Métodos quase-Newton para Minimização de Funções e Aplicação a Aproximação por Penalização Interna do Problema de Programação Linear

Este exemplar corresponde a redação final da tese devidamente corrigida e defendida pela Srta. Denise Pizarro Vieira e aprovada pela Comissão Julgadora.

Campinas, 18 de Setembro de 1990.

Prof. Dr. José Mário Martínez
(Orientador)

Dissertação apresentada ao Instituto de Matemática, Estatística e Ciência da Computação, UNICAMP, como requisito parcial para obtenção do Título de Mestre em Matemática Aplicada.

Métodos quase-Newton para Minimização de Funções e Aplicação a Aproximação por Penalização Interna do Problema de Programação Linear

Denise Pizarro Vieira

AGRADECIMENTOS

Ao concluir este trabalho agradeço:

A José Mário Martínez pela orientação, tolerância e atenção.

Ao CNPq e à Capes pelo suporte financeiro.

Aos professores do Departamento de Matemática Aplicada, pelo bom relacionamento e por tornarem-me capaz de realizar este trabalho.

Em especial, a Profa. Valéria Podestá-Gomes pelo incentivo, pela sincera amizade e pela imensa dedicação no desenvolvimento deste trabalho.

Ao Prof. Petrônio Pulino pelo estímulo.

Aos funcionários da secretaria, a Fátima e ao Dorival, que nos prestaram vários serviços.

A Clarice Favaretto pelas sugestões construtivas.

Aos meus amigos do mestrado que, de alguma maneira, auxiliaram no cumprimento deste trabalho. Estejam certos de que este sonho realizado constitui a presença marcante de cada um.

Em particular, a Roseli e Regina pelo carinho e pelo afeto. Sobretudo, pela amizade e pelos inúmeros momentos de alegria em Campinas.

Aos meus irmãos, Deise e Douglas, pelo apoio e por não me deixarem desistir de meus objetivos.

A Arlindo Vieira

e Vanda Pizarro Vieira,

meus pais.

A Hilton (Pepo).

Existem duas maneiras de passar pela vida sem problemas:
acreditar em tudo ou duvidar de tudo;
de ambos os modos se evita pensar.
"**Alfred Korzybski**"

Conteúdo

Introdução

Esta tese trata da resolução de problemas de Programação Linear por Penalização Interna.

Até a publicação do algorítmo de Karmarkar em 1984, não havia tentativas vitoriosas no sentido de aplicar técnicas de programação não linear para resolver problemas lineares. Nesses últimos anos, pesquisadores de todo o mundo fazem implementação de métodos de pontos interiores, e existe uma grande discussão a respeito do desempenho dos novos métodos, em comparação ao método Simplex.

Em nosso trabalho resolvemos problemas não lineares que são uma aproximação de problemas lineares usando uma penalização do tipo *barreira*. Os métodos utilizados para resolver esses problemas não lineares são o método de Newton e alguns métodos quase-Newton. Dentre os quase-Newton, consideramos: BFGS, DFP, Greenstadt, QN Diagonal, Correção de Posto 1 e um método desenvolvido por Martínez (quase-Newton Estruturado).

O nosso objetivo é verificar qual desses métodos é mais eficiente para resolver o problema penalizado.

A função barreira utilizada é a logarítmica, que foi inicialmente sugerida por Frisch, em 1955.

No Capítulo 1 apresentamos a função barreira, a aplicação da função barreira logarítmica a problemas lineares e um teorema que mostra que a solução ótima do problema irrestrito é uma boa aproximação da solução ótima do problema restrito.

No Capítulo 2, analisamos uma família de métodos quase-Newton introduzida por Martinez e mostramos como alguns métodos quase-Newton se enquadram nessa família.

No Capítulo 3 apresentamos o método **quase-Newton Diagonal** e o método **Correção Posto 1** que são métodos que não se enquadram na família descrita no capítulo anterior.

No Capítulo 4 descrevemos o teorema de convergência global adaptado à função penalizada.

No Capítulo 5, temos a documentação das subrotinas e no Capítulo 6, alguns experimentos são mostrados.

Finalmente, no Capítulo 7 apresentamos as conclusões finais.

Capítulo 1

1 Método de Barreira Aplicado a um Problema Linear

1.1 Método de Barreira

O método de Barreira é um caso particular de uma família de algorítmos conhecidos como métodos de Penalidades, no qual um problema com restrições é transformado em uma sequência de problemas sem restrições por meio de uma penalização das restrições. A penalização pode ser externa ou interna.

No método de penalização externa, uma função penalidade é adicionada á função objetivo para toda violação de restrições. Esse método gera uma sequência de pontos infactíveis, cujo limite é uma soluçãq ótima do problema original [1]. Os métodos de penalização externa não serão abordados nessa dissertação.

No método de penalização interna, ou Método de Barreira, uma função barreira é adicionada à função objetivo evitando que os pontos gerados estejam fora da região factível ou próximos da fronteira . O método gera uma sequência de pontos factíveis cujo limite é a solução ótima do problema original. A essa função barreira associa-se um parâmetro t - chamado de parâmetro barreira - que determina o grau de interioridade da solução do subproblema [1].

Considere o problema não linear:

$$\begin{array}{ll} min\ f(x) & \\ s.a & \left\{ \begin{array}{ll} g_i(x) \geq 0 & i = 1, \ldots, m \\ x \in \Omega \subset \Re^n & \end{array} \right. \end{array} \tag{1}$$

$$\text{onde } f,\ g_i \in C^{(2)} \qquad i = 1, \ldots, m$$

Seja $S = \{x \in \Omega$ tal que $g_i(x) \geq 0 \qquad i = 1, \ldots, m\}$ e assuma que o interior de S é não vazio.

Aplicando-se a transformação barreira ao problema (1), obtemos o seguinte problema sem restrições:

$$min\ (f(x) + t\ P(x)). \tag{2}$$

onde $P(x)$ é uma função barreira que é contínua no interior do conjunto S e tende a infinito à medida que aproximamos da fronteira de S.

A função barreira é definida por:

$$P(x) = \sum_{i=1}^{m} p_i\ (g_i(x)). \tag{3}$$

onde $p_i(.)$ satisfaz:

(i) $p_i(y) \in C^{(2)}$ para $y \in int\ S$ [1].

(ii) Se $\lim_{k\to\infty} g_i(x^k) = 0$ então $\lim_{k\to\infty} p_i\ (g_i(x^k)) = \infty, \ \forall i$.

(iii) $\frac{\partial\, p_i\ (g_i(x))}{\partial\, g_i} < 0$ e $\frac{\partial^2\, p_i\ (g_i(x))}{\partial\, g_i^2} > 0$ para $x \in int\ S$.

(iv) $\frac{\partial^2\, p_i}{\partial\, g_i^2}$ é uma função monotônica decrescente de $g_i(x)$.

[1] int S é o interior do conjunto S

Podemos resolver (1) através de problemas do tipo (2) para t suficientemente pequeno, ou por uma sequência de problemas da forma (2) com valores decrescentes de t. Nesse último caso, o ponto ótimo de (2) para um determinado t_k será o ponto inicial para o problema (2) para t_{k+1} onde $t_{k+1} < t_k$.

Revisaremos alguns resultados que mostram que a solução do problema (2) tende para a solução do problema (1).

Considere o problema:

$$min\ f(x)\ ,\quad x \in \Omega$$

Teorema 1.1 *Se f é contínua e Ω é compacto então $\exists\, \overline{x} \in \Omega$ tal que $\overline{x}$ é minimizador global de f em Ω.*

(A demonstração encontra-se em [10]).

Teorema 1.2 *Seja f contínua em $\Omega \subset \Re^n$.*

$\{x \in \Omega$ tal que $f(x) \leq f(x^0)\}$ limitado $\Rightarrow f$ tem minimizador global em Ω.

<u>Demonstração</u>

Seja $A = \{x \in \Omega$ tal que $f(x) \leq f(x^0)\} = f^{-1}\,(-\infty, f(x^0)]$

Como $(-\infty, f(x^0)\,]$ é fechado em $\Re$ e f é contínua, temos que $f^{-1}\,(-\infty, f(x^0)]$ é fechado em Ω.

Fechado e limitado em $\Re^n$ é compacto em Ω.

Portanto, pelo **Teorema 1.1**, f tem minimizador global x^* em **A**. Pela definição de **A**, x^* é também minimizador global em Ω.

$\square$

Considere, agora, o problema:

$$P_1 \;:\; min f(x)$$
$$s.a \begin{cases} g_i(x) \geq 0 & i = 1, \ldots, m \\ x \in \Omega, & \Omega \text{ fechado} \end{cases}$$

onde $f,\ g_i \in C^{(2)}\ i = 1, \ldots, m$

e o problema penalizado

$$P_2 \;:\; min\ f_t(x) = min\ [f(x) - t \sum_{i=1}^{m} p_i\ (g_i(x))].$$
$$x \in \Omega'\ onde\ \Omega' = \{x \in \Omega\ tal\ que\ g_i(x) > 0\}$$

Seja $p(x) = -\sum_{i=1}^{m} p_i(g_i(x))$

São válidas as seguintes propriedades:

Prop. 1:- Se $t' < t$ então $f_t(x(t')) < f_t(x(t))$

Prop. 2:- Se $t' < t$ então $p(x(t')) \geq p(x(t))$

Prop. 3:- Se $t' < t$ então $f(x(t')) \leq f(x(t))$

A demonstração das propriedades descritas, encontram-se em [1].

O próximo resultado mostra-nos sob quais condições a solução ótima $x(t)$ do problema penalizado P_2 existe.

Teorema 1.3 *Se Ω é limitado então existe minimizador global de P_2.*

<u>Demonstração</u>

Seja $A = \{x \in \Omega$ tal que $f_t(x) \leq f_t(x^0)\ \}$. Observemos que $A \neq \emptyset$ pois $x^0 \in A$ e verifiquemos que A é compacto:

1. A é limitado pois $A \subset \Omega$, e Ω é limitado.

2. A é fechado

Então, consideremos uma sequência $(x^k) \subset A$ tal que $x^k \rightarrow \overline{x} \in \overline{\Omega'}$ (fechado).

Suponhamos que $\overline{x} \in \overline{\Omega'} - \Omega' \Rightarrow$ existe $g_i(\overline{x}) = 0$.

Logo $\lim_{k \rightarrow \infty} f_t(x^k) = \infty$ (definição de penalização interna).

absurdo

Portanto $\overline{x} \in \Omega'$

Agora, pela continuidade da f_t em Ω, temos:

$$\lim_{k \rightarrow \infty} x^k = \overline{x} \overset{contda\ f_t}{\Longrightarrow} \lim_{k \rightarrow \infty} f_t(x^k) = f_t(\overline{x}) \tag{4}$$

Como $x^k \in \mathbf{A}\ \forall\, k \in \mathcal{N}$.

$$f_t(x^k) \leq f_t(x^0) \qquad \forall\, k \in \mathcal{N} \tag{5}$$

De (4) e (5), temos $\lim_{k \rightarrow \infty} f_t(x^k) = f_t(\overline{x}) \leq f_t(x^0)$

Portanto $\overline{x} \in \mathbf{A}$

Assim, pelo Teorema 1.2, $f_t(x)$ tem minimizador global em Ω '.

Portanto, se Ω é limitado, $x(t)$ existe.

□

Teorema 1.4 *Seja* $t_k > 0$ *tal que* $\lim_{k \rightarrow \infty} t_k = 0$.

Seja $x^k = x(t_k)$ *minimizador global de* f_{t_k} *e* x^* *ponto de acumulação de* (x^k) .

Então x^* *é minimizador global de* f.

Demonstração

Sem perda de generalidade $\lim_{k\to\infty} x^k = x^* \in \Omega$ (fechado).

Ainda $g(x^k) > 0 \; \forall k$

$\Rightarrow \lim_{k\to\infty} g(x^k) = g(x^*) \geq 0.$

$$\Rightarrow x^* \text{ é ponto factível de } P_1.$$

Seja $\hat{x} \in \Omega$ e $g(\hat{x}) \geq 0$.

Vamos provar que $f(\hat{x}) \geq f(x^*)$.

Como f é contínua, temos $\lim_{k\to\infty} f(x^k) = f(x^*)$. Pela Prop. 3, temos que $(f(x^k))$ é não crescente, logo:

$$f(x^k) \geq f(x^*) \; \forall \, k \tag{6}$$

Suponhamos $f(\hat{x}) < f(x^*)$. Existe um $c > 0$ tal que:

$$f(\hat{x}) \leq f(x^*) - c$$

Seja

$$y \in \Omega \; tal \; que \begin{cases} g(y) > 0 \\ f(y) \leq f(x^*) - \frac{c}{2} \end{cases}$$

Para t_k suficientemente pequeno $t_k p(y) \leq \frac{c}{4}$.

Logo:

$$f_t(y) = f(y) + t_k p(y) \leq f(x^*) - \frac{c}{4}$$

se k é suficientemente grande.

Por (6) temos:

$$f(x^*) - \frac{c}{4} < f(x^k) < f(x^k) + tp(x^k) = f_t(x^k)$$

pois $p(x^k)$ é positiva pela definição de função barreira.

Como x^k é minimizador global de f_t temos o absurdo procurado.

Logo $f(\hat{x}) \geq f(x^*)$ e portanto

x^* é minimizador global de P_1.

$\square$

1.2 Função Barreira Logarítmica Aplicado a um Problema Linear

A função barreira que vamos trabalhar é:

$$P(x) = -\sum_{i=1}^{m} \log(g_i(x)) \tag{7}$$

que foi primeiramente discutida por *Frisch* em 1955.

Considere o problema linear:

$$\begin{array}{c} min\ b^T y \\ s.a \begin{cases} A\,y = c \\ y \geq 0 \end{cases} \end{array} \tag{8}$$

onde

$$\text{A é uma matriz } n \times m \text{ com } m \geq n$$
$$b,\ y \in \Re^m \text{ e } c \in \Re^n$$

Nosso objetivo é resolver o problema linear (8) usando uma estratégia não linear. Queremos reduzir o custo o mais rápido possível e caminhar o mais longe possível sem atingir a fronteira. O nosso receio, ao aproximarmos da fronteira, é diminuir muito variáveis que no final não tenderão a zero (variáveis básicas na solução ótima). Uma maneira de se evitar a fronteira é usar uma *função barreira* [9]. Iremos aplicar a transformação barreira diretamente ao problema (8).

A transformação barreira pode ser aplicada apenas para as restrições de desigualdade, assim o subproblema associado ao (8) trata as restrições de igualdade diretamente, obtendo:

$$min\ f_t(y) = min\ [b^T y - t \sum_{j=1}^{m} \log y_j]$$

$$s.a \{ A y = c \tag{9}$$

onde

$$t \text{ é o } parâmetro\ barreira$$

Uma maneira de resolver (9) é usar um método de descida.

Partindo de y_0 factível, as sucessivas iterações são obtidas por:

$$y^{k+1} = y^k + \lambda d \qquad\qquad \lambda > 0$$

onde o cálculo de λ (tamanho do passo) e d (direção de descida) devem assegurar que:

$$A\, y^{k+1} = c$$

e

$$f(y^{k+1}) < f(y^k)$$

Observe que para manter a factibilidade, a direção d deve estar num subespaço particular (o núcleo de A). Para evitar isso, vamos escrever uma alternativa baseada no problema linear[2]:

[2]Todo problema linear pode ser colocado na forma (10)

$$\begin{aligned} & min\ c^T x \\ & s.a\ \left\{ A^T x \leq b \right. \end{aligned} \tag{10}$$

onde

$$\mathbf{A} \text{ é uma matriz } \boldsymbol{n} \times \boldsymbol{m} \text{ com } \boldsymbol{m} \geq \boldsymbol{n}$$
$$\boldsymbol{x},\ \boldsymbol{c} \in \Re^n \text{ e } \boldsymbol{b} \in \Re^m$$

Aplicando a transformação barreira a (10), temos:

$$min f(x) = min\ [c^T x - t \sum_{j=1}^{m} \log(b_j - a_j^T x)]. \tag{11}$$

que é um problema sem restrições.

Existem muitos métodos para se resolver problemas do tipo (11). Dentre eles, destacam-se os métodos quase-Newton.

Capítulo 2

2 A Família Geral de Métodos quase-Newton

2.1 Métodos quase-Newton

A família de métodos quase-Newton introduzida por Martínez [16] resolve sistemas de equações algébricas não lineares.

Considere o problema:

$$min\ f(x) \qquad \text{onde } f : \Omega \subset \Re^n \longrightarrow \Re, \qquad \Omega \text{ aberto} \tag{12}$$

Uma condição necessária para encontrar a solução $\overline{x}$ de (12) é $\nabla f(\overline{x}) = 0$.

Dessa maneira, podemos encontrar o ponto ótimo de (12) resolvendo o seguinte sistema não linear:

$$\begin{aligned} \text{Encontrar} \quad & x \in \Omega \ tal\ que\ F(x) = 0 \\ \text{onde} \quad & F : \Omega \subset \Re^n \longrightarrow \Re^n \qquad F \in C^1(\Omega) \\ \text{Identificando} \quad & F(x) = \nabla f(x) \end{aligned} \tag{13}$$

O método mais popular para resolver (13) é o método de Newton. O método de Newton parte de um ponto x^0 arbitrário, $x^0 \in \Omega$, e calcula as sucessivas aproximações (x^k) , de acordo com a fórmula:

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

onde $J(x^k) = \nabla^2 f(x^k)$.

A cada iteração do método de Newton:

(a) a matriz $J(x^k)$ deve ser calculada

(b) o sistema linear $n \times n$

$$J(x^k)\, z = -F(x^k)$$

deve ser resolvido

Para evitar o cálculo do $J(x^k)$ em toda iteração, foram introduzidos os métodos quase-Newton [3], que são baseados na iteração :

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

onde B_{k+1} é obtida a partir de B_k usando procedimentos "baratos" que, em princípio não envolvem derivadas.

Alternativamente, B_{k+1}^{-1} pode ser obtida diretamente a partir de B_k^{-1} , ou uma fatoração adequada de B_{k+1} pode ser obtida a partir do mesmo tipo de fatoração de B_k . Existem muitos métodos que usam a "idéia quase-Newton". Alguns utilizados em nosso trabalho são: Greenstadt, DFP, BFGS, Correção de Posto 1 [17]-, e outros que serão vistos nesse capítulo e no próximo.

Martínez [16] introduziu uma família de métodos que envolve muitos métodos quase-Newton conhecidos. Os métodos dessa família tem convêrgencia superlinear. Vamos, agora, descrever essa família.

2.2 A Família Geral

Seja $F : \Omega \subset \Re^n \longrightarrow \Re^n$, Ω aberto e convexo e $F \in C^1(\Omega)$.

Suponhamos uma condição Lipschitz para J, isto é, existem $x^* \in \Omega$, $p, M > 0$ tais que, para todo $x \in \Omega$,

$$^3 \mid J(x) - J(x^*) \mid \leq M \mid x - x^* \mid^p \tag{14}$$

(14) implica que para todo x, $z \in \Omega$,

$$\mid F(z) - F(x) - J(x^*)(z - x) \mid \leq m \mid z - x \mid \sigma(x, z)^p \tag{15}$$

onde

$$\sigma(x, z) = max \{\mid x - x^* \mid , \mid z - x^* \mid\}$$

Suponhamos $F(x^*) = 0$ e $J(x^*)$ não singular.

Seja χ um espaço vetorial de dimensão finita, com um produto interno $\langle \, , \, \rangle$ cuja norma associada denotaremos por $\| \cdot \|$.

Definimos $\langle \, , \, \rangle_{xz}$ um outro produto interno cuja norma associada é $\| \cdot \|_{xz}$, para cada x, $z \in \Omega$.

Axioma 2.1 *Suponhamos que existem* q, $c_1 > 0$ *tais que, para todo* x, $z \in \Omega$, $E \in \chi$

$$\|E\|_{xz} \leq [1 + c_1 \, \sigma(x, z)^q] \, \|E\|$$

Axioma 2.2 *Suponhamos que existem* q, $c_1 > 0$ *tais que, para todo* x, $z \in \Omega$, $E \in \chi$,

$$\|E\| \leq [1 + c_1 \, \sigma(x, z)^q] \, \|E\|_{xz}$$

[3] $| \cdot |$ denota uma norma vetorial qualquer, e sua norma matricial subordinada

Esses dois Axiomas significam que as normas $\| \,.\, \|_{xz}$ ficam muito próximas de $\| \,.\, \|$, quando x e z são próximos de x^*.

Axioma 2.3 *Seja* $E_* \in \chi$, $\varphi : \Re^n \times \chi \longrightarrow \Re^{n \times n}$ *uma função contínua definida numa vizinhança de* (x^*, E_*) *tal que* $\varphi(x, E)$ *é não singular no seu domínio, e*

$$| \; I - \varphi(x^*, E_*)^{-1} \, J(x^*) \; | \leq r^* < 1.$$

Esse axioma nos diz que há um parâmetro "ideal" E_*, tal que $\varphi(x^*, E_*)$ é bastante próximo de $J(x^*)$.

O teorema (2.1), abaixo, mostra-nos que se E é suficientemente próximo de E_*, a aplicação *"Newtoniana"*

$$x \longrightarrow x - \varphi(x, E)^{-1} \, F(x)$$

aproxima o ponto x de x^* com uma taxa próxima de r^*.

Esse teorema, bem como sua demonstração, encontram-se em [16]

Teorema 2.1 *Seja* $r \in (r^*, 1)$. *Então, existem vizinhanças* Ω_1 *de* x^*, *e* $\mathcal{N}$, *de* E_*, *tais que, para todo* $x \in \Omega_1$, $E \in \mathcal{N}$ $| \; \varphi(x, E) \; |$, $| \; \varphi(x, E)^{-1} \; |$, $\|E\|$ *estão uniformemente limitados, e*

$$| \; x - \varphi(x, E)^{-1} \, F(x) - x^* \; | \leq r \; | \; x - x^* \; |.$$

Axioma 2.4 *Para cada par* $x, z \in \Omega$, *seja* $V = V(x, z)$ *uma variedade afim em* χ.

Suponhamos que, para todo $x, z \in \Omega$, *existe* $E \in V(x, z)$ *tal que:*

$$\|E - E_*\| \leq c_2 \, \sigma(x, z)^p$$

com $c_2, \, p > 0$.

Este axioma afirma que a distância entre cada variedade $V(x, z)$ e E_* é um $O(e)$.

2.3 Convergência Local Linear para os Métodos da Família Geral

Nessa secção, analisaremos as propriedades de convergência de algoritmos cuja forma geral é dada abaixo.

Algoritmo 2.1

Seja x^0 um ponto inicial, arbitrário, $E_0 \in \chi$.

Para todo $k = 0, 1, \ldots$ definimos:

$$x^{k+1} = x^k - \varphi(x^k, E_k)^{-1} F(x^k)$$

e, E_{k+1} a projeção ortogonal de E_k em $V(x^k, x^{k+1})$ relativa à norma $\| \cdot \|_{x^k,x^{k+1}}$ (chamaremos a norma $\|.\|_{x^k,x^{k+1}}$ daqui por diante de $\| \cdot \|_k$).

Os teoremas apresentados aqui, bem como suas respectivas demonstrações, encontram-se em [16].

Teorema 2.2 *Seja* $r_1 \in (r^*, 1)$. *Existem* $\epsilon = \epsilon(r_1)$, $\delta = \delta(r_1)$ *tais que, se* $| x^0 - x^* | \leq \epsilon$ e $\|E_0 - E_*\| \leq \delta$, *então a sequência gerada pelo Algorítmo 2.1 está bem definido e, para todo* $k = 0, 1, 2, \ldots$,

$$| x^{k+1} - x^* | \leq r_1 \; | x^k - x^* |$$

O teorema 2.2 nos mostra que, se x^0 e E_0 estão suficientemente próximo de x^* e E_* respectivamente, a sequência (x^k) está bem definida e converge linearmente para x^*.

Teorema 2.3 *Vamos supor que*

$$\lim_{k \to \infty} \frac{| [\varphi(x^{k+1}, E_{k+1}) - \varphi(x^*, E_*)] (x^{k+1} - x^k) |}{| x^{k+1} - x^k |} = 0. \tag{16}$$

Então

$$\lim\ sup \frac{| x^{k+1} - x^* |}{| x^k - x^* |} \leq r^* \tag{17}$$

Logo, a convergência é superlinear se $r^* = 0$.

Até aqui vimos a teoria da família de métodos quase-Newton. Nas secções seguintes analisaremos métodos específicos, verificando se eles são aplicáveis à teoria exposta. Muitos deles são métodos quase-Newton conhecidos. Para verificar se eles estão incluidos nessa família, devemos seguir os seguintes passos:

(a) Definição de χ, φ, E_*, $V(x, z)$, $\|\cdot\|$, $\|\cdot\|_{xz}$

(b) Verificação dos *Axiomas* 2.1, 2.2, 2.3 e 2.4 e de (16).

Com isso podemos afirmar que, se x^0 e E_0 estão suficientemente perto de x^* e E_*, respectivamente, a sequência gerada converge para x^* com a taxa ideal r^* (superlinear se $r^* = 0$).

2.4 Método de Greenstadt

Algoritmo 2.2

Sejam $x^0 \in \Re^n$, $H_0 \in \Re^{n \times n}$, H_0 simétrica não singular. Para $k = 0, 1, 2, \ldots$, calcular x^{k+1}, H_{k+1}, seguindo os passos:

Passo 1:- Obtenção de x^{k+1}

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

Passo 2:- Atualização de H_k

$$H_{k+1} = H_k + \frac{(s - H_k y)\, y^T + y\, (s - H_k y)^T}{y^T y} - \frac{(s - H_k y)^T\, y\, y\, y^T}{(y^T y)^2}$$

com $s = x^{k+1} - x^k, \quad y = F(x^{k+1}) - F(x^k)$

Passo 3:- Cálculo da direção d_k

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

Passo 4:- Otenção do novo ponto

$$x^{k+1} = x^k + d_k$$

Esse algoritmo se enquadra na teoria da família Geral.

De fato:

(a) Definimos

- $\chi = \{ H \in \Re^{n \times n}$ tal que $H = H^T \}$.
- $\varphi(x, H) = H^{-1}$ para todo $x \in \Omega$, $H \in \chi$.
- $E_* = [J(x^*)]^{-1}$.
- $V(x, z) = \{ H \in \chi$ tal que $H\, [F(z) - F(x)] = z - x \}$.
- $\| \cdot \| = \| \cdot \|_{xz} = \| \cdot \|_F$[4] para todo $x,\ z \in \Omega$.

(b) • *Axioma* 2.1 e *Axioma* 2.2 são triviais, pois $\| \cdot \| = \| \cdot \|_{xz}$ para todo $x,\ z$.

[4] $\|.\|_F$ é a norma de Frobenius

- *Axioma* 2.3 : Devemos mostrar que

$$| I - \varphi(x^*, E_*)^{-1} J(x^*) | \leq r^* < 1$$

Mas $\varphi(x^*, E_*) = E_*^{-1} = J(x^*)$, portanto segue o resultado com $r^* = 0$.

- *Axioma* 2.4: Seja $\tilde{J} = \int_0^1 J(x + t(z - x))dt$

$$\begin{aligned}
\tilde{J}(z - x) &= F(z) - F(x) \\
\|E_* - E\| &= \|J(x^*)^{-1} - \tilde{J}^{-1}\| \\
&\leq \|J(x^*) - \tilde{J}\| \; \|J(x^*)^{-1}\| \; \|\tilde{J}^{-1}\| \\
&\leq | J(x^*) - \int_0^1 J(x + t(z - x))dt \; | \; | \; [J(x^*)]^{-1} \; | \\
&\quad | \; [\int_0^1 J(x + t(z - x))dt]^{-1} \; | \\
&\leq m \, [\int_0^{-1} | J(x^*) - J(x + t(z - x)) | \; dt] \\
&\quad [\int_0^1 | \; J(x + t(z - x)) \; |^{-1} \; dt] \\
&\leq m \, [\int_0^1 m' \; | x + t(z - x) - x^* |^p \, dt] \, [\int_0^1 | \; J(x + t(z - x)) \; |^{-1} \, dt] \\
&\leq mm' \, max \, | x + t(z - x) - x^* |^p \int_0^1 | \; J(x + t(z - x)) \; |^{-1} \, dt \\
&\leq m \, \sigma(x, z)^p \int_0^1 | \; J(x + t(z - x)) \; |^{-1} \, dt \\
&\leq m \, \sigma(x, z)^p
\end{aligned}$$

para $\int_0^1 | \; J(x + t(z - x)) \; |^{-1} \, dt$ limitado.

- Vamos verificar (16)

Pela definição de $V(x, z)$, temos que $\varphi(x^{k+1}, E_{k+1})$ verifica:

$$\varphi(x^{k+1}, E_{k+1}) \, (x^{k+1} - x^k) = F(x^{k+1}) - F(x^k)^5 \qquad (18)$$

[5](18) é a equação fundamental dos **quase-Newton** (também chamado de Equação Secante)

Então

$$| \; [\varphi(x^{k+1}, E_{k+1}) \; - \; \varphi(x^*, E_*)] \, (x^{k+1} \; - \; x^k) \; | =$$

$$| \; F(x^{k+1}) \; - \; F(x^k) \; - \; \varphi(x^*, E_*) \, (x^{k+1} \; - \; x^k) \; | =$$

$$| \; F(x^{k+1}) - F(x^k) - J(x^*) \, (x^{k+1} - x^k) \; | \leq \; M \; | \; x^{k+1} - x^k \; | \; \sigma(x^{k+1}, x^k)^p$$

onde $\sigma(x^{k+1}, x^k) \; = \; max\{| \; x^{k+1} \; - \; x^* \; |, | \; x^k \; - \; x^* \; |\}$

Assim

$$\lim_{k \to \infty} \frac{| \; [\varphi(x^{k+1}, E_{k+1}) \; - \; \varphi(x^*, E_*)] \, (x^{k+1} \; - \; x^k) \; |}{| \; x^{k+1} \; - \; x^k \; |} \; \leq$$

$$\lim_{k \to \infty} \frac{M \; | \; x^{k+1} \; - \; x^k \; | \; \sigma(x^{k+1}, x^k)^p}{| \; x^{k+1} \; - \; x^k \; |}$$

$$\lim_{k \to \infty} M \; \sigma(x^{k+1}, x^k)^p \; = \; 0$$

Provamos assim que:

Existem ϵ, $\delta \; > \; 0$ tais que, se $| \; x^0 - x^* \; | \leq \; \epsilon$, $| \; H_0 - J(x^*)^{-1} \; | \leq \; \delta$, a sequência (x^k) gerada pelo *Algoritmo Greendstadt* converge superlinearmente para x^*.

No restante deste Capítulo, apresentaremos métodos onde as normas $\| \cdot \|_{xz}$ não coincidem com $\| \cdot \|$.

2.5 Método DFP (Davidon - Fletcher - Powell)

Algoritmo 2.3

Seja $x^0 \in \Re^n$, $H_0 \in \Re^{n \times n}$, H_0 não singular e simétrica. Para $k = 0, 1, 2, \ldots$, calcular x^{k+1}, H_{k+1}, , mediante os passos seguintes:

Passo 1:- Cálculo x^{k+1}

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

Passo 2:- Atualização de H_k

$$H_{k+1} = H_k + \frac{s\, s^T}{s^T\, y} - \frac{H\, y\, y^T\, H}{y^T\, H\, y} \tag{19}$$

onde $s = x^{k+1} - x^k$ e $y = F(x^{k+1}) - F(x^k)$

Passo 3:- Cálculo da direção

$$d_k = -\, H_{k+1}\, F(x^k)$$

Passo 4:- $x^{k+1} = x^k + d_k$

Supomos aqui que $J(x)$ é simétrico para todo $x \in \Omega$ e que $J(x^*)$ é simétrico e positivo definido.

Então podemos considerar $J(x)$ semi-positivo definido em uma vizinhança de x^*. Sem perda de generalidade, suponhamos que $J(x)$ é semi-positivo definido para todo $x \in \Omega$. Como Ω é convexo, $J(x + t(z - x))$ é semi-positivo definido para todo $x, z \in \Omega, 0 \leq t \leq 1$, e também:

$$\bar{J}(x, z) = \int_0^1 J(x + t(z - x))dt$$

é semi positivo definido para todo $x, z \in \Omega$. Portanto, para todo $x, z \in \Omega$ está definido uma matriz triangular inferior $L(x,z)$ tal que $L(x,z)\, L(x,z)^T = \tilde{J}(x,z)$ (fatoração de Cholesky de $\tilde{J}(x,z)$).

Analisando o Algoritmo para obter a fatoração de Cholesky, vemos que cada coeficiente de $L(x,z)$ é uma função continuamente diferenciável dos coeficientes de $\tilde{J}(x,z)$. Portanto, restringindo um pouco mais Ω , se for necessário, obtemos pelo **Teorema do Valor Médio**, que existe $b_1 > 0$ tal que:

$$| L(x,z) - L(x^*,x^*) | \leq b_1 \; | \tilde{J}(x,z) - \tilde{J}(x^*,x^*) |$$

Por (14), existe $b_2 > 0$ tal que:

$$| L(x,z) - L(x^*,x^*) | \leq b_2\, \sigma(x,z)^p$$

e pelo lema de Banach:

$$| L^{-1}(x,z) - L^{-1}(x^*,x^*) | \leq b_3\, \sigma(x,z)^p \tag{20}$$

para certo $b_3 > 0$, e uma desigualdade similar vale para L^T.

Vamos, agora, provar a convergência superlinear do Algoritmo.

(a) Definimos

- $\chi = \{H \in \Re^{n \times n}$ tal que $H = H^T\}$
- $\varphi(x,H) = H^{-1}$ para todo $x \in \Omega,\ H \in \chi$
- $E_* = J(x^*)^{-1}$
- $V(x,z) = \{H \in \chi$ tal que $H\,[F(z) - F(x)] = z - x\}$
- $\|H\| = \|L^{-1}(x^*,x^*)\; H\; L^{-T}(x^*,x^*)\|_F$
- $\|H\|_{xz} = \|L^{-1}(x,z)\; H\; L^{-T}(x,z)\|_F$ para todo $x,\ z \in \Omega$

(b) Verificação dos Axiomas e da relação (16)

- *Axioma* 2.1: Vamos usar a seguinte notação:

$$L_*^{-1} = L^{-1}(x^*, x^*) \quad \text{e} \quad L_*^{-T} = L^{-T}(x^*, x^*)$$

$$\begin{aligned}
\|H\|_{xz} &= \|L^{-1}(x,z)\, H\, L^{-T}(x,z)\|_F \\
&= \|[L^{-1}(x,z) - L_*^{-1} + L_*^{-1}]\, H\, [L^{-T}(x,z) - L_*^{-T} + L_*^{-T}]\|_F \\
&\leq \|[L^{-1}(x,z) - L_*^{-1}]\, H\, [L^{-T}(x,z) - L_*^{-T}]\|_F + \\
&+ \|[L^{-1}(x,z) - L_*^{-1}]\, H\, L_*^{-1}\|_F + \\
&+ \|L_*^{-1}\, H\, [L^{-T}(x,z) - L_*^{-T}]\|_F + \|L_*^{-1}\, H\, L_*^{-1}\|_F \\
&\leq \|H\| + b_2 \{\, |\, [L^{-1}(x,z) - L_*^{-1}]\, H\, [L^{-T}(x,z) - L_*^{-T}]\, | + \\
&+ |\, [L^{-1}(x,z) - L_*^{-1}]\, H\, L_*^{-T}\, | + \\
&+ |\, L_*^{-1}\, H\, [L^{-T}(x,z) - L_*^{-T}]\, |\, \}
\end{aligned}$$

pela equivalência de norma em dimensão finita.

Portanto:

$$\begin{aligned}
\|H\|_{xz} \leq\ & \|H\| + b_2\ |\, H\, |\ \{|\, L^{-1}(x,z) - L_*^{-1}\, |\, |\, L^{-T}(x,z) - L_*^{-T}\, | + \\
& + |\, L^{-1}(x,z) - L_*^{-1}\, |\, |\, L_*^{-T}\, | + \\
& + |\, L_*^{-1}\, |\, |\, L^{-T}(x,z) - L_*^{-T}\, |\} \\
\leq\ & \|H\| + b_2\ |\, H\, |\ \{\, b_4^2\, \sigma(x,z)^{2p} + b_5\, |\, L_*^{-T}\, |\ \sigma(x,z)^p + \\
& + b_6\, |\, L_*^{-1}\, |\ \sigma(x,z)^p\}
\end{aligned}$$

por (20).

Seja $b_3 = max\{\, b_4,\ b_5,\ b_6\}$

$$\begin{aligned}
\|H\|_{xz} \leq\ & \|H\| + b_2 b_3\ |\, H\, |\ \{b_3\, \sigma(x,z)^{2p} + |\, L_*^{-T}\, |\ \sigma(x,z)^p + \\
& + |\, L_*^{-1}\, |\ \sigma(x,z)^p\}
\end{aligned}$$

Pela equivalência de normas em dimensão finita, existe $b_1 > 0$ tal que $| H | \leq b_1 \|H\|$ para todo $H \in \chi$, logo:

$$\|H\|_{xz} \leq \|H\| \{1+b_1 b_2 b_3 \sigma(x,z)^p [b_3\sigma(x,z)^p + | L_*^{-1} | + | L_*^{-T} |] \}$$

Esta desigualdade implica o Axioma 1, considrando Ω limitado.

- *Axioma* 2.2:

$$\begin{aligned}
\|H\| &= \|L_*^{-1} H L_*^{-T}\|_F = \\
&= \|[L_*^{-1} - L^{-1}(x,z) + L^{-1}(x,z)] H [L_*^{-T} - L^{-T}(x,z) + L^{-T}(x,z)]\|_F \\
&\leq \|[L_*^{-1} - L^{-1}(x,z)] H [L_*^{-T} - L^{-T}(x,z)]\|_F + \\
&\quad \|[L_*^{-1} - L^{-1}(x,z)] H L^{-T}(x,z)\|_F + \\
&\quad \|L^{-1}(x,z) H [L_*^{-T} - L^{-T}(x,z)]\|_F + \\
&\quad \|L^{-1}(x,z) H L^{-T}(x,z)\|_F \\
&\leq \|H\|_{xz} + b_2 \{| [L_*^{-1} - L^{-1}(x,z)] H [L_*^{-T} - L^{-T}(x,z)] | \\
&\qquad + | [L_*^{-1} - L^{-1}(x,z)] H L^{-T}(x,z) | \\
&\qquad + | L^{-1}(x,z) H [L_*^{-T} - L^{-T}(x,z)] | \}
\end{aligned}$$

pela equivalência de normas em dimensão finita.

Portanto:

$$\begin{aligned}
\|H\| &\leq \|H\|_{xz} + b_2 | H | \{| L_*^{-1} - L^{-1}(x,z) | | L_*^{-T} - L^{-T}(x,z) | + \\
&\qquad + | L_*^{-1} - L^{-1}(x,z) | | L^{-T}(x,z) | + \\
&\qquad + | L^{-1}(x,z) | | L_*^{-T} - L^{-T}(x,z) | \} \\
&\leq \|H\|_{xz} + b_2 | H | \{b_3^2 \sigma(x,z)^{2p} + \\
&\qquad + b_3 | L^{-T}(x,z) | \sigma(x,z)^p + \\
&\qquad + b_3 | L^{-1}(x,z) | \sigma(x,z)^p\}
\end{aligned}$$

por (20).

Pela equivalência de normas em dimensão finita, existe $b_1 > 0$ tal que $| H | \leq b_1 \|H\|_{xz}$ para todo $H \in \chi$, logo:

$$\|H\| \leq \|H\|_{xz} + b_2 b_1 \|H\|_{xz} \{b_3^2 \sigma(x,z)^{2p} + b_3 | L^{-T}(x,z) | \sigma(x,z)^p + \\ + b_3 | L^{-1}(x,z) | \sigma(x,z)^p \}$$

$$\|H\| \leq \|H\|_{xz} \{1 + b_1 b_2 b_3 \sigma(x,z)^p [b_3 \sigma(x,z) + | L^{-1}(x,z) | + \\ | L^{-T}(x,z) |] \}$$

Esta desigualdade implica o Axioma 2, considerando Ω limitado.

- ***Axioma*** 2.3: $\varphi(x^*, E_*) = (E_*)^{-1} = J(x^*)$, portanto o Axioma 2.3 se verifica com $r^* = 0$.
- ***Axioma*** 2.4: A matriz simétrica $\tilde{J} = \int_0^1 J(x + t(z - x))dt$ verifica:

$$\tilde{J}(z - x) = F(z) - F(x)$$

e satisfaz $| \tilde{J} - J(x^*) | \leq \sigma(x,z)^p$

Portanto, o Axioma 2.4 se verifica restringindo suficientemente Ω e usando o lema de Banach:

$$\|B^{-1} - A^{-1}\| \leq \|B - A\| \|B^{-1}\| \|A^{-1}\|$$

Segue análogo ao Algoritmo de Greenstadt

- Relação (16) segue análogo ao Algoritmo de Greenstadt.

Provamos assim que:

Existem $\epsilon, \delta > 0$ tais que, se $| x^0 - x^* | \leq \epsilon, | H_0 - J(x^*)^{-1} | \leq \delta$, a sequência definida pelo *Algoritmo DFP* converge superlinearmente para x^*.

2.6 Método BFGS (Broyden, Fletcher, Goldfarb e Shanno)

Algoritmo 2.4

Seja $x^0 \in \Re^n, H_0 \in \Re^{n\times n}, H_0$ não singular simétrica.

Para $k = 0, 1, 2, \ldots$, calcular x^{k+1}, H_{k+1}, , mediante os passos seguintes:

Passo 1:- Cálculo x^{k+1}

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

Passo 2:- Atualização de H_k

$$H_{k+1} = H_k + [1 + \frac{y^T H_k y}{s^T y}] \frac{s s^T}{s^T y} - \frac{s y^T H_k + H_k y s^T}{s^T y}$$

onde $s = x^{k+1} - x^k$ e $y = F(x^{k+1}) - F(x^k)$

Passo 3:- Cálculo da direção

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

Passo 4:- Cálculo do novo ponto:

$$x^{k+1} = x^k + d_k$$

Observação: Todos os comentários feitos em relação ao $J(x)$ e $\bar{J}$ no método DFP, valem também aqui.

Provemos então a convergência superlinear do Algoritmo:

(a) Definimos

- $\chi = \{H \in \Re^{n\times n}$ tal que $H = H^T\}$.

- $\varphi(x, H) = H^{-1}$ para todo $x \in \Omega$, $H \in \chi$.
- $E_* = J(x^*)^{-1}$.
- $V(x, z) = \{H \in \chi$ tal que $H\,[F(z) - F(x)] = z - x\}$.
- $\|H\| = \|L^{-1}(x^*, x^*)\; H\; L^{-T}(x^*, x^*)\|_F$.
- $\|H\|_{xz} = \|L^{-1}(x, z)\; H\; L^{-T}(x, z)\|_F$ para todo $x, z \in \Omega$.

(b) Verificação dos Axiomas e da relação (16) seguem análogos ao Algoritmo DFP.

Portanto, existem $\epsilon,\ \delta > 0$ tais que, se $\mid x^0 - x^* \mid \leq \epsilon$, $\mid H_0 - J(x^*)^{-1} \mid \leq \delta$, a sequência definida pelo *Algoritmo BFGS* converge superlinearmente para x^*.

2.7 Método quase-Newton Estruturado

Nessa seção vamos usar uma estratégia diferente da que temos usado até aqui. Não vamos apresentar o algoritmo diretamente e sim escrever sobre o método primeiramente.

O nosso objetivo é resolver o problema (11). A função f é diferenciável e suas derivadas são dadas por:

(1) $F(x) = \nabla f(x) = c + t\, A\, (\frac{1}{b - A^T x})$ com a convenção

$$\frac{1}{b - A^T x} = \begin{pmatrix} \frac{1}{(b - A^T x)_1} \\ \vdots \\ \frac{1}{(b - A^T x)_m} \end{pmatrix}$$

onde $(b - A^T x)_i$ é a i-ésima linha de $(b - A^T x)$.

(2) $J(x) = \nabla^2 f(x) = t\ A\ D(x)\ A^T$ onde

$$D(x) = \begin{bmatrix} \frac{1}{(b-A^T x)_1^2} & 0 & \dots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & \frac{1}{(b-A^T x)_m^2} \end{bmatrix}$$

Para aproveitar a estrutura $\nabla^2 f$, vamos aproximar $A\ D(x^{k+1})\ A^T$ por $A\ H_k\ A^T$ onde H_k é uma matriz simétrica e positiva definida, como $D(x^{k+1})$.

Pela fórmula DFP (19), H_{k+1} é positiva definida se H_k é, sempre que $s^T y > 0$. Devemos, então escolher s e y de maneira que a propriedade $s^T y > 0$ seja garantida. Para manter a equação secante, fazemos:

$$s = A^T(x^{k+1} - x^k) \quad \text{e} \quad y = D(x^k)s$$

Com essas idéias, Martinez [16] definiu o algoritmo:

Algoritmo 2.5

Seja $x_0 \in \Omega$,

$$H_0 = t \begin{bmatrix} \frac{1}{(b-A^T x)_1^2} & 0 & \dots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & \frac{1}{(b-A^T x)_m^2} \end{bmatrix}$$

Para $k = 1, 2, \dots$, calcular x^{k+1}, H_{k+1}, pelos passos:

Passo 1:- $x^{k+1} = x^k - (AH_kA^T)^{-1}\ F(x^k)$

Passo 2:- $s = A^T(x^{k+1} - x^k)$, $y = D(x^k)s$

$$H_{k+1} = H_k + \frac{yy^T}{s^T y} - \frac{H_k s s^T H_k}{s^T H_k s} \tag{21}$$

Passo 3:- Cálculo da direção d_k

$$d_k = -(A\ H_k\ A^T)^{-1}\ F(x^k)$$

Passo 4:- $x^{k+1} = x^k + d_k$

Vejamos que o algoritmo supracitado é um método prático.

De fato, da fórmula (21) acima vem:

$$AH_{k+1}A^T = AH_kA^T + \frac{Ayy^TA^T}{s^Ty} - \frac{AH_kss^TH_kA^T}{s^TH_ks} \tag{22}$$

Portanto, a fatoração de Cholesky de $A\ H_{k+1}\ A^T$ pode ser obtida a partir da fatoração de Cholesky de $A\ H_k\ A^T$.

Fazendo $B_k = A\ H_k\ A^T$ podemos escrever a equação acima como:

$$B_{k+1} = B_k + \frac{Ay(Ay)^T}{s^Ty} - \frac{(AH_ks)(AH_ks)^T}{s^TH_ks}.$$

Esse Algoritmo está incluído na família de Métodos quase-Newton.

(a) Definimos

$$\chi = \Re^{n\times n}$$

$$\varphi(x,E) = A\ E^{-1}\ A^T$$

$$E_* = \frac{1}{t}\begin{bmatrix} \frac{1}{(b-A^Tx)_1^2} & 0 & \dots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & \frac{1}{(b-A^Tx)_m^2} \end{bmatrix}$$

$$V(x,z) = \{E \in \chi\ tal\ que\ E\ simétrica,\ Ey = s\}$$

$$\|E\| = \|D^{1/2}(x^*)\ E\ D^{1/2}(x^*)\|_F\ para\ todo\ E \in \chi$$

$$\|E\|_{xz} = \|D^{1/2}(x)\ E\ D^{1/2}(x)\|_F\ para\ todo\ x,z \in \Omega\ \mathrm{e}\ E \in \chi$$

(b) • *Axioma* 2.1:

$$
\begin{aligned}
\|E\|_{xz} &= \|D^{1/2}(x)\ E\ \ D^{1/2}\|_F \\
&= \|\ [D^{1/2}(x) - D^{1/2}(x^*) + D^{1/2}(x^*)]\ E\ [D^{1/2}(x) - D^{1/2}(x^*) + D^{1/2}(x^*)]\ \|_F \\
&\leq \|\ [D^{1/2}(x) - D^{1/2}(x^*)]\ E\ [D^{1/2}(x) - D^{1/2}(x^*)]\ \|_F\ + \\
&\quad + \|\ [D^{1/2}(x) - D^{1/2}(x^*)]\ E\ D^{1/2}(x^*)\ \|_F\ + \\
&\quad \|D^{1/2}(x^*)\ E\ [D^{1/2}(x) - D^{1/2}(x^*)]\ \|_F\ +\ \|D^{1/2}(x^*)\ E\ D^{1/2}(x^*)\ \|_F \\
&\leq \|E\|\ +\ b_1\ \{|\ \ [D^{1/2}(x) - D^{1/2}(x^*)]\ E\ [D^{1/2}(x) - D^{1/2}(x^*)]\ \ |\ + \\
&\quad +\ |\ \ [D^{1/2}(x) - D^{1/2}(x^*)]\ E\ D^{1/2}(x^*)\ \ |\ + \\
&\quad +\ |\ \ D^{1/2}(x)\ E\ [D^{1/2}(x) - D^{1/2}(x^*)]\ \ |\}
\end{aligned}
$$

Pela equivalência de normas em dimensaão finita.

$$
\begin{aligned}
\|E\|_{xz} &\leq \|E\|\ +\ b_1\ |\ E\ |\ \ \{|\ D^{1/2}(x) - D^{1/2}(x^*)\ |\ |\ D^{1/2}(x) - D^{1/2}(x^*)\ | \\
&\quad +\ |\ D^{1/2}(x) - D^{1/2}(x^*)\ |\ |\ D^{1/2}(x^*)\ |\ + \\
&\quad +\ |\ D^{1/2}(x)\ |\ |\ D^{1/2}(x) - D^{1/2}(x^*)\ |\}
\end{aligned}
$$

Pela definição de D e fazendo análogo ao que foi feito para o Algoritmo **DFP**, temos:

$$\|E\|_{xz}\ \leq\ \|E\|\ \{1\ +\ b'\ \sigma(x,z)^q\}$$

• *Axioma* 2.2: O procedimento é análogo ao que foi feito para o Axioma 2.1.

• *Axioma* 2.3:

$$\varphi(x,E)\ =\ A\ E^{-1}\ A^T\ \Rightarrow\ \varphi(x^*,E_*)\ =\ A\ E_*^{-1}\ A^T$$

onde $E_*\ =\ H_k^{-1}$. Portanto

$$|\ I\ -\ \varphi(x^*,E_*)^{-1}\ J(x^*)\ | = 0\ <\ 1$$

isto é, o Axioma 2.3 é satisfeito com $r^*\ =\ 0$.

- *Axioma* 2.4: $V(x,z) = \{E \in X$ tal que E simétrica, $E\,y = s\}$. Portanto :

$$\tilde{E} = \frac{1}{t}\begin{bmatrix} (b-A^Tx)_1^2 & 0 & \dots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & (b-A^Tx)_m^2 \end{bmatrix}$$

  pertencente a $V(x,z)$ e o Axioma 2.4 segue naturalmente.

- Relação (16):- É uma consequência da definição de s e y.

Capítulo 3

3 Outros Métodos quase-Newton

Esse capítulo está dedicado a métodos quase-Newton que não estão incluídos na Família de Métodos quase-Newton descrita no capítulo anterior.

3.1 Método Correção de Posto 1

Algoritmo 3.1

Seja $x^0 \in \Re^n$, $H_0 \in \Re^{n \times n}$, H_0 não singular simétrica. Para $k = 0, 1, 2, \ldots$, calcular x^{k+1}, H_{k+1} , mediante os passos seguintes:

Passo 1:- Cálculo x^{k+1}

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

Passo 2:- Atualização de H_k

$$H_{k+1} = H_k + \frac{(s - H_k y)(s - H_k y)^T}{(s - H_k y)^T y}$$

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

Passo 3:- Cálculo da direção d_k

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

Passo 4:- Novo ponto

$$x^{k+1} = x^k + d_k$$

Não se enquadra na família de métodos quase-Newton.

3.2 Método quase-Newton Diagonal

Seja $f : \Re^n \longrightarrow \Re$, $f \in C^2$, q um inteiro positivo.

Algoritmo 3.2

Seja x^0 ponto inicial arbitrário.

Sejam m_1, M_1 um número pequeno e grande, respectivamente.

Passo 1:- Se $\nabla f(x^k) \cong 0$, parar

Passo 2:- 1. Se $k \equiv 0 \ (mod\ q)$ proceder:

Calcular a fatoração de Cholesky de $\nabla^2 f(x^k)$.

Se o processo "fracassa" pelo meio, acrescentar a diagonal $\nabla^2 f(x^k)$:

Encontre um $\mu > 0$ tal que $(\nabla^2 f(x^k) + \mu I) > 0$, μ suficientemente pequeno. Encontre a fatoração de Cholesky dessa Hessiana modificada.

Como resultado, teremos LDL^T, $D > 0$.

Chamemos $B_k = LDL^T$

2. Se $K \not\equiv 0 \ (mod\ q)$ proceder:

Seja $s = x^{k+1} - x^k$ e $y = \nabla f(x^{k+1}) - \nabla f(x^k)$

Calcular D_{k+1} da maneira que se descreve em [14]. Deixar a mesma L.

Normalizar D_{k+1} da seguinte maneira:

- Se $d_i > M_1$, $d_i = M_1$
- Se $d_i < m_1$, $d_i = m_1$

Chamamos $B_k = LDL^T$

Passo 3:- Resolver $B_k d = -\nabla f(x^k)$

(Ou seja, $LDL^T\, d = -\nabla f(x^k)$)

Passo 4:- Calcular λ_k usando o bem conhecido "*backtracking*" [3]

1. $\lambda = 1$
2. Se $f(x^k + \lambda d) > f(x^k) + 10^{-4}\, \lambda \langle \nabla f(x^k), d \rangle$ vá para 4.
3. $\lambda_k = \lambda$, vá para o Passo 5
4. Calcular um novo $\lambda \in [0.1\lambda, 0.9\lambda]$ por exemplo "*Interpolação Cúbica*".

Passo 5:- Novo ponto

$$x^{k+1} = x^k + \lambda_k\, d_k$$

3.2.1 Resultados de Convergência

Os teoremas, bem como suas demonstrações, se encontram em [14].

Teorema 3.1 *Todo ponto limite gerado pelo Algoritmo 3.2 é estacionário.*

Teorema 3.2 *Se* x^* *é minimizador local isolado, existe uma vizinhança* N *de* x^* *tal que se* $x^k \in \mathcal{N}$ *, então* $x^k \longrightarrow x^*$.

Teorema 3.3 *O método tem convergência linear ou superlinear se o método for com recomeços.*

Capítulo 4

4 Globalização para minimização sem restrições

4.1 Conceitos Fundamentais

Considere o problema sem restrições:

$$min\ f(x), \qquad \text{onde } f : \Omega \subset \Re^n \rightarrow \Re,\ \Omega \text{ aberto}$$

A idéia básica para encontrarmos o minimizador da função f é:

Dado um ponto x^k , escolhemos uma direção d a partir de x^k e encontramos um novo ponto x^{k+1} nessa direção de tal modo que:

$$f(x^{k+1}) < f(x^k)$$

A essa direção chamamos *direção de descida*. Formalmente podemos dizer que d é uma direção de descida se:

$$f(x + \lambda d) < f(x) \quad \text{para } \lambda \in (0, \epsilon)$$

Para isso é suficiente que $\langle \nabla f(x), d \rangle < 0$.

Do cálculo conhecemos o seguinte teorema:

Teorema 4.1 *Seja* $\varphi : \Re \longrightarrow \Re$, $\varphi'(0) < 0$ *e* $\theta \in (0,1)$.
Então existe λ_0 *tal que:*

$$\varphi(\lambda) < \varphi(0) + \theta\,\lambda\,\varphi'(0), \qquad \forall\,\lambda \in (0,\lambda_0)$$

Demonstração:-

Pela definição de derivada:

$$\lim_{\lambda\to 0} \frac{\varphi(\lambda) - \varphi(0)}{\lambda} = \varphi'(0)$$

Então:

$$\lim_{\lambda\to 0} \frac{\frac{\varphi(\lambda) - \varphi(0)}{\lambda}}{\varphi'(0)} = 1$$

Dado $\theta < 1$, existe λ_0 tal que para $\lambda < \lambda_0$, temos:

$$\left|\frac{\varphi(\lambda) - \varphi(0)}{\lambda\,\varphi'(0)} - 1\right| < \theta$$

$$\frac{\varphi(\lambda) - \varphi(0) - \lambda\varphi'(0)}{\lambda\varphi'(0)} < \theta$$

$$\varphi(\lambda) - \varphi(0) - \lambda\varphi'(0) < \theta\lambda\varphi'(0)$$

$$\varphi(\lambda) < \varphi(0) + (1+\theta)\lambda\varphi'(0)$$

Como $\varphi'(0) < 0$, temos:

$$\varphi(\lambda) < \varphi(0) + \theta\,\lambda\,\varphi'(0)$$

□

Com isso, podemos demonstrar o **Corolário de Armijo** que nos dá uma condição de descida suficiente.

Corolário:

Se $\langle \nabla f(x), d \rangle < 0$ e $0 < \theta < 1$, então existe $\lambda_0 > 0$ tal que:

$$f(x + \lambda d) < f(x) + \theta\lambda\langle\nabla f(x), d\rangle \ \textit{para todo} \ \lambda \in (0, \lambda_0)$$

<u>Demonstração</u> :-

Definimos $\varphi(\lambda) = f(x + \lambda d)$

Então usando a *Regra da Cadeia*, temos:

$$\varphi'(\lambda) = \langle\nabla f(x + \lambda d), d\rangle \Rightarrow \varphi'(0) = \langle\nabla f(x), d\rangle < 0$$

Pelo teorema 4.1, temos:

Para $\lambda \in (0, \lambda_0)$,

$$\varphi(\lambda) < \varphi(0) + \theta \lambda \varphi'(0) \Rightarrow$$

$$\Rightarrow f(x + \lambda d) < f(x) + \theta \lambda \langle \nabla f(x), d \rangle \ \textit{para todo} \ \lambda \in (0, \lambda_0)$$

□

A condição de Armijo, apesar de dar-nos uma condição de descida suficiente, não garante boas propriedades de convergência global. Temos que evitar passos muito pequenos em relação ao decrescimento inicial da função f e ângulo formado pelo ∇f (gradiente da função f) ë d (direção) arbitrariamente próximo de 90^o.

Assim podemos escrever um modelo de algoritmo para convergência global.

Algoritmo 4.1

Dados

$$\begin{gathered} x^k \\ \beta\ ,\ \beta > 0 \\ \theta_1\ ,\ 0 < \theta_1 < 1 \\ \theta_2\ ,\ 0 < \theta_2 < 1 \end{gathered}$$

1. Se $\nabla f(x^k) \simeq 0$ parar
2. Seja d_k tal que:
   - $[\,x^k, x^k + d_k\,] \subset \Omega$
   - $\| d_k \| \geq \beta \| \nabla f(x^k) \|$
   - $\nabla f(x^k)^T d_k \leq -\theta_2 \|\nabla f(x^k)\| \, \|d_k\|$
3. Fazer $x^{k+1} = x^k + \lambda_k d_k$ onde λ_k é obtido pelo Algoritmo:
   - (i) $\lambda = 1$.
   - (ii) Se λ satisfaz *Armijo* com θ_1 tomar $\lambda_k = \lambda$ e vá para (1).
   - (iii) Se λ não satisfaz *Armijo* com θ_1, escolher um novo λ no intervalo $[\sigma_1 \lambda, \sigma_2 \lambda\,]$ e repetimos o passo (ii).

Com esses conceitos vamos escrever o teorema de convergência global.

Teorema 4.2 (Convergência Global) *Se $x^* \in \Omega$ é um ponto limite da sequência (x^k) , gerada pelo Algoritmo 4.1, então $\nabla f(x^*) = 0$.*

<u>Demonstração</u>

Suponhamos que exista um ponto limite x^* de $\{x^k\}$, $k \in \mathcal{N}$, isto é, $\exists\, k_1 \subset \mathcal{N}$ tal que $\lim_{k \in k_1} x^k = x^*$ com $\nabla f(x^*) \neq 0$.

Vamos provar que:

$$\lim_{k \in k_1} (x^{k+1} - x^k) = 0$$

Por Armijo:

$$\begin{aligned} f(x^{k+1}) &\leq f(x^k) + \alpha \lambda d_k^T \nabla f(x^k) = \\ &= f(x^k) + \alpha (x^{k+1} - x^k)^T \nabla f(x^k) \end{aligned} \tag{23}$$

Mas

$$\begin{aligned} d_k^T \nabla f(x^k) &\leq -\theta \| \nabla f(x^k) \| \| d_k \| \Rightarrow \\ \lambda d_k^T \nabla f(x^k) &\leq -\theta \| \nabla f(x^k) \| \|\lambda d_k\| \Rightarrow \\ (x^{k+1} - x^k)^T \nabla f(x^k) &\leq -\theta \| \nabla f(x^k) \| \|x^{k+1} - x^k\| \end{aligned} \tag{24}$$

Substituindo (24) em (23) , temos:

$$f(x^{k+1}) \leq f(x^k) + \alpha (-\theta \|\nabla f(x^k)\| \|x^{k+1} - x^k\|) \Rightarrow$$

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

Como $(f(x^k))$ é uma sequência decrescente de números reais, $\lim_{k \in \mathcal{N}} f(x^k)$ existe [finito ou ($-\infty$)], isto é $\lim_{k \in \mathcal{N}} f(x^k) = L$ onde L pode ser ($-\infty$).

Logo $\lim_{k \in k_1} f(x^k) = L$, pois se uma sequência tem um limite, toda subsequência tem o mesmo limite.

Agora, como f é contínua, $\lim_{k \in k_1} x^k = x^*$ implica que $L = f(x^*)$.

Logo $L \neq -\infty$ e $\lim_{k \in \mathcal{N}} f(x^k) = f(x^*)$.

Como $f(x^k)$ é decrescente:

$$f(x^k) \geq f(x^*) \, \forall \, k \in \mathcal{N}$$

Pela convergência da sequência $f(x^k)$:

$$\lim_{k \in k_1} \left(f(x^{k+1}) - f(x^k) \right) = 0 \tag{25}$$

Portanto:

$$f(x^{k+1}) \leq f(x^k) - \alpha\, \theta\, \|\nabla f(x^k)\|\, \|\, x^{k+1} - x^k \,\| \Rightarrow$$

$$\frac{f(x^{k+1}) - f(x^k)}{\|\, \nabla f(x^k) \,\|} \leq -\alpha\, \theta\, \|\, x^{k+1} - x^k \,\| \Rightarrow$$

$$\|\, x^{k+1} - x^k \,\| \leq \frac{f(x^k) - f(x^{k+1})}{\|\, \nabla(x^k) \,\|\, \alpha\, \theta}$$

Como :

$$\lim_{k \in k_1} x^k = x^* \Rightarrow$$

$$\lim_{k \in k_1} \nabla f(x^k) = \nabla f(x^*) \neq 0 \Rightarrow$$

$$\lim_{k \in k_1} \|\nabla f(x^k)\| \neq 0$$

Por (25) , segue que :

$$\begin{aligned} \lim_{k \in k_1} \|x^{k+1} - x^k\| &= 0 \Rightarrow \\ \lim_{k \in k_1} (x^{k+1} - x^k) &= 0 \end{aligned} \tag{26}$$

Existem duas possibilidades:

(i) $\exists\ k_2 \subset_\infty k_1$ tal que $x^{k+1} = x^k + d_k,\ \forall\ k \in k_2$, isto é, sempre há sucesso em *Armijo* com direção pura para alguma subsequência.

(ii) $\exists\ k_0 \in N$ tal que $\forall\ k \geq k_0\ ,\ x^{k+1} = x^k + \lambda_k\, d_k,\ \ \lambda_k < 1$ (a negação de (i)).

- Analisando (i) :

Temos que:

$$d_k = x^{k+1} - x^k\ ,\ \forall\ k \in k_2$$

Portanto:

$$lim_{k \in k_2}\ d_k = lim_{k \in k_2}\ (x^{k+1} - x^k) \overset{(26)}{=} 0 \Rightarrow$$

$$lim_{k \in k_2}\ d_k = 0$$

Mas

$$\|d_k\| \geq \beta\ \|\nabla f(x^k)\| \Rightarrow$$

$$lim_{k \in k_2} \|\nabla f(x^k)\| = 0 \Rightarrow$$

$$\nabla f(x^*) = 0$$

absurdo !

- Analisando (ii) :

Sejam:

- $\lambda_k = \lambda_{novo}$ tal que $f(x^k + \lambda_k d_k) \leq f(x^k) + \alpha \lambda_k d_k^T \nabla f(x^k)$ e
- $\overline{\lambda_k} = \lambda_{velho}$ (fracassado) que gerou λ_k

Assim:

$$\lambda_k \in [0.1\,\overline{\lambda_k},\, 0.9\,\overline{\lambda_k}\,] \Rightarrow \lambda_k \geq 0.1\,\overline{\lambda_k} \Rightarrow \overline{\lambda_k} \leq 10\,\lambda_k$$

Como *Armijo* não foi satisfeito, para $\overline{\lambda_k}$ temos:

$$f(x^k + \overline{\lambda_k}\, d_k) > f(x^k) + \alpha\, \overline{\lambda_k}\, d_k^T\, \nabla f(x^k)$$

ou seja:

$$\frac{f(x^k + \overline{\lambda_k}\, d_k) - f(x^k)}{\overline{\lambda_k}} > \alpha\, d_k^T\, \nabla f(x^k)$$

Pelo Teorema do Valor Médio:

$$\frac{f(x^k + \overline{\lambda_k}\, d_k) - f(x^k)}{\overline{\lambda_k}} = \frac{\nabla f(x^k + \xi_k\, \overline{\lambda_k}\, d_k)^T\, \overline{\lambda_k}\, d_k}{\overline{\lambda_k}}, \quad \xi_k \in [0,1]$$

$$\Rightarrow \frac{\nabla f(x^k + \xi_k\, \overline{\lambda_k}\, d_k)^T\, \overline{\lambda_k}\, d_k}{\overline{\lambda_k}} > \alpha\, d_k^T\, \nabla f(x^k) \;\; \xi_k \in [0,1].$$

Dividindo a desigualdade por $\|\, d_k \,\|$ e chamando:

$$\frac{d_k}{\|d_k\|} = \tilde{d}_k$$

temos $\|\tilde{d}_k\| = 1$ (esfera unitária - que é um conjunto compacto) e

$$\nabla f(x^k + \xi_k\, \overline{\lambda_k}\, d_k)^T\, \tilde{d}_k > \alpha\, \tilde{d}_k^T\, \nabla f(x^k). \tag{27}$$

Como num conjunto compacto toda sequência possui uma subsequência convergente, segue que:

$$\exists\, k_3 \;\subset_\infty\; k_1 \quad \text{tal que} \quad \lim_{k \in k_3} \tilde{d}_k \;=\; \tilde{d}$$

Além disso,

$$\lim_{k \in k_1} \lambda_k\, d_k \;=\; \lim_{k \in k_1} x^{k+1} - x^k \;=\; 0 \;\Rightarrow\; \lim_{k \in k_1} \|\overline{\lambda_k}\, d_k\| \;\leq\; \lim_{k \in k_1} 10\, \|\lambda_k\, d_k\| \;=\; 0$$

$$\Rightarrow\; \overline{\lambda_k}\, d_k \;\longrightarrow\; 0 \qquad \quad para \;\; k \;\in\; k_1$$

$$\Rightarrow\; \lim_{k \in k_1} \left(x^k \;+\; \xi_k\, \overline{\lambda_k}\, d_k\right) \;=\; \lim_{k \in k_1} x^k \;=\; x^*$$

Assim, passando (27) ao limite para $k \;\in\; k_3 \;(k_3 \;\subset_\infty\; k_1)$, temos:

$$\lim_{k \in k_3} \;\nabla f(x^k + \xi_k\, \overline{\lambda_k}\, d_k)^T\, \tilde{d}_k \;=\; \nabla f(x^*)^T\, \tilde{d} \;\geq\; \alpha\, \nabla f(x^*)^T\, \tilde{d}_k$$

Agora, $\nabla f(x^*)^T\, \tilde{d} \;<\; 0.$

De fato, como:

$$\tilde{d} \;=\; \lim_{k \in k_3} \frac{d_k}{\|d_k\|}$$

$$\lim_{k \in k_3} x^k \;=\; x^*$$

e d_k é tal que:

$$\nabla f(x^k)^T\, d_k \;\leq\; -\,\theta\, \|\nabla f(x^k)\|\, \|d_k\| \qquad \theta \;\in\; (0,1)$$

vem que

$$\nabla f(x^k)^T\, \frac{d_k}{\|d_k\|} \;\leq\; -\,\theta\, \|\nabla f(x^k)\|.$$

Logo

$$\lim_{k \in k_3} \nabla f(x^k)^T\, \frac{d_k}{\|d_k\|} \;=\; \nabla f(x^*)\, \tilde{d} \;\leq\; -\,\theta\, \|\nabla f(x^*)\| \;<\; 0$$

Desta forma, $\nabla f(x^*)^T \tilde{d} \geq \alpha \nabla f(x^*)^T \tilde{d}_k$ implica que $\alpha \geq 1$

absurdo !

Logo $\nabla f(x^*) = 0$

□

4.2 Globalização do Método de Newton para minimização sem restrições

Algoritmo 4.2

Dado x^0, calcular $\nabla f(x^0)$.

1. Se $\nabla f(x^k) = 0$, pare.
2. Calcule $\nabla^2 f(x^k)$
3. Resolver o sistema
$$\nabla^2 f(x^k)\, d_k = -\nabla f(x^k)$$
4. Encontre λ_k tal que a condição de *Armijo* seja satisfeita (igual ao que foi feito no Algoritmo 4.1)
5. $x^{k+1} = x^k + \lambda_k d_k$
6. Calcule $\nabla f(x^{k+1})$ e retorne a (1)

Sabemos que o Método de Newton tem convergência local quadrática. Vamos analisar o Algoritimo (4.2) para transformá-lo num método globalmente convergente.

Para as hipóteses do teorema de convergência global serem verificadas, a direção d_k deve satisfazer:

(a) Condição de descida

$$d_k^T \nabla f(x^k) \leq -\theta \|\nabla f(x^k)\| \|d_k\| \tag{28}$$

(b) $\|d_k\| \geq \beta \|\nabla f(x^k)\|$

- Analisemos a condição (a)

Como $d_k = -\nabla^2 f(x^k)^{-1} \nabla f(x^k)$, podemos escrever (28) da seguinte forma:

$$-\nabla f(x^k) \nabla^2 f(x^k)^{-1} \nabla f(x^k) \leq -\theta \| \nabla f(x^k)\| \| \nabla^2 f(x^k)^{-1} \nabla f(x^k)\| \Rightarrow$$

$$\Rightarrow \nabla f(x^k) \nabla^2 f(x^k)^{-1} \nabla f(x^k) > 0$$

Isso acontece se tivermos $\nabla^2 f(x^k) > 0$ (Hessiana positiva definida).

- Condição (b)

$$\|\nabla^2 f(x^k) d_k\| = \|\nabla f(x^k)\| \leq \|\nabla^2 f(x^k)\| \|d_k\| \Rightarrow$$

$$\Rightarrow \|d_k\| \geq \frac{\|\nabla f(x^k)\|}{\|\nabla^2 f(x^k)\|}$$

Se garantimos que $\|\nabla^2 f(x^k)\| \leq \sigma$, isto é, a matriz hessiana é uniformemente limitada para todo $x \in \Omega$, então:

$$\|d_k\| \geq \frac{1}{\sigma} \|\nabla f(x^k)\|$$

isto é, a condição é satisfeita para $\beta = \frac{1}{\sigma}$.

Assim, para o Método de Newton ter convergência global, temos que resolver dois problemas:

(i) A matriz hessiana deve sempre ser positiva definida.

(ii) Garantir $\|\nabla^2 f(x^k)\| \leq \sigma, \forall\, x \in \Omega$.

- Resolvendo (i)

  Uma maneira segura de verificarmos se a matriz é positiva definida é fazer a decomposição de Cholesky.

  Se $\nabla^2 f(x^k) \not> 0$ (Hessiana não é positiva definida), então tomamos $\underline{\mu}$ tal que $(\nabla^2 f(x^k) + \mu I) > 0$ (Hessiana modificada é positiva definida).

  O Algoritmo (4.3), abaixo, é uma versão do Algoritmo (4.2) que resolve o problema (i).

Algoritmo 4.3

Dado $x^0 \in \Omega$, $\theta \in (0,1)$, α, β

Passo 1:- Se $\nabla f(x^k) = 0$, pare.

Passo 2:- Tentar a fatoração de Cholesky de $\nabla^2 f(x^k)$.

Passo 3:- Se o Passo 2 der certo, obter d_k resolvendo

$$LL^T d_k = -\nabla f(x^k)$$

Passo 4:- Se o Passo 2 não der certo, definir

$$B_k = \nabla^2 f(x^k) + \mu I, \; \mu > 0$$

de maneira que $B_k > 0$ (positiva definida)

[Por exemplo: $\mu = max \sum_{j \neq i} | a_{ij} |$]

Obter a fatoração de Cholesky da nova B_k e resolver:

$$LL^T d_k = -\nabla f(x^k)$$

Passo 5:- Testar o fato (improvável) de que:

$$\nabla f(x^k)^T d_k > -\theta \|\nabla f(x^k)\| \|d_k\| \qquad (29)$$

Se (29) não se verifica vá para o Passo 6. Senão modificar d_k como indicado abaixo:

$d_{k\,novo} = \tilde{\lambda} (d_k + \nabla f(x^k)) - \nabla f(x^k)$ (ver Apêndice A).

$d_k = d_{k\,novo}$

Passo 6:- Se $\|d_k\| \geq \beta \|\nabla f(x^k)\|$ vá para o Passo 7. Se não, faça:

$$d_k = \frac{d_k}{\|d_k\|} \beta \|\nabla f(x^k)\|$$

Passo 7:- Encontre λ_k tal que a condição de Armijo seja satisfeita (análoga ao Algoritmo 4.1)

Passo 8:- O novo ponto

$$x^{k+1} = x^k + \lambda_k d_k$$

Retorne ao Passo 1.

- Resolvendo (ii)

Para garantir (ii), basta usar o fato de que o conjunto

$$A = \{x \; tal \; que \; f(x) \leq f(x^0)\} \; \text{é compacto}^6$$

Então, temos:

[6] A prova que $\underline{\mathbf{A}}$ é compacto, encontra-se no Capítulo 1, Teorema (1.2)

$$f \,:\, A \subset \Re^n \longrightarrow \Re \text{ de classe } C^2, \ \underline{A} \text{ compacto}$$

Assim, f e todas as suas derivadas parciais até ordem 2 são limitadas, isto é:

$$\|f\|, \ \|\nabla f(x^k)\|, \ \|\nabla^2 f(x^k)\| \ \textit{são limitados}$$

Com isso, as hipóteses do teorema de convergência global (4.2) são verificadas para o Algoritmo de Newton. Logo o Método de Newton pode ser transformado num método globalmente convergente.

4.3 Globalização dos Métodos quase-Newton para Minimização sem restrições

Já vimos que a direção de um Método Quase-Newton é dado por:

$$d = -H_k \nabla f(x^k)$$

Para d ser uma direção de descida devemos ter:

$$\langle d, \nabla f(x^k)\rangle < 0 \Rightarrow \langle -H_k \nabla f(x^k), \nabla f(x^k)\rangle =$$

$$= -\nabla f(x^k)^T H_k \nabla f(x^k) < 0 \Leftrightarrow H_k > 0$$

(H_k deve ser positiva definida para todo k)

Para garantirmos que $H_k > 0$ em algumas fórmulas de atualização de H_k partindo de H_0 simétrica e positiva definida (exemplo: DFP, BFGS) devemos ter $y^T s > 0$ [17] isto é:

$$\begin{aligned} y^T s &= \langle \nabla f(x^{k+1}) - \nabla f(x^k), s \rangle \\ &= \langle \nabla f(x^k + \lambda_k s) - \nabla f(x^k), s \rangle \\ &= \langle \nabla f(x^k + \lambda_k s), s \rangle - \langle \nabla f(x^k), s \rangle > 0 \end{aligned}$$

Assim

$$\langle \nabla f(x^k + \lambda_k s), s \rangle > \langle \nabla f(x^k), s \rangle.$$

Temos, então, uma nova condição do Método quase-Newton para minimização sem restrições que é dada por:

$$\langle \nabla f(x^k + \lambda_k s), s \rangle > \langle \nabla f(x^k), s \rangle. \tag{30}$$

Escrevemos um novo algoritmo que garanta a condição nova e a condição de Armijo.

Algoritmo 4.4

Sejam $x^0 \in \Omega$, $\beta > 0$, $\alpha \in (0,1)$, $\theta \in [0,1]$, H_0 não singular, simétrica e positiva definida.

Passo 1:- Se $\nabla f(x^0) = 0$ parar.

Passo 2:- Cálculo da direção d_k

$$d_k = - H_k \nabla f(x^k)$$

Passo 3:- Se $\nabla f(x^k)^T d_k \leq -\theta \, \|\nabla f(x^k)\| \, \|d_k\|$ vá para o Passo 5.

Passo 4:- Modificação na direção d_k (ver Apêndice A)

$$\tilde{\alpha} = \frac{(1-\theta) \, \|\nabla f(x^k)\|^2}{\nabla f(x^k)^T d_k + (1-\theta) \, \|\nabla f(x^k)\|^2 + \theta \, \|\nabla f(x^k)\| \, \|d_k\|}$$

$$d_k = \tilde{\alpha}(d_k + \nabla f(x^k)) - \nabla f(x^k)$$

Passo 5:- Encontrar λ_k tal que

- (i) $f(x^k + \lambda_k d_k) \leq f(x^k) + \alpha \lambda_k \langle \nabla f(x^k), d_k \rangle$
- (ii) $\langle \nabla f(x^k + \lambda_k d_k), d_k \rangle \geq \beta \langle \nabla f(x^k), d_k \rangle$

sejam satisfeitos.

Passo 6:- Novo ponto

$$x^{k+1} = x^k + \lambda_k d_k$$

Passo 7:- Se $\nabla f(x^{k+1}) = 0$ parar.

Passo 8:- Atualização H_k

Sejam $s_k = x^{k+1} - x^k$ e $y_k = \nabla f(x^{k+1}) - \nabla f(x^k)$, H_{k+1} é obtida a partir de H_k, s_k, e y_k, usando procedimentos "baratos".

(H_{k+1} simétrica e positiva definida).

Vá para o Passo 2.

Teorema 4.3 (Convergência Global) *Se $x^* \in \Omega$ é um ponto limite da sequência (x^k) gerada pelo Algoritmo (4.4), então $\nabla f(x^*) = 0$.*

Demonstração:-

Seja $lim_{k \in k_1 \subset \mathcal{N}} \, x^k = x^*$ e suponha $\nabla f(x^*) \neq 0$. Vamos provar que $lim_{k \in k_1 \subset \mathcal{N}} \, (x^{k+1} - x^k) = 0$.

Por Armijo

$$\begin{aligned} f(x^{k+1}) &\leq f(x^k) + \alpha\lambda_k\langle\nabla f(x^k), d_k\rangle = \\ &= f(x^k) + \alpha(x^{k+1} - x^k)^T\nabla f(x^k). \end{aligned}$$

Portanto

$$f(x^{k+1}) \leq f(x^k) + \alpha(x^{k+1} - x^k)^T\nabla f(x^k). \tag{31}$$

Mas

$$\begin{aligned} d_k^T\nabla f(x^k) &\leq -\theta\|\nabla f(x^k)\|\,\|d_k\| \Rightarrow \\ \lambda_k d_k^T\nabla f(x^k) &\leq -\theta\|\nabla f(x^k)\|\,\|\lambda_k d_k\| \Rightarrow \\ (x^{k+1} - x^k)^T\nabla f(x^k) &\leq -\theta\|\nabla f(x^k)\|\,\|x^{k+1} - x^k\|. \end{aligned} \tag{32}$$

Logo substituindo (32) em (31), temos:

$$\begin{aligned} f(x^{k+1}) &\leq f(x^k) + \alpha\,(\,-\theta\,\|\nabla f(x^k)\|\,\|x^{k+1} - x^k\|\,) \Rightarrow \\ f(x^{k+1}) &\leq f(x^k) - \alpha\theta\,\|\nabla f(x^k)\|\,\|x^{k+1} - x^k\|. \end{aligned}$$

Mas $(\ f(x^k)\ )$ é uma sequência decrescente de números reais. Logo, existe

$$lim_{k\in\mathcal{N}}\ f(x) \quad \text{(finito ou infinito)}$$

Seja $L = lim_{k\in\mathcal{N}}\ f(x^k)$.

Como f é contínua e $lim_{k\,\in\,k_1}\ x^k = x^*$ temos $L = f(x^*)$, logo:

$$f(x^k) \geq f(x^*) \qquad \forall\, k \in N.$$

e

$$\lim_{k \in k_1} \left(f(x^{k+1}) - f(x^k) \right) = 0. \tag{33}$$

Portanto

$$f(x^{k+1}) \leq f(x^k) - \alpha\theta \,\|\nabla f(x^k)\| \,\|x^{k+1} - x^k\| \Rightarrow$$

$$\frac{f(x^{k+1}) - f(x^k)}{\|\nabla f(x^k)\|} \leq -\alpha\theta \,\|x^{k+1} - x^k\|$$

$$\|x^{k+1} - x^k\| \leq \frac{f(x^k) - f(x^{k+1})}{\alpha\theta \;\; \nabla f(x^k)}$$

Como $lim_{k \in k_1} x^k = x^*$ e $f \in C^2$ temos:

$$lim_{k \in k_1} \nabla f(x^k) = \nabla f(x^*) \neq 0\,.$$

Por (33), segue que $lim_{k \in k_1} \|x^{k+1} - x^k\| = 0$

$$\textit{que implica} \;\; lim_{k \in k_1} (x^{k+1} - x^k) = 0 \tag{34}$$

Existem duas possibilidades:

(a) $\exists\, k_2 \subset k_1$ tal que $x^{k+1} = x^k + d_k,\ \forall\, k \in k_2$

(b) $\exists\, k_0 \in N$ tal que $\forall\, k \geq k_0,\ x^{k+1} = x^k + \lambda_k d_k,\ 0 < \lambda_k < 1$

- Analisando (a)

  Sabemos que:

$$\begin{aligned} \langle \nabla f(x^{k+1}), d_k \rangle &\geq \beta \, \langle \nabla f(x^k), d_k \rangle \\ d_k^T \, \nabla f(x^{k+1}) &\geq \beta \, d_k^T \, \nabla f(x^k) \end{aligned} \tag{35}$$

  Sabemos que $d_k^T \; \nabla f(x^{k+1}) \; < \; 0$ e $d_k^T \; \nabla f(x^k) \; < \; 0$, pois $d_k \; \nabla f(x^k) \; \leq \; -\theta \; \|\nabla f(x^k)\| \; \|d_k\| \; < \; 0 \; \forall \, k \; \in \; k_2$

  Assim, podemos escrever (35) da seguinte maneira:

$$- \, d_k^T \; \nabla f(x^{k+1}) \; \leq \; -\beta \; d_k^T \; \nabla f(x^k)$$

  Aplicando o limite para $k \; \in \; k_2$ temos:

$$- \; \nabla f(x^*) \; lim_{k \, \in \, k_2} \; d_k^T \; \leq \; -\beta \; \nabla f(x^*) \; lim_{k \, \in \, k_2} \; d_k^T \; \Rightarrow$$

$$1 \; \leq \; \beta, \qquad \text{onde} \; \beta \; \in \; (0,1)$$

absurdo !

- Analisando (b)

  Por Armijo temos:

$$f(x^{k+1}) - f(x^k) \; \leq \; \nabla f(x^k)^T \; (x^{k+1} - x^k)$$

$$f(x^{k+1}) \; \leq \; f(x^k) \; + \; \langle \nabla f(x^k), d_k \rangle \; < \; f(x^k)$$

pois $\langle \nabla f(x^k), d_k \rangle < 0$.

$$f(x^k + \lambda_k d_k) < f(x^k)$$

Passando o limite, temos:

$$f(x^* + \lambda_k d_k) < f(x^*) \qquad para\ 0 < \lambda_k < 1$$

$$absurdo$$

Logo, a afirmação da tese é verdadeira.

Capítulo 5

5 Documentação de Subrotinas

Apresentaremos, a seguir, as subrotinas por nós utilizadas, bem como seus parâmetros.

A subrotina QND "chama" as demais subrotinas e é "chamada" pelo programa principal, que consta, apenas, do problema linear a ser resolvido e dos dados de saída.

5.1 Subrotina QND

Essa subrotina encontra o mínimo de uma função f de N variáveis, sem restrições, por um método quase-Newton.

<u>Parâmetros</u>

(a) Parâmetros de Entrada

- $-N\ldots$ Variável inteira que indica número de variáveis.
- $-M\ldots$ Variável interia que indica número de restrições.
- $-NMAX\ldots$ Variável inteira que indica o valor máximo que M pode assumir. Primeira dimensão das matrizes A, H, AUX, HK e dos vetores s, y, d.

—t... Variável real; parâmetro da transformação da função objetiva z numa função f não linear.

—A... Matriz real de dimensão $NMAX \times N$; matriz de restrição.

—B... Vetor real de dimensão M; é o vetor das constantes.

—C... Vetor real de dimensão N ; contém os coeficientes da função objetiva.

—Q... Variável inteira; frequência com que se faz uma iteração de Newton. Se $Q = 1$ temos o método de Newton.

—$IMPR$... Variável inteira que controla as impressões internas da subrotina. Se $IMPR = 0$ nada será impresso. Se $IMPR > 0$, será impresso a cada $IMPR$ iterações: o número de iterações, número de avaliação de função e gradiente, número de iterações de Newton.

—$MAXIT$... Variável inteira; número máximo de iterações permitida.

—$MAXAV$... Variável inteira; número máximo de avaliação de função e gradiente permitido.

—LP... Variável inteira; número da unidade de saída da subrotina.

—MP... Variável real, usada para normalizar a matriz D da decomposição LDL^T de Cholesky. Todos os elementos de D estão no intervalo (MP, MG).

—MG... Variável real; análogo ao MP.

—EPS... Variável real; é a precisão usada pelo método. O processo termina se a norma (MAX) do gradiente de f no ponto x for menor que EPS ($IER = 0$).

—$EPS1$... Variável real; é a precisão usada na decomposição de Cholesky. Se algum elemento da diagonal D for menor do que $EPS1$, o processo termina com $IER = 1$.

$-ALFA\ldots$ Variável real, usada para atualizar a matriz D no processo quase Newton.

$-TIPO\ldots$ Caracter que indica qual método será utilizado.

- tipo = 'N', usaremos Newton
- tipo = 'M', usaremos Newton Modificado
- tipo = 'Q', usaremos quase-Newton Diagonal
- tipo = 'G', usaremos Greenstadt
- tipo = 'D', usaremos DFP
- tipo = 'B', usaremos BFGS
- tipo = 'P', usaremos Correção de Posto 1
- tipo = 'X', usaremos quase-Newton Estruturado

$-Busca\ldots$ Caracter que indica qual interpolação será utilizada.

- Busca = 'N', não faz busca unidirecional.
- Busca = 'C', faz Interpolação Cúbica.
- Busca = 'G', faz Interpolação Cúbica com minimização.

$-X1\ldots$ Vetor real de dimensão N. Na entrada $X1$ contém o ponto inicial dado. No decorrer do processo, $X1$ conterá o ponto atual. No final, $X1$ conterá o melhor ponto encontrado pelo método.

(b) Parâmetros Auxiliares

$-X0, G0, G1, s, y, d\ldots$ Vetores reais de trabalho, cada um com dimensão N. (No método quase-Newton Estruturado os vetores s, y tem dimensão M.

$-WG\ldots$ É um vetor real que guarda a diagonal da matriz D para o cálculo do vetor y no método quase-Newton Estruturado.

$-AUX, H\ldots$ Matrizes reais,cada uma com dimensão $NMAX \times N$ usadas como matrizes de trabalho. H contém a hessiana de f (aproximada ou verdadeira) em cada iteração do processo.

$-BK\ldots$ Matriz real de dimensão $N \times N$. Esta matriz guarda a subtração acumulada da nova hessiana $(H_{k+1} = H_k + BK)$ nos métodos: **Greenstadt, DFP, BFGS, Correção de Posto 1.**

$-HK\ldots$ Matriz real de dimensão $M \times M$.

(c) Parâmetros de Saída

$-F1\ldots$ Variável real; valor da função f no último ponto encontrado.

$-K\ldots$ Variável inteira; número de iterações efetuadas.

$-KN\ldots$ Variável inteira; número de iterações de Newton.

$-NAFG\ldots$ Variável inteira; número de avaliações de função e gradiente.

$-IER\ldots$ Variável inteira; diagnóstico do erro.

- $IER = 0 \longrightarrow$ o processo convergiu.
- $IER = 1 \longrightarrow$ fracasso na decomposição de Cholesky (a hessiana não é positiva definida).
- $IER = 2 \longrightarrow$ a derivada direcional não é negativa na busca unidirecional.
- $IER = 3 \longrightarrow$ foi atingido o número máximo de iterações ($MAXIT$).
- $IER = 4 \longrightarrow$ foi atingido o número máximo de avaliações de função e gradiente ($MAXAV$)
- $IER = 5 \longrightarrow$ a variável lamb é muito pequena na busca unidirecional, isto é, X_{k+1} é muito próximo de X_k.

$-IFLAG \ldots$ Código de saída

- 0 $\longrightarrow$ convergência
- 2 $\longrightarrow$ o valor do passo é tão pequeno que é insignificante em comparação com o valor inicial.
- 4 $\longrightarrow$ número de avaliações de função e gradiente (número de chamada da **FUNGR**) já atingiu MAX
- 5 $\longrightarrow$ ou o valor inicial de **STEP** não é positivo e/ou a derivada direcional inicial é não negativo.

5.2 Subrotina Factível

Essa subrotina diminui, se necessário, a direção $\underline{d}$ para o ponto não sair da região factível.

Parâmetros

(a) Parâmetros de Entrada

$-N \ldots$ Variável inteira; número de variáveis.

$-M \ldots$ Variável inteira; número de restrições.

$-NMAX \ldots$ Variável inteira; valor máximo que M pode assumir. Primeira dimensão da matriz $\boldsymbol{A}$ e dimensão do vetor direção $\boldsymbol{d}$.

$-A \ldots$ Matriz real de dimensão $NMAX \times N$; matriz de restrição.

$-B \ldots$ Vetor real de dimensão M ; vetor das constantes.

$-X \ldots$ Vetor real de dimensão N ; contém o ponto na iteração atual.

(b) Parâmetro de Saída

$-d\ldots$ Vetor real de dimensão $NMAX$. Direção de descida.

5.3 Subrotina FUNGR

Esta subrotina calcula o valor da função f e de seu gradiente.

<u>Parâmetros</u>

$-N\ldots$ Variável inteira; número de variáveis.

$-M\ldots$ Variável inteira; número de restrições.

$-NMAX\ldots$ Variável inteira; valor máximo que M pode assumir. Primeira dimensão da matriz A e dimensão de vetor s .

$-t\ldots$ Variável real; parâmetro da transformação da função objetiva z numa função f não linear.

$-X\ldots$ Vetor real de dimensão N . Na entrada X contém o vetor $X1$ (ponto na atual iteração).

$-A\ldots$ Matriz real de dimensão $NMAX \times N$; matriz de restrição.

$-B\ldots$ Vetor real de dimensão M ; vetor das constantes.

$-C\ldots$ Vetor real de dimensão N ; contém os coeficientes da função objetiva.

$-s\ldots$ Vetor real de dimensão $NMAX$. Vetor de trabalho ($s = A * X$).

$-F\ldots$ Valor da função no ponto. na entrada F contém o valor da função no ponto $X1$ ($F1$).

$-G\ldots$ Vetor real de dimensão N ; gradiente da função no ponto.
G contém o valor do gradiente no ponto $X1$ $(G1)$.

$-IER\ldots$ diagnóstico do erro. Vide subrotina QND

5.4 Subrotina Hess

Nesta subrotina, temos o cálculo da hessiana da função f.

Parâmetros

$-N\ldots$ Variável inteira; número de variáveis.

$-M\ldots$ Variável inteira; número de restrições.

$-NMAX\ldots$ Variável inteira; valor máximo que M pode assumir. Primeira dimensão da matriz A , da matriz H e da matriz AUX .

$-t\ldots$ Variável real; parâmetro da transformação da função objetiva z numa função f não linear.

$-X\ldots$ Vetor real de dimensão N . Na entrada X contém o vetor $X1$ (ponto na atual iteração).

$-A\ldots$ Matriz real de dimensão $NMAX \times N$; matriz de restrição.

$-B\ldots$ Vetor real de dimensão M ; vetor das constantes.

$-AUX\ldots$ Matriz real de dimensão $NMAX \times M$. Matriz de trabalho; guarda o produto $A^T * D$, onde D é a matriz diagonal cujos elementos são da forma:

$$\bullet\ d_{ii} = \frac{1}{(B - AX)_i^2}$$

$-H\ldots$ Matriz real de dimensão $NMAX \times N$ que contém a hessiana da função f.

$-IER\ldots$ Diagnóstico do erro. Vide subrotina QND.

5.5 Subrotina Cholesky

Esta subrotina faz a decomposição de Cholesky de uma matriz A ($A = L\ D\ L^T$),de ordem N, simétrica e positiva definida.

<u>Parâmetros</u>

$-A\ldots$ Matriz real de ordem N,simétrica e definida positiva de dimensão $NMAX \times N$. Na entrada, é a matriz Hessiana. Na saída contém a matriz L.

$-N\ldots$ Variável inteira; ordem da matriz A.

$-NMAX\ldots$ Variável inteira; primeira dimensão da matriz A .

$-EPS1\ldots$ Precisão desejada no teste do pivô.

$-R\ldots$ Vetor trabalho; real e de dimensão N. Contém os elementos da matriz diagonal.

$-IERC\ldots$ Indica o que ocorreu com o processo.

- $IERC = 0 \rightarrow$ A decomposicão de Cholesky foi feita com sucesso.
- $IERC \neq 0 \rightarrow$ Encontrado pivô negativo na linha de número $IERC$.

5.6 Subrotina SOLVE

Esta subrotina resolve os sistemas triangulares da decomposição de Cholesky, ou seja, $L\,z = B$; $D\,y = z$ e $L^T\,x = y$.

<u>Parâmetros</u>

$-A\ldots$ Matriz real de dimensão $NMAX \times N$, contendo na parte triangular inferior, a matriz L; e, na diagonal, a matriz D

$-N\ldots$ Ordem dos sitemas lineares.

$-NMAX\ldots$ Primeira dimensão da matriz A.

$-B\ldots$ Vetor real de dimensão N. Na entrada B contém o lado direito do sistema linear $A\,x = B$. Na saída, B contém a solução do sistema linear.

5.7 Subrotina GSRCH

Essa subrotina minimiza f na direção d.

<u>Parâmetros</u>

$-N\ldots$ Variável inteira que indica número de variáveis.

$-M\ldots$ Variável interia que indica número de restrições.

$-NMAX\ldots$ Variável inteira que indica o valor máximo que M pode assumir. Primeira dimensão das matrizes A.

$-t\ldots$ Variável real; parâmetro da transformação da função objetiva z numa função f não linear.

$-A\ldots$ Matriz real de dimensão $NMAX \times N$; matriz de restrição.

$-B\ldots$ Vetor real de dimensão M ; é o vetor das constantes.

$-X\ldots$ Vetor real de dimensão N. Contém o ponto $X1$ na iteração atual. Na saída terá o ponto

$$X1 + STEP * d$$

$-CF\ldots$ Contém o vetor C (coeficientes da função objetiva).

$-s\ldots$ Vetor real de dimensão N ; contém a diferença $X1 - X0$.

$-F\ldots$ Valor da função no ponto $X1$ $(F1)$. Naída contém o valor da função no ponto $X1 + STEP * d$.

$-G\ldots$ Valor do gradiente no ponto $X1$ $(G1)$. Na saída contém o valor do gradiente $X1 + STEP * d$.

$-IER\ldots$ Diagnóstico do erro. Vide subrotina QND.

$-d\ldots$ Vetor direção (direção de busca).

$-STEP\ldots$ Comprimento do passo. Entra com valor $LAMB$.

5.8 Subrotina NormaD

Normaliza a Matriz D de modo que todos os seus elementos fiquem no intervalo (MP, MG).

<u>Parâmetros</u>

$-H\ldots$ Matriz de dimensão $NMAX \times N$, cuja diagonal contém a matriz D da decomposição de Cholesky ($H = L\, D\, L^T$).

$-N\ldots$ Ordem da matriz H.

$-NMAX\ldots$ Primeira dimensão da matriz H.

$-MP\ldots$ Variável real, usada para normalizar a matriz D da decomposição LDL^T de Cholesky.

$-MG\ldots$ Análogo ao anterior.

Capítulo 6

6 Resultados Computacionais

Implementamos três tipos diferentes de problemas de programação linear com solução conhecida.

O programa está implementado em linguagem de programação **FORTRAN 77** e os testes foram realizados no computador **DigiRede**.

Apresentaremos, a seguir, o tipo de problema abordado e os resultados obtidos em tabelas. Cada tabela consta dos seguintes ítens:

- Número de variáveis (N).
- Número de restrições (M).
- De quantas em quantas iterações, (Q), o método realiza uma iteração de-Newton (Quando usamos o método de Newton, temos Q=1).
- Parâmetro Barreira (t).
- Tipo de interpolação usado.
- Método utilizado para resolver o PPL.
- Tempo total de CPU (incluindo o tempo gasto para a entrada de dados).
- O número de iterações que o método realizou para encontrar as restrições ativas.

- Número de iterações total realizados pelo método para um determinado t.

- Mensagem de erro.

6.1 Programa ESP

Esse programa gera o seguinte P.P.L.

Dado N, $M = 2 * N - 1$

- Gera a matriz A

```
Para I de 1 até M  faça
                   Início
                   Para J de 1 até N  faça
                                      Início
                                       A(I,J) = 0
                                      Fim
                   Fim
Para I de 1 até (N-1)  faça
                       Início
                        A(I,I) = 1
                        A(I,I+1) = 2
                       Fim
Para I de 1 até N  faça
                   Início
                    A(N-1+I,I) = -1
                   Fim
A(N,1) = 0
A(N,2) = -1
A(N+1,1) = -1
A(N+1,2) = 0
```

- Gera o vetor B

  Para I de 1 até $(N-1)$ faça
  Início
  $B(I) = 10$
  Fim

  Para I de N até M faça
  Início
  $B(I) = 0$
  Fim

- Gera o vetor C

  Para I de 1 até N faça
  Início
  $C(I) = -1$
  Fim

- Solução Ótima

  $X0(1) = 10$
  $X0(2) = 0$

  Para I de 3 até N faça
  Início
  $X0(I) = \frac{10 - X0(I-1)}{2}$
  Fim

- Solução Inicial

  Para I de 1 até N faça
  Início
  $X1(I) = 0.1$
  Fim

Os resultados são mostrados nas tabelas a seguir.

$N = 4$	$M = 7$	$Q = 20$	$t = 0.1$	Interpolação Cúbica
Métodos	CPU total	Restrições Ativas	Nº Iterações	Mensagem de Erro
Newton	00.30	6	9	convergiu
DFP	00.70	22	28	convergiu
Greenstadt	00.50	não encontrou	11	IER = 2
Correção Posto 1	00.50	não encontrou	11	IER = 2
BFGS	00.50	18	24	convergiu
Q.N. Diagonal	01.00	21	41	convergiu
Q.N. Estruturado	03.50	41	53	convergiu

$N = 4$	$M = 7$	$Q = 20$	$t = 0.01$	Interpolação Cúbica
Métodos	CPU total	Restrições Ativas	Nº Iterações	Mensagem de Erro
Newton	00.10	4	10	convergiu
DFP	00.50	8	29	convergiu
Greenstadt	00.20	não encontrou	6	IER = 2
Correção Posto 1	00.30	não encontrou	10	IER = 2
BFGS	00.50	6	22	convergiu
Q.N. Diagonal	01.30	21	54	convergiu
Q.N. Estruturado	02.80	21	42	convergiu

$N = 4$	$M = 7$	$Q = 20$	$t = 0.1$	Interpolação: GSRCH
Métodos	CPU total	Restrições Ativas	Nº Iterações	Mensagem de Erro
Newton	00.30	6	7	convergiu
DFP	00.70	22	31	convergiu
Greenstadt	00.20	não encontrou	7	convergiu
Correção Posto 1	00.60	não encontrou	13	convergiu
BFGS	00.40	17	23	convergiu
Q.N. Diagonal	01.70	41	44	convergiu
Q.N. Estruturado	02.40	32	41	convergiu

N = 4 M = 7 Q = 20 t = 0.01 Interpolação: GSRCH				
Métodos	CPU total	Restrições Ativas	Nº Iterações	Mensagem de Erro
Newton	00.10	5	8	convergiu
DFP	00.20	não encontrou	12	convergiu
Greenstadt	00.40	não encontrou	7	convergiu
Correção Posto 1	00.30	não encontrou	7	convergiu
BFGS	00.40	8	21	convergiu
Q.N. Diagonal	02.10	22	51	IER = 5
Q.N. Estruturado	04.10	32	61	convergiu

N = 10 M = 19 Q = 20 t = 0.1 Interpolação Cúbica				
Métodos	CPU total	Restrições Ativas	Nº Iterações	Mensagem de Erro
Newton	01.50	6	9	convergiu
DFP	05.80	22	43	convergiu
Greenstadt	01.00	não encontrou	6	IER = 2
Correção Posto 1	01.10	não encontrou	9	IER = 2
BFGS	04.10	21	30	convergiu
Q.N. Diagonal	05.70	21	43	convergiu
Q.N. Estruturado	20.70	24	51	convergiu

N = 10 M = 19 Q = 20 t = 0.01 Interpolação Cúbica				
Métodos	CPU total	Restrições Ativas	Nº Iterações	Mensagem de Erro
Newton	01.90	4	10	convergiu
DFP	05.40	10	41	convergiu
Greenstadt	09.30	21	62	convergiu
Correção Posto 1	01.20	não encontrou	8	IER = 2
BFGS	03.50	15	27	convergiu
Q.N. Diagonal	09.60	21	48	convergiu
Q.N. Estruturado	12.20	21	28	IER = 5

N = 10 M = 19 Q = 20 t = 0.1 Interpolação: GSRCH				
Métodos	CPU total	Restrições Ativas	Nº Iterações	Mensagem de Erro
Newton	01.20	6	7	convergiu
DFP	04.80	24	41	convergiu
Greenstadt	01.50	não encontrou	10	convergiu
Correção Posto 1	01.20	não encontrou	6	convergiu
BFGS	02.60	20	22	convergiu
Q.N. Diagonal	05.60	22	44	convergiu
Q.N. Estruturado	16.30	21	42	convergiu

N = 10 M = 19 Q = 20 t = 0.01 Interpolação: GSRCH				
Métodos	CPU total	Restrições Ativas	Nº Iterações	Mensagem de Erro
Newton	01.80	5	9	convergiu
DFP	07.60	41	56	convergiu
Greenstadt	09.80	21	63	convergiu
Correção Posto 1	01.60	não encontrou	9	convergiu
BFGS	03.70	12	31	convergiu
Q.N. Diagonal	10.30	22	58	IER =5
Q.N. Estruturado	19.00	22	46	convergiu

N = 20 M = 39 Q = 20 t = 0.1 Interpolação Cúbica				
Métodos	CPU total	Restrições Ativas	Nº Iterações	Mensagem de Erro
Newton	08.30	6	9	convergiu
DFP	20.20	25	41	convergiu
Greenstadt	06.70	não encontrou	11	IER = 2
Correção Posto 1	04.50	não encontrou	8	IER = 2
BFGS	13.60	19	28	convergiu
Q.N. Diagonal	21.50	21	42	convergiu
Q.N. Estruturado	01:18.90	21	48	convergiu

N = 20 M = 39 Q = 20 t = 0.1 Interpolação: GSRCH				
Métodos	CPU total	Restrições Ativas	No Iterações	Mensagem de Erro
Newton	06.80	6	7	convergiu
DFP	14.60	22	30	convergiu
Greenstadt	04.40	não encontrou	7	convergiu
Correção Posto 1	04.80	não encontrou	9	convergiu
BFGS	11.50	19	22	convergiu
Q.N. Diagonal	19.00	21	43	convergiu
Q.N. Estruturado	01:50.30	21	42	convergiu

N = 40 M = 79 Q = 20 t = 0.1 Interpolação Cúbica				
Métodos	CPU total	Restrições Ativas	No Iterações	Mensagem de Erro
Newton	46.60	6	9	convergiu
DFP	01:15.90	22	39	convergiu
Greenstadt	22.90	não encontrou	9	IER = 2
Correção Posto 1	17.10	não encontrou	7	IER = 2
BFGS	52.70	21	26	convergiu
Q.N. Diagonal	01:19.80	21	41	convergiu
Q.N. Estruturado	05:36.40	21	49	convergiu

N = 40 M = 79 Q = 20 t = 0.1 Interpolação: GSRCH				
Métodos	CPU total	Restrições Ativas	No Iterações	Mensagem de Erro
Newton	44.50	6	8	convergiu
DFP	01:27.00	26	34	convergiu
Greenstadt	24.20	não encontrou	10	convergiu
Correção Posto 1	17.30	não encontrou	7	convergiu
BFGS	48.70	21	24	convergiu
Q.N. Diagonal	01:15.90	22	43	convergiu
Q.N. Estruturado	04:42.90	21	37	IER = 5

N = 75 M = 149 Q = 20 t = 0.1 Interpolação Cúbica				
Métodos	CPU total	Restrições Ativas	Nº Iterações	Mensagem de Erro
Newton	05:14.00	6	10	convergiu
DFP	04:06.80	24	33	convergiu
Greenstadt	01:22.60	não encontrou	8	IER = 2
Correção Posto 1	01:09.20	não encontrou	7	IER = 2
BFGS	05:21.20	18	32	IER = 5
Q.N. Diagonal	05:27.20	21	41	convergiu
Q.N. Estruturado	20:41.90	21	45	IER = 5

N = 100 M = 199 Q = 20 t = 0.1 Interpolação Cúbica				
Métodos	CPU total	Restrições Ativas	Nº Iterações	Mensagem de Erro
Newton	11:46.40	6	10	convergiu
DFP	09:40.40	23	37	converigu
Greenstadt	04:34.40	não encontrou	19	IER = 2
Correção Posto 1	02:18.80	não encontrou	7	IER = 2
BFGS	07:53.60	21	30	convergiu
Q.N. Diagonal	09:40.60	21	42	convergiu
Q.N. Estruturado	44:23.00	21	48	IER = 5

6.2 Programa GD

Esse programa gera o seguinte P.P.L.

Dados N, M

- Gera o problema

  Para I de 1 até M faça
  Início
  Para J de 1 até N faça
  Início
  $A(I,J) = 0$
  Fim
  Fim

  – Gera a faixa diagonal da Matriz A.

  $AUX1 = M - N$
  $L = 1$
  $K = N$

  Para J de 1 até N faça
  Início
  Se $J + AUX1 \leq N$ então $k = J + AUX1$
  Para I de J até k faça
  Início
  $A(I,J) = 5|sin(L)| + 5$
  $L = L + 1$
  Fim
  Fim

  Para I de 1 até N faça
  Início
  $X1(I) = 2\,|sin(L)| + 1$
  $L = L + 1$
  Fim

Para I de 1 até N faça
Início
$B(I) = 0$
Para J de 1 até N faça
Início
$B(I) = B(I) + A(I, J)X1(J)$
Fim
Fim

– Gera as <u>AUX</u> linhas restantes.

Para I de $(N + 1)$ até M faça
Início
$K = I - AUX1$
10 Para J de 1 até N faça
Início
Se $J \geq K$ então
Início
$A(I, J) = 5|sin(L)| + 5$
$L = L + 1$
Fim
Fim

$B(I) = B(N) + 5|sin(L)|$
$L = L + 1$
$z = 0$
Para J de 1 até N faça
Início
$z = z + A(I, J)X1(J)$
Fim
Se $z \geq B(I)$ vá para 10
Fim

Para I de 1 até N faça
Início
$MI(I) = 3|sin(L)| + 3$
$L = L + 1$
Fim

Para J de 1 até N faça
Início
$C(J) = 0$
Para I de 1 até N faça
Início
$C(J) = C(J) - A(I,J)MI(I)$
Fim
Fim

- Solução Inicial

Para J de 1 até N faça
Início
$X0(J) = X1(J)$
$X1(J) = 0$
Fim

As tabelas abaixo são referentes ao problema descrito acima GD

N = 15 M = 20 Q = 20 t = 0.1 Interpolação Cúbica				
Métodos	CPU total	Restrições Ativas	Nº Iterações	Mensagem de Erro
Newton	05.10	6	11	IER = 5
BFGS	12.20	21	48	IER = 5
Q.N. Diagonal	01:10.70	223	278	IER = 5
Q.N. Estruturado	04:55.80	não encontrou	500	IER = 3

N = 15 M = 20 Q = 20 t = 0.1 Interpolação: GSRCH				
Métodos	CPU total	Restrições Ativas	Nº Iterações	Mensagem de Erro
Newton	07.10	6	13	IER = 5
BFGS	14.70	22	48	IER = 5
Q.N. Diagonal	01:00.00	201	248	IER = 5
Q.N. Estruturado	04:45.30	não encontrou	500	IER = 3

N = 15	M = 20	Q = 20	t = 0.01	Interpolação Cúbica
Métodos	CPU total	Restrições Ativas	Nº Iterações	Mensagem de Erro
Newton	05.00	6	12	IER = 5
BFGS	08.30	18	36	IER = 2
Q.N. Diagonal	01:48.00	não encontrou	452	IER = 4
Q.N. Estruturado	20.40	não encontrou	37	IER = 5

N = 15	M = 20	Q = 20	t = 0.01	Interpolação: GSRCH
Métodos	CPU total	Restrições Ativas	Nº Iterações	Mensagem de Erro
Newton	05.30	6	12	IER = 5
BFGS	22.70	21	53	IER = 5
Q.N. Diagonal	01:53.30	não encontrou	500	IER = 3
Q.N. Estruturado	04:59.00	não encontrou	500	IER = 3

N = 20	M = 50	Q = 20	t = 0.1	Interpolação Cúbica
Métodos	CPU total	Restrições Ativas	Nº Iterações	Mensagem de Erro
Newton	39.30	7	22	IER = 5
BFGS	41.60	34	57	IER = 5
Q.N. Diagonal	05:58.50	não encontrou	365	IER = 4
Q.N. Estruturado	18:59.00	não encontrou	500	IER = 3

N = 20	M = 50	Q = 20	t = 0.1	Interpolação GSRCH
Métodos	CPU total	Restrições Ativas	Nº Iterações	Mensagem de Erro
Newton	34.60	8	13	IER = 5
BFGS	01:01.00	34	66	IER = 5
Q.N. Diagonal	06:12.70	não encontrou	403	IER = 4
Q.N. Estruturado	01:32.00	não encontrou	40	IER = 5

N = 20	M = 50	Q = 20	t = 0.01	Interpolação GSRCH
Métodos	CPU total	Restrições Ativas	Nº Iterações	Mensagem de Erro
Newton	52.30	14	21	IER = 5
BFGS	01:03.70	23	50	IER = 5
Q.N. Diagonal	01:04.20	não encontrou	83	IER = 5
Q.N. Estruturado	17:17.80	não encontrou	438	IER = 4

N = 70	M = 72	Q = 20	t = 0.1	Interpolação Cúbica
Métodos	CPU total	Restrições Ativas	Nº Iterações	Mensagem de Erro
Newton	03:25.20	5	13	IER = 5
BFGS	31:01.90	416	419	IER = 2
Q.N. Diagonal	29:15.20	não encontrou	500	IER = 3
Q.N. Estruturado	+ 2 hs	não encontrou	250	Interrompido

N = 20	M = 200	Q = 20	t = 0.1	Interpolação Cúbica
Métodos	CPU total	Restrições Ativas	Nº Iterações	Mensagem de Erro
Newton	02:44.70	9	22	IER = 5
BFGS	03:03.30	48	68	IER = 5
Q.N. Diagonal	24:12.40	não encontrou	341	IER = 4
Q.N. Estruturado	+ 2:00	não encontrou	240	Interrompido

N = 70	M = 72	Q = 20	t = 0.1	Interpolação: GSRCH
Métodos	CPU total	Restrições Ativas	Nº Iterações	Mensagem de Erro
Q.N. Diagonal	05:41.70	76	109	IER = 5

6.3 Programa GERA

- Gera o Problema

```
$L = 1$
Para $J$ de 1 até $N$  faça
                       Início
                       Para $I$ de 1 até $N$  faça
                                              Início
                                              $A(I,J) = 5|sin(L)| + 5$
                                              $L = L + 1$
                                              Fim
                       Fim
Para $I$ de 1 até $N$  faça
                       Início
                       $X1(I) = 2|sin(L)| + 1$
                       $L = L + 1$
                       Fim
Para $I$ de 1 até $N$  faça
                       Início
                       $B(I) = 0$
                       Para $J$ de 1 até $N$  faça
                                              Início
                                              $B(I) = B(I) + A(I,J)X1(J)$
                                              Fim
                       Fim
Para $I$ de  $(N+1)$ até $M$ faça
            Início
10          Para $J$ de 1 até $N$          faça
                                           Início
                                           $A(I,J) = 5|sin(L)| + 5$
                                           $L = L + 1$
                                           Fim
            $B(I) = B(1) + 5\ sin(L)$
            $L = L + 1$
            $z = 0$
```

Para J de 1 até N faça
Início
$z = z + A(I,J)X1(J)$
Fim
Se $z \geq B(I)$ vá para 10
Fim

Para I de 1 até N faça
Início
$MI(I) = 3|sin(L)| + 3$
$L = L + 1$
Fim

Para J de 1 até N faça
Início
$C(J) = 0$
Para I de 1 até N faça
Início
$C(J) = C(J) - A(I,J)MI(I)$
Fim
Fim

- Solução Inicial

Para J de 1 até N faça
Início
$X0(J) = X1(J)$
$X1(J) = 0$
Fim

As tabelas a seguir estão relacionadas com o programa **GERA**

N = 20	M = 40	Q = 20	t = 0.1	Interpolação Cúbica
Métodos	CPU total	Restrições Ativas	Nº Iterações	Mensagem de Erro
Newton	25.00	5	14	IER = 5
BFGS	12.50	10	18	IER = 2
Q.N. Diagonal	04:22.10	não encontrou	212	IER = 4
Q.N. Estruturado	12.60	não encontrou	6	IER = 1

N = 20	M = 50	Q = 20	t = 0.1	Interpolação Cúbica
Métodos	CPU total	Restrições Ativas	No Iterações	Mensagem de Erro
Newton	45.80	5	19	IER = 5
BFGS	38.30	9	42	IER = 5
Q.N. Diagonal	05:18.90	não encontrou	201	IER = 4
Q.N. Estruturado	16.50	não encontrou	6	IER = 1

N = 20	M = 50	Q = 20	t = 0.1	Interpolação GSRCH
Métodos	CPU total	Restrições Ativas	No Iterações	Mensagem de Erro
Newton	32.50	7	13	IER = 5
BFGS	28.00	13	25	IER = 5
Q.N. Diagonal	37.60	não encontrou	30	IER = 5
Q.N. Estruturado	19.20	não encontrou	7	IER = 1

N = 20	M = 50	Q = 20	t = 0.01	Interpolação GSRCH
Métodos	CPU total	Restrições Ativas	No Iterações	Mensagem de Erro
Newton	47.20	não encontrou	14	IER = 5
BFGS	37.60	13	36	IER = 5
Q.N. Diagonal	11.20	não encontrou	10	IER = 5
Q.N. Estruturado	25.90	não encontrou	10	IER = 1

N = 20	M = 50	Q = 20	t = 0.01	Interpolação Cúbica
Métodos	CPU total	Restrições Ativas	No Iterações	Mensagem de Erro
Newton	25.20	não encontrou	8	IER = 5
BFGS	18.40	9	20	IER = 5
Q.N. Diagonal	16.80	não encontrou	13	IER = 5
Q.N. Estruturado	43.90	não encontrou	15	IER = 5

N = 40 M = 80 Q = 20 t = 0.1 Interpolação Cúbica				
Métodos	CPU total	Restrições Ativas	Nº Iterações	Mensagem de Erro
Newton	02:44.90	não encontrou	10	IER = 5
BFGS	01:18.40	16	22	IER = 5
Q.N. Diagonal	38.30	não encontrou	9	IER = 5
Q.N. Estruturado	01:19.20	não encontrou	7	IER = 5

N = 75 M = 150 Q = 20 t = 0.1 Interpolação Cúbica				
Métodos	CPU total	Restrições Ativas	Nº Iterações	Mensagem de Erro
Newton	12:46.30	não encontrou	8	IER = 5
BFGS	06:12.80	não encontrou	26	IER = 2
Q.N. Diagonal	02:22.20	não encontrou	7	IER = 5
Q.N. Estruturado	06:34.30	não encontrou	11	IER = 5

Capítulo 7

7 Conclusões

A partir dos resultados obtidos nos dois tipos de Interpolação, levando em conta a variação dos fatores: `número de variáveis`, `número de restrições`, `parâmetro barreira` e `tipo de interpolação` , levantamos algumas conclusões a respeito dos métodos utilizados.

Os métodos Greenstadt, DFP, Correção de Posto 1 não são competitívos, quanto à eficiência computacional, ao serem comparados com o método de Newton.

Quando o número de variáveis aumenta, o método BFGS torna-se mais eficiente que o método de Newton, apesar do número de iterações ter aumentado. Este resultado era esperado, pois o método de Newton calcula a inversa da matriz Hessiana em cada iteração.

Os resultados obtidos pelo método quase-Newton Estruturado não foram satisfatórios, apesar de possuir boas propriedades teóricas. O tempo de CPU (que foi expressivo) não pode ser considerado como tal, pois este método, particularmente, não atualiza B_{k+1}^{-1} em função da B_k^{-1} ; e sim, a B_{k+1} em função de B_k (aproximação da matriz Hessiana). Não compensaria aplicar Sherman Morrison para encontrarmos a B_k^{-1} , visto que o número de iterações também foi muito grande em relaçào aos demais métodos.

Podemos colocar como pontos para futuros estudos, implementar um algoritmo para matrizes esparsas. Esperamos que um programa específico para problemas esparsos seja mais eficiente. Podemos, também, tentar um outro tipo de interpolação.

8 Apêndice A

Modificação na direção d_k se a condição:

$$\nabla f(x^k)^T d_k \leq -\theta \|\nabla f(x^k)\| \|d_k\| \tag{36}$$

não for satisfeita.

Seja:

$$\nabla f(x^k)^T d_k = -\theta_1 \|\nabla f(x^k)\| \|d_k\| > -\theta \|\nabla f(x^k)\| \|\tilde{d}_k\| \tag{37}$$

Tomemos

$$\tilde{d}_k = \alpha d_k + (1-\alpha)[-\nabla f(x^k)], \qquad \alpha \in (0,1) \tag{38}$$

e vamos impor que:

$$\nabla f(x^k)^T \tilde{d}_k \leq -\theta \|\nabla f(x^k)\| \|\tilde{d}_k\| \tag{39}$$

$$\begin{aligned}
\nabla f(x^k)^T \tilde{d}_k &\overset{37}{=} \nabla f(x^k)^T [\alpha d_k + (\alpha-1)\nabla f(x^k)] \\
&\overset{38}{\leq} -\theta \|\nabla f(x^k)\| \|\alpha d_k + (\alpha-1)\nabla f(x^k)\| \Leftrightarrow \\
\nabla f(x^k)^T \tilde{d}_k &\leq -\theta \|\nabla f(x^k)\| + [| \alpha | \|d_k\| + | \alpha - 1 | \|\nabla f(x^k)\|] \Leftrightarrow \\
\nabla f(x^k)^T \tilde{d}_k &\leq -\theta \|\nabla f(x^k)\| [\alpha \|d_k\| + (1-\alpha) \|\nabla f(x^k)\|]
\end{aligned} \tag{40}$$

Assim

$$\nabla f(x^k)^T \tilde{d}_k \overset{37}{=} \alpha \, \nabla f(x^k)^T d_k \, + \, (\alpha - 1)\|\nabla f(x^k)\|^2 \overset{39}{\leq}$$

$$- \alpha \, \theta \, \|\nabla f(x^k)\| \, \|d_k\| \, - \, \theta \, \|\nabla f(x^k)\|^2 \, + \, \alpha \, \theta \, \|\nabla f(x^k)\|^2$$

Então:

$$\alpha \, [\nabla f(x^k)^T d_k + (1{-}\theta) \, \|\nabla f(x^k)\|^2 + \theta \, \|\nabla f(x^k)\| \, \|d_k\|] \, \leq \, (1{-}\theta) \, \|\nabla f(x^k)\|^2 \implies$$

$$\alpha \leq \frac{(1-\theta) \, \|\nabla f(x^k)\|^2}{[\nabla f(x^k)^T d_k \, + \, (1-\theta)\|\nabla f(x^k)\|^2 \, + \, \theta\|\nabla f(x^k)\| \, \|d_k\|]} < 1$$

Como $\alpha \in [0,1]$ para todo $\alpha \in [0, \tilde{\alpha}]$

Tomando

$$\tilde{\alpha} = \frac{(1-\theta) \, \|\nabla f(x^k)\|^2}{\nabla f(x^k)^T d_k \, + \, (1-\theta)\|\nabla f(x^k)\|^2 \, + \, \theta\|\nabla f(x^k)\|\|d_k\|]} \tag{41}$$

a direção $\tilde{d}_k = \tilde{\alpha} \, [d_k + \nabla f(x^k)] - \nabla f(x^k)$ vai satisfazer a condição (36).

Logo, se d_k não satisfaz a condição (36), tomemos

$$d_k \, novo = \tilde{\alpha} \, (d_k + \nabla f(x^k)) - \nabla f(x^k)$$

para $\tilde{\alpha}$ dado por (40).

9 Bibliografia

Referências

[1] Bazaraa,M.S.,and Shetty,C.M., "*Nonlinear Programming - Theory and Algorithms*", John Wiley & Sons, 1979.

[2] Dantzig, G.B., "*Linear Programming and Extensions*". Princeton University Press, Princeton, N.J., 1963.

[3] Dennis, J.E. and Schnabel,R.B. "*Numerical Methods for Unconstrained Optimization and Nonlinear Equations*". Prentice-Hall Series in Comput. Math., Prentice-Hall, N.J., 1983.

[4] Dennis, J.E. and Moré, J.J., "*Quasi-Newton Methods - motivation and theory*". SIAM Review, v.19, 1977.

[5] Fiacco, A.V., "*Estimates of Sensitivity Information Using Penalty Functions*". Introdution to Sensitivity and Stability Analysis in Nonlinear Programming, v.165, Academic Press, N.Y., 1983.

[6] Fletcher, R., "*Practical methods of optimization.*". John Wiley and Sons, London, 1981.

[7] Gill, P.E., Murray, W., Saunders, M.A. and Wright, M.H., "*A Note on Nonlinear Approaches to Linear Programming*". Report SOL 86-7, Systems Optimazation Laboratory, Dept. of Operations Research, Stanford University, Stanford, 1986.

[8] Gill, P.E., Murray, W., Saunders, M.A. and Wright, M.H., "*Recent Developments in Constrained Optimization*". Journal of Computational and Applied Mathematics, vol.22, North-Holland, 1988, pp. 257-70.

[9] Gonzaga, C.C., "*Algoritmos de Pontos Interiores para Programação Linear*". 17º Colóquio Brasileiro de Matemática, IMPA, Rio de Janeiro, 1989.

[10] Lima, E.L., "*Espaços Métricos*". 2ª Ed., Hamburg, R.J., 1977, pp. 215.

[11] Lootsmar, F.A., "*A Hybrid Algorithm for Non-Linear Programming*" Numerical Methods for Nonlinear Optimization. Academic Press, N.Y., 1972.

[12] Luenberger, D.G., "*Linear and Nonlinear Programming*". 2nd editions, Addison Wesley, 1984.

[13] Martínez, J.M., "*A Family os Quasi-Newton Methods with Direct Secant Updates of Matrix Factorizations*". SIAM J. Numer. Anal., 1990.

[14] Martínez, J.M., "*A Quasi-Newton Method with a New Updating for the LDU Factorization of the Approximate Jacobian*", Mat. Aplic. e Comput., vol.2, 1983, pp. 131-42.

[15] Martínez, J.M., "*Quasi-Newton Methods with Factorization Scaling for Solving Sparse Nonlinear Systems of Equations.*", Computing, vol.38, 1987, pp. 133-41.

[16] Martínez, J.M., "*Local Convergence Theory of Inexact Newton Methods Based on Structured Secant Procedures*". Mathematics of Computation, 1990.

[17] Minoux, M., "*Programmation Mathématique.*", Théorie et Algorithmes. BORDAS et C.N.E.T.-E.N.S.T., Paris, 1983.