 315-325.

[28] Martínez, J. M. [1983]: A quasi-Newton method with a new updating for the LDU factorization of the approximate Jacobian, Matemática Aplicada e Computacional 2, pp. 131-142.

[29] Martínez, J. M. [1987]: Quasi-Newton Methods with Factorization Scaling for Solving Sparse Nonlinear Systems of Equations, Computing 38, pp. 133-141.

[30] Martínez, J. M. [1984]: A quasi-Newton method with modification of one column per iteration, Computing, 33, pp. 353-362.

[31] Martínez, J. M. [1990]: Local convergence theory of inexact Newton methods based on structured least change updates, Math. Comput., 5, nº 191.

[32] Martínez, J. M. [1990]: A family of quasi-Newton method for nonlinear equation with direct secant updates of matrix factorizations, SIAM J. Anal., (por aparecer).

[33] Marwil, E. S. [1979]: Convergence results for Schubert's method for solving sparse nonlinear equations, SIAM J. Numer. Anal. 16, pp. 588-604.

[34] Matthies, H.; Strang, G. [1979]: The solution of nonlinear finite element equations, Int. J. on numerical Methods in Engineering 14, pp. 1613-1626.

[35] Moré, J. J.; Trangestein, J. A. [1976]: On the global convergence of Broyden's method, Mathematics of Computation 30, pp. 523-540.

[36] Ortega, J. M.; Rheinbolt, W. C. [1970]: *Iterative Solution of Nonlinear Equations in Several Variables*, Academic Press, New York.

[37] Osterby, O., Zlatev, Z. [1982]: *Direct Methods for Sparse Matrices*, Lectures Notes in Computer Science, nº 157, Springer-Verlag.

[38] Ostrowski, A. M. [1973]: Solution of equations in Euclidean and Banach spaces, Academic Press, New York and London.

[39] Sachs, E. [1986]: Broyden's method in Hilbert space, Mathematical Programming, 35, pp. 71-82.

[40] Schubert, L. K. [1970]: Modification of a quasi-Newton method for nonlinear equations with a sparse Jacobian, Math. Comput. 24, pp. 27-30.

[41] Schwandt, H. [1984]: An interval arithmetic approach for the construction of an almost globally convergent method for the solution of the nonlinear Poisson equation on the unit square, SIAM J. Sci. Stat. Comput. 5, pp. 427-452.

[42] Toint, Ph. L. [1986]: Numerical solution of large sets of algebraic nonlinear equations, Mathematics of Computation 16, pp. 175-189.

[43] Zambaldi, M. C. [1990]: Estruturas estáticas e dinâmicas para resolver Sistemas Não Lineares esparsos, Tese de Mestrado, Departamento de Matemática Aplicada, UNICAMP, Campinas, Brasil.

[44] Zlatev, Z. [1980]: On some pivotal strategies in Gaussian elimination by sparse technique, SIAM J. Numer. Anal., 17, pp. 18-30.