



Universidade Federal de Santa Catarina
Centro Tecnológico
Programa de Pós-Graduação em Engenharia Elétrica

EEL6000 - MÉTODOS NUMÉRICOS DE OTIMIZAÇÃO

Trabalho Computacional

Alunos:	Daniel Collier Michel Rokembach de Carvalho
Professores:	Edson Luiz e Silva Erlon Cristian Finardi

Florianópolis, Maio de 2008.

SUMÁRIO

1	INTRODUÇÃO	4
2	OBJETIVOS.....	5
3	MÉTODOS DE OTIMIZAÇÃO.....	7
3.1	MÉTODO DO GRADIENTE	7
3.2	MÉTODO QUASE-NEWTON	9
3.3	MÉTODO NEWTON GRADIENTE CONJUGADO	11
3.4	ALGORITMOS AUXILIARES.....	13
3.4.1	Seleção do Passo Inicial de Busca α_0	13
3.4.2	Cálculo do Passo de Busca: “Backtracking Line Search”	14
3.4.3	Direção de Busca pelo Método do Gradiente Conjugado Linear	15
4	RESULTADOS.....	17
4.1	ANÁLISE DA ESCOLHA DO PASSO INICIAL DE BUSCA A_0	18
4.1.1	Método do Gradiente	18
4.1.1.1	Ponto Inicial $x_0 = [-1 \ -2 \ 1 \ -1]^T$	18
4.1.1.2	Outros Pontos Iniciais	22
4.1.1.3	Conclusões	24
4.1.2	Método Quase-Newton	25
4.1.2.1	Ponto Inicial $x_0 = [-1 \ -2 \ 1 \ -1]^T$	25
4.1.2.2	Outros Pontos Iniciais	28
4.1.2.3	Conclusões	29
4.1.3	Método Newton Gradiente Conjugado.....	30
4.1.3.1	Ponto Inicial $x_0 = [-1 \ -2 \ 1 \ -1]^T$	30
4.1.3.2	Outros Pontos Iniciais	35
4.1.3.3	Conclusões	35
4.2	ANÁLISE DA CONDIÇÃO DE CURVATURA $s_k \times y_k^T$	36
4.3	ANÁLISE DA VARIAÇÃO DO PARÂMETRO DA SEQUÊNCIA FORÇANTE	38
4.4	COMPARAÇÃO ENTRE OS MÉTODOS DE OTIMIZAÇÃO	39
5	CONCLUSÃO	40
6	REFERÊNCIAS BIBLIOGRÁFICAS	41
	APÊNDICE.....	42
	APÊNDICE I - LISTAGEM DO CÓDIGO DO ARQUIVO DE DADOS	42
	APÊNDICE II – LISTAGEM DO CÓDIGO: MÉTODO DO GRADIENTE	43
	APÊNDICE III - LISTAGEM DO CÓDIGO: MÉTODO QUASE-NEWTON	44
	APÊNDICE IV - LISTAGEM DO CÓDIGO: MÉTODO NEWTON-GC	45
	APÊNDICE V - LISTAGEM DO CÓDIGO: “BACKTRANKING LINE SEARCH”	46
	APÊNDICE VI - LISTAGEM DO CÓDIGO: MÉTODO DO GCL	46

APÊNDICE VII - LISTAGEM DO CÓDIGO: FUNÇÃO OBJETIVO	47
APÊNDICE VIII - LISTAGEM DO CÓDIGO: GRADIENTE DA FUNÇÃO OBJETIVO	47
APÊNDICE XI - LISTAGEM DO CÓDIGO: HESSIANA DA FUNÇÃO OBJETIVO	47

1 Introdução

A otimização é uma ferramenta essencial na vida do ser humano. Uma companhia de viação a utiliza pra reduzir ao máximo seus gastos com combustíveis, tripulações e aeronaves. Investidores a utilizam para reduzir seus riscos enquanto esperam obter o maior retorno possível. Fabricantes estão preocupados em obter a máxima eficiência em seus processos produtivos.

Na otimização o primeiro passo é identificar um objetivo, por exemplo o lucro, que é descritos por variáveis que descrevem o problema. A meta é encontrar valores de variáveis que otimizem o processo. Dependendo do caso esse problema pode ser restrito ou irrestrito.

O processo de identificação de objetivos, variáveis e restrições é chamado de modelagem. A construção de um modelo adequado é um passo fundamental para a otimização. Um modelo muito simplista poderá resultar em uma idéia errada do problema, enquanto um muito complexo poderá ser de difícil solução.

Considerando que o modelo foi feito adequadamente um algoritmo de otimização pode ser utilizado pra encontrar a solução. Devido a grande complexidade dos modelos e algoritmos o uso do computador é essencial.

Um detalhe importante é que não há um algoritmo de otimização universal. Os que existem são classes de algoritmos cada um apropriado para determinado problema. Portanto é de responsabilidade do usuário escolher o algoritmo de maneira adequada. Sendo que o uso equivocado pode resultar em soluções demasiadamente lentas para serem encontradas, ou até mesmo não encontrar nenhuma solução.

Após o uso correto do algoritmo o usuário deve ser capaz de interpretar a solução, utilizando conhecimentos matemáticos pode-se concluir se determinada solução realmente pode ser utilizada.

2 Objetivos

O objetivo deste trabalho é o ajuste de uma curva de concentração no sangue de uma determinada droga em função do tempo. O ajuste de curvas é formulado como um problema de mínimos quadrados em que se busca determinar o valor das constantes da função proposta.

A curva experimental com os dados em função do número de amostras é apresentada na Figura 2-1. Os dados das medições estão no Apêndice I, já no formato a ser apresentado ao programa *MatLab*.



Figura 2-1: Curva experimental obtida a partir dos dados das medições.

O modelo proposto é a seguinte combinação de funções exponenciais:

$$f(t) = x_3 e^{x_1 t} + x_4 e^{x_2 t} \quad (2-1)$$

Do ponto de vista da otimização, o objetivo consiste em determinar os valores das constantes x_1 , x_2 , x_3 e x_4 , por meio da resolução do seguinte problema de otimização irrestrita:

$$\min_{x \in \mathbb{R}^4} F(x) = \sum_{i=1}^{45} \frac{1}{2} \left[y_i - (x_3 e^{x_1 t_i} + x_4 e^{x_2 t_i}) \right]^2 \quad (2-2)$$

onde:

- y_i é a i-ésima amostra, com $i \in [1, 45]$;
- t_i é o instante de tempo associado com a i-ésima amostra, dado por $t_i = 0,02i$.

Os métodos a serem utilizados são do Gradiente, Quase-Newton e Newton Gradiente Conjugado. Vale lembrar que tais métodos buscam iterativamente por um ponto de mínimo local, cabendo ao usuário do método analisar a viabilidade de utilizar o ponto. Como por exemplo, analisar se com o ponto encontrado o ajuste de curvas é adequado.

Os métodos foram implementados no ambiente computacional MATLAB®, com as devidas considerações para poupar tempo de execução e ter um bom aproveitamento da memória. Não foram utilizados programas de cálculo simbólico, que aumentariam o tempo de execução. Os algoritmos implementados podem ser facilmente utilizados com qualquer função objetivo desde que sejam modificados os arquivos referentes ao cálculo da função objetivo (fx.m), do gradiente (dfx.m) e da Hessiana (hessiana.m). Esses códigos podem ser encontrados no Apêndice VII, no Apêndice VIII e no Apêndice IX, respectivamente.

3 Métodos de Otimização

Uma apresentação sucinta dos métodos de otimização implementados será realizada, devido à necessidade de serem apresentadas as considerações utilizadas nas implementações computacionais.

Os algoritmos utilizados em todos os métodos (seleção do passo inicial de busca e “backtracking line search”) e o método para cálculo da direção de busca via gradiente conjugado linear será apresentado na seção “Algoritmos Auxiliares”. O entendimento desses algoritmos auxiliares faz-se necessário, uma vez que os seus desempenhos são essenciais para a convergência e para o tempo de execução dos métodos de otimização.

3.1 Método do Gradiente

A principal vantagem do método do gradiente é de utilizar somente o gradiente da função objetivo, não havendo necessidade do cálculo de derivadas de segunda ordem. A direção de descida é dada por:

$$p_k = -\nabla f(x_k) \quad (3.1)$$

Essa direção é sempre de descida, pois $p_k^T \nabla f(x_k) = -\nabla f(x_k)^T \nabla f(x_k) < 0$. Em algoritmos de busca linear essa direção é a de maior decréscimo. A taxa de convergência do método é linear o que é uma desvantagem em relação a outros métodos.

Desvantagens são encontradas em problemas mal condicionados (auto-valores da matriz Hessiana de ordem de grandezas muito diferentes) em que a resolução do problema pode-se torna excessivamente lenta.

Com base nas sugestões e exigências propostas para a implementação do método, montou-se o fluxograma apresentado na Figura 3-1. Os blocos como o da rotina “Backtracking Line Search” serão explicados posteriormente neste capítulo. A listagem do código implementado em *MatLab* é apresentado no Apêndice II.

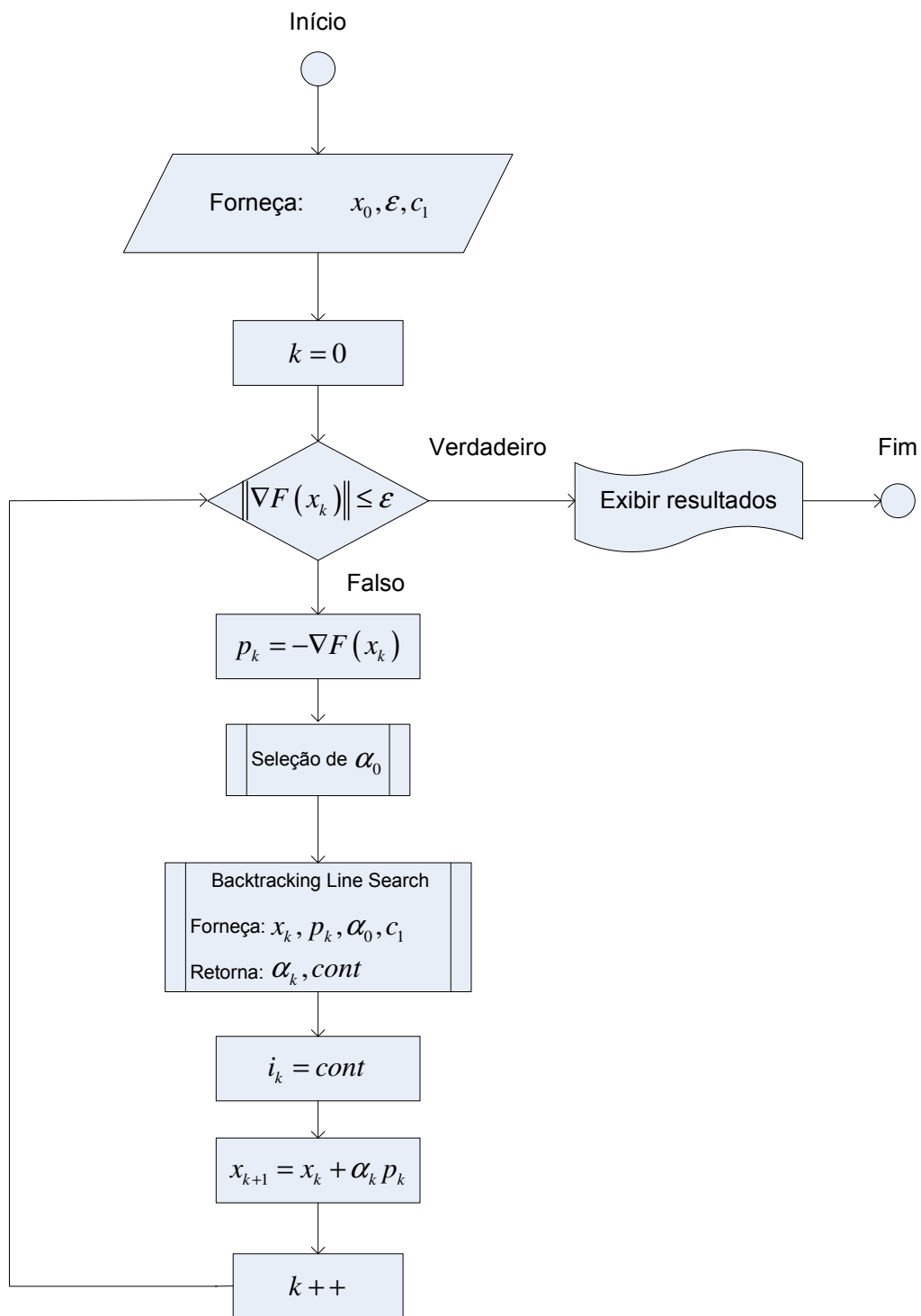


Figura 3-1: Fluxograma do algoritmo do Método do Gradiente.

3.2 Método Quase-Newton

Os métodos de Quase-Newton e o método do gradiente utilizam apenas informações do gradiente da função objetivo para o cálculo da direção de busca. No entanto, o método Quase-Newton pode alcançar taxas de convergências superlinear e em alguns casos ser mais eficiente do que os métodos de Newton, que requerem o cálculo de derivadas de segunda ordem.

A versão do método implementado foi proposta por Broyden, Fletcher, Goldfarb e Shanno (BFGS) e é considerada a variação mais eficiente. A idéia básica do método consiste em atualizar uma aproximação da matriz Hessiana baseado em informações da curvatura medida na interação corrente. Essa idéia é o fator determinante para o baixo custo computacional do método.

É importante ressaltar que a performance do método BFGS pode não ser satisfatória, pois o algoritmo de busca linear não é baseado nas condições de Wolfe. Dessa forma, não há garantias que a condição de curvatura seja atendida ($y_k^T s_k > 0$).

Na Figura 3-2 é apresentado o fluxograma do Algoritmo BFGS. A listagem do código implementado em *MatLab* é apresentado no Apêndice III.

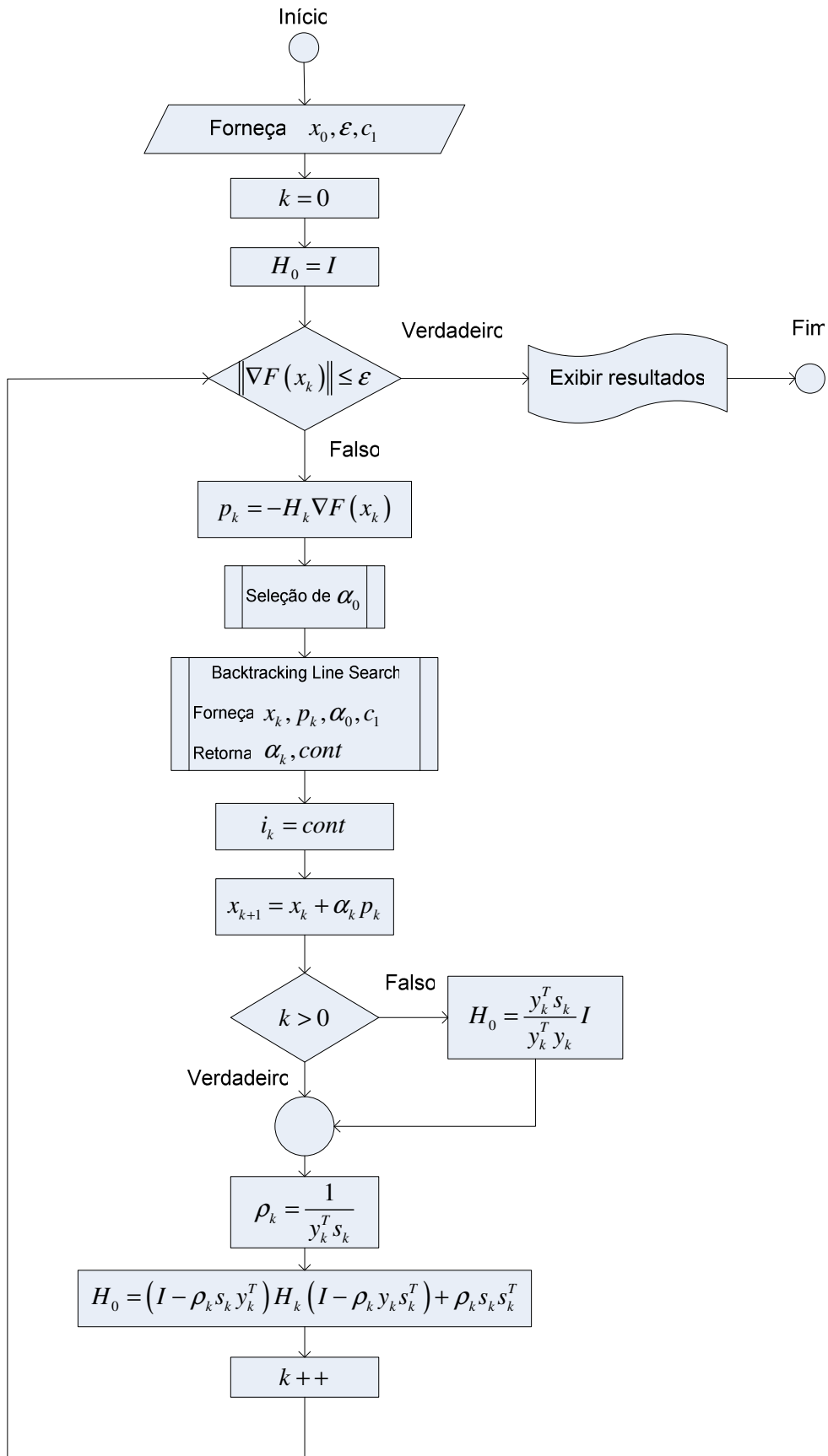


Figura 3-2: Fluxograma do algoritmo do Método Quase-Newton.

3.3 Método Newton Gradiente Conjugado

A direção de Newton pode ser inadequada para problemas reais, pois o algoritmo pode não convergir para soluções distantes dos pontos iniciais. Também pode apresentar um comportamento errático nas regiões em que a função objetivo não é convexa, o que significa que quando a matriz Hessiana não é definida positiva, ou quase singular, a direção pode ser de subida ou muito grande.

Variante do método de Newton visou contornar tais problemas. O objetivo neste trabalho é apresentar a versão chamada de método de Newton inexato (Newton Gradiente Conjugado). Nesta versão calcula-se uma direção de busca aproximada resolvendo de maneira inexata o seguinte sistema linear:

$$\nabla^2 f(x_k) p_k = -\nabla f(x_k) \quad (3-3)$$

Essa aproximação é válida, pois o modelo quadrático pode não ser uma boa aproximação da função objetivo em torno do ponto x_k .

O algoritmo apresenta altas taxas de convergência, tornando-se quadrática quando a região de convergência é encontrada. Os maiores gastos computacionais são decorrentes do cálculo da matriz Hessiana.

A realização prática do método é apresentada em fluxograma na Figura 3-3. O algoritmo para cálculo da direção de busca será explicado na Seção 3.4.3. A listagem do código implementado em *MatLab* é apresentada no Apêndice IV.

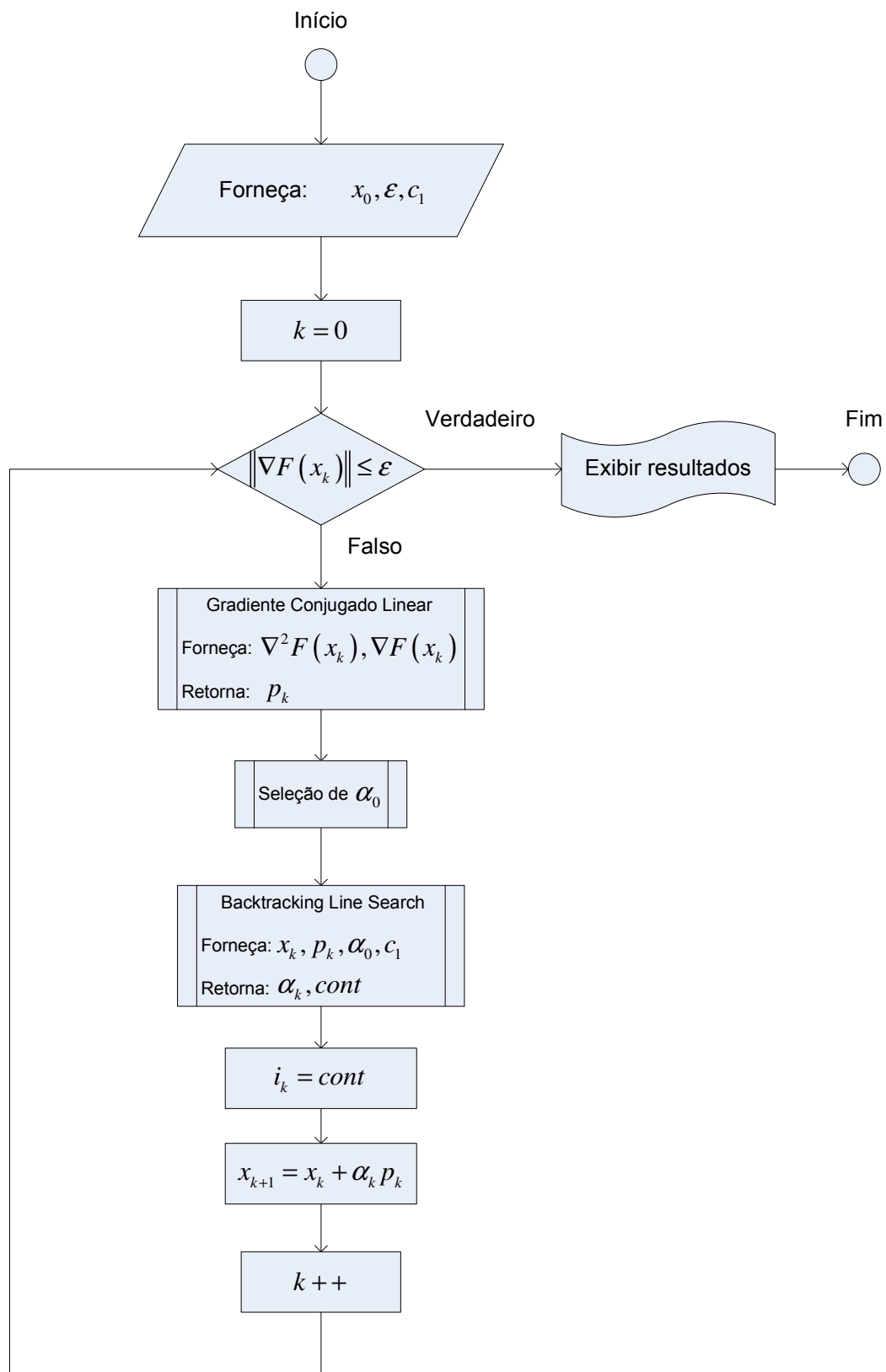


Figura 3-3: Fluxograma do algoritmo do Método Newton Gradiente Conjugado.

3.4 Algoritmos Auxiliares

Para facilitar a implementação dos algoritmos separaram-se em rotinas específicas as instruções que são utilizadas em todos os algoritmos. No caso do método de Newton-GC, o algoritmo para o cálculo da direção de busca será apresentado de forma detalhada para facilitar a compreensão do método.

3.4.1 Seleção do Passo Inicial de Busca α_0

A seleção do passo inicial de busca é importante etapa na determinação do passo de busca, pois de acordo com o critério escolhido pode-se evitar a realização de cálculos desnecessários para a convergência do método de otimização escolhido.

As formas utilizadas para o cálculo de α_0 , de acordo com [1], são apresentadas na Figura 3-4. O primeiro método é simplesmente o valor unitário, o segundo é a interpolação de uma quadrática a partir de f_k , f_{k-1} e $\phi'(0) = \nabla f_k^T p_k$, definindo α_0 como o respectivo minimizador. O terceiro é consequência da suposição de que a mudança no termo de primeira de ordem é a mesma da iteração anterior.

<u>Método 1</u>	<u>Método 2</u>	<u>Método 3</u>
$\alpha_0 = 1$	$\alpha_0 = \frac{2(f_k - f_{k-1})}{\nabla f_k^T p_k}$	$\alpha_0 = \alpha_{k-1} \frac{\nabla f_{k-1}^T p_{k-1}}{\nabla f_k^T p_k}$

Figura: 3-4: Métodos de seleção de α_0 .

Neste caso, não foi implementada nenhuma rotina específica, sendo as instruções colocadas diretamente no código. Na apresentação dos resultados será dada uma explicação sobre qual método de seleção estará em uso e do desempenho dos métodos de otimização para os diferentes métodos de seleção do passo inicial de busca.

3.4.2 Cálculo do Passo de Busca: “Backtracking Line Search”

O cálculo da direção de busca é realizado na rotina *backtracking line search* e acontece da seguinte maneira: o passo de busca deve atender a condição de decréscimo suficiente e ser atualizado por meio de uma interpolação quadrática.

A condição de decréscimo suficiente não garante que o algoritmo realizará progresso suficiente numa dada direção de busca. No entanto, utilizando algoritmos de *backtracking* é possível ignorar a condição de curvatura e obter um decréscimo apropriado da função objetivo na direção de busca.

A atualização por meio de uma interpolação quadrática poderá evitar uma grande quantidade de iterações (no algoritmo do *backtracking*), quando comparada à atualização pelo método linear proposto em [1]. Um nome mais adequado para o algoritmo de busca implementado seria “Backtracking Quadratic Search”. Uma visão geral do método é apresentada em forma de fluxograma na Figura 3-5.

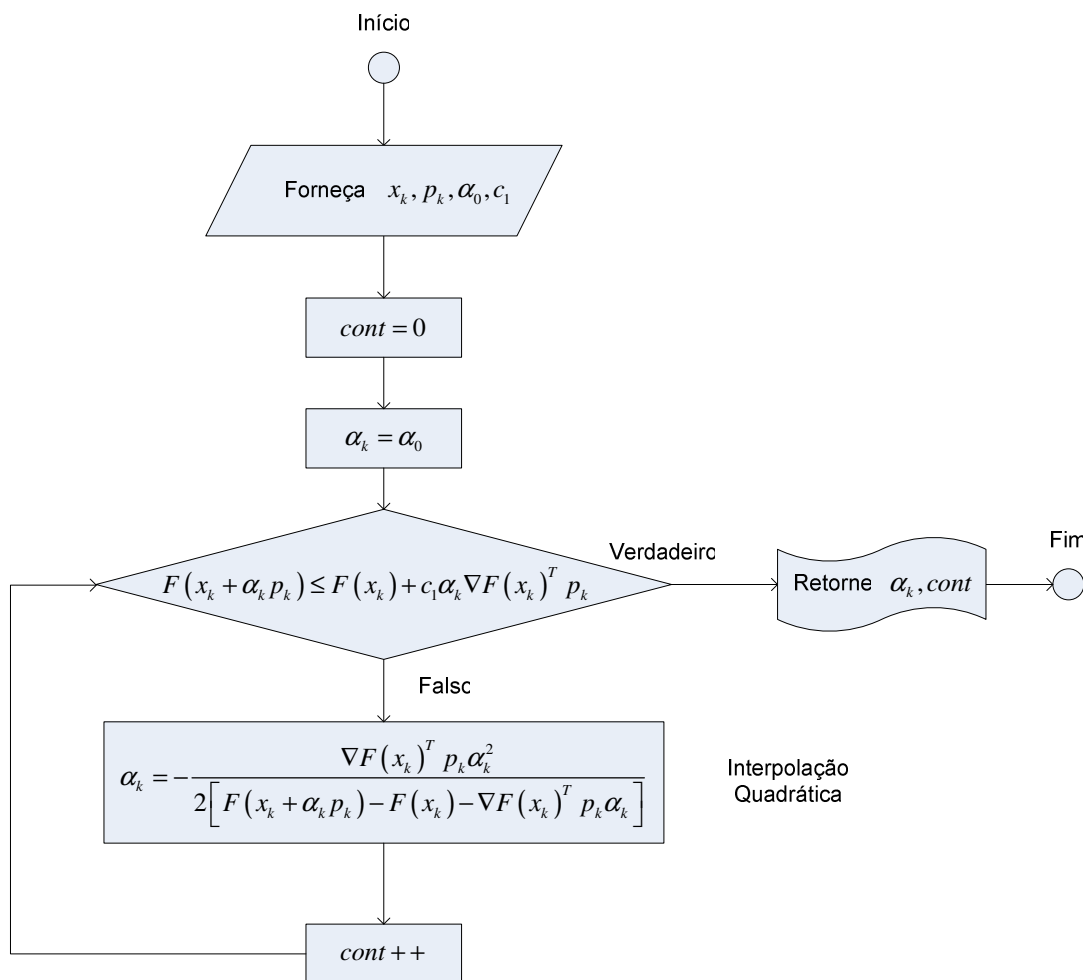


Figura 3-5: Fluxograma do Algoritmo “Backtracking Line Search”.

3.4.3 Direção de Busca pelo Método do Gradiente Conjugado Linear

A solução inexata da Equação 3-3 é feita pelo método do Gradiente Conjugado Linear (GCL). Entretanto, o método do GCL é um método de solução de sistemas lineares definidos positivos e há situações em que a matriz Hessiana pode ter autovalores negativos fazendo com que a solução do problema não seja encontrada. Então para tornar possível a implementação do método do GCL, verifica-se toda iteração se alguma curvatura negativa é encontrada. Essa adaptação garante que uma direção de busca é de descida e que a convergência rápida do Método de Newton é mantida.

Em linhas gerais pode-se sumarizar a adaptação do algoritmo do GCL da seguinte maneira:

- O ponto inicial para o GCL é o ponto $z_0 = 0$;
- Teste de curvatura negativa. Se a direção de busca do GCL satisfaz

$$p_n^T A p_n \leq 0$$

então:

- Se a iteração atual é $n=0$, calcula-se o ponto z_1 com esta direção e se interrompe o processo (*stop*). Isto significa que o algoritmo tem que andar para pelo menos algum lugar, mas como neste ponto só tem como seguir uma direção de subida, anda-se o mínimo possível nesta direção. Uma estratégia para evitar isso é fazer o ponto inicial igual a zero ($z_0 = 0$) obtendo $p_0 = -\nabla F(x_k)$, que é uma direção de descida.
 - Se $n>0$, para o algoritmo imediatamente com a direção calculada mais recente. Pois, como a última direção foi calculada com curvatura positiva significa que esta última é uma boa direção de descida.
- O passo de Newton é definido com o resultado final z_n do GCL.

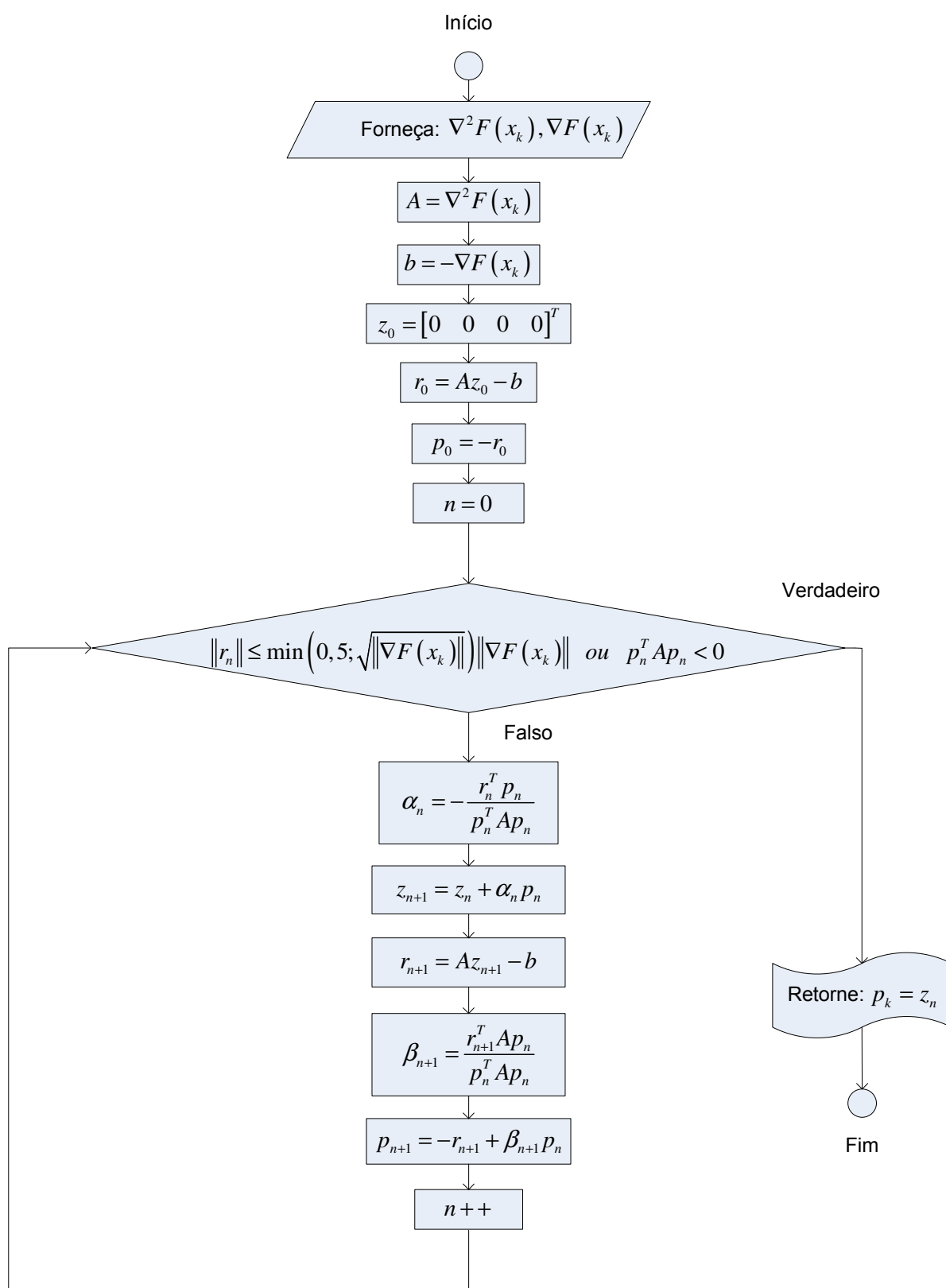


Figura 3-6: Fluxograma do algoritmo para o cálculo da direção de busca via GCL.

4 Resultados

Os resultados obtidos da aplicação dos métodos de otimização irrestrita na Equação 2-1 serão apresentados neste capítulo.

Primeiramente será analisado o melhor método de escolha do passo inicial. A partir dos resultados obtidos serão analisadas as questões específicas aos métodos Quase-Newton e Newton Gradiente Conjugado. No método Quase-Newton será avaliado o impacto de analisar a condição de curvatura como critério para atualização da aproximação da matriz Hessiana. E, para o método de Newton-GC será analisada a convergência para diferentes parâmetros da seqüência forçante utilizado para o cálculo da direção de busca pelo método do GCL.

Após essas análises, os métodos são configurados com os parâmetros que proporcionaram uma maior velocidade. Então, os desempenhos dos métodos são comparados para avaliar o melhor desempenho computacional.

4.1 Análise da Escolha do Passo Inicial de Busca α_0

4.1.1 Método do Gradiente

4.1.1.1 Ponto Inicial $x_0 = [-1 \ -2 \ 1 \ -1]^T$

Para o método do gradiente recomenda-se em [1] a escolha do passo inicial de busca a partir de informações de iterações passadas. Como apresentado na seção 3-4-2 foram implementados três métodos de para a escolha de α_0 , dentre os quais é dada uma maior atenção para o segundo e o terceiro método, sendo o primeiro utilizado apenas para efeito comparativo.

Partindo do ponto inicial $x_0 = [-1 \ -2 \ 1 \ -1]^T$ e com uma tolerância de 1×10^{-4} são obtidos os resultados apresentados na Tabela 4-1. Infere-se da Tabela 4-1 que a melhor escolha é a número 3 por apresenta o menor número de iterações e menor tempo de execução. Também, observa-se que os demais resultados convergem para valores semelhantes.

Tabela 4-1: Resultados para $x_0 = [-1 \ -2 \ 1 \ -1]^T$ e $\epsilon = 1 \times 10^{-4}$

Método de Escolha do Passo Inicial	Nº de Iterações	Nº de Interpolações	Nº Máximo de Interpolações	Tempo [s]	Norma	Custo	x(1)	x(2)	x(3)	x(4)
1	3525	3525	1	8,31	9,993E-05	5,029E-03	-3,5246	-5,7601	1,8393	-1,8438
2	3473	9	1	7,28	9,869E-05	5,029E-03	-3,5244	-5,7597	1,8393	-1,8438
3	3424	22	1	6,99	9,672E-05	5,029E-03	-3,5238	-5,7588	1,8389	-1,8433

É importante observar também o comportamento numérico de algumas variáveis durante o processo iterativo. Uma maneira interessante é o comportamento gráfico das variáveis.

Na Figura 4-1, na Figura 4-2 e na Figura 4-3 são visualizados os comportamentos dos α 's durante o processo iterativo. É possível visualizar o comportamento oscilatório dos α 's nas iterações finais, uma característica que pode tornar o método do gradiente excessivamente lento em problemas mal condicionados. Os pontos pretos nas figuras indicam as iterações que ocorreram interpolações (na Figura 4-1 ocorreu em todas as iterações).

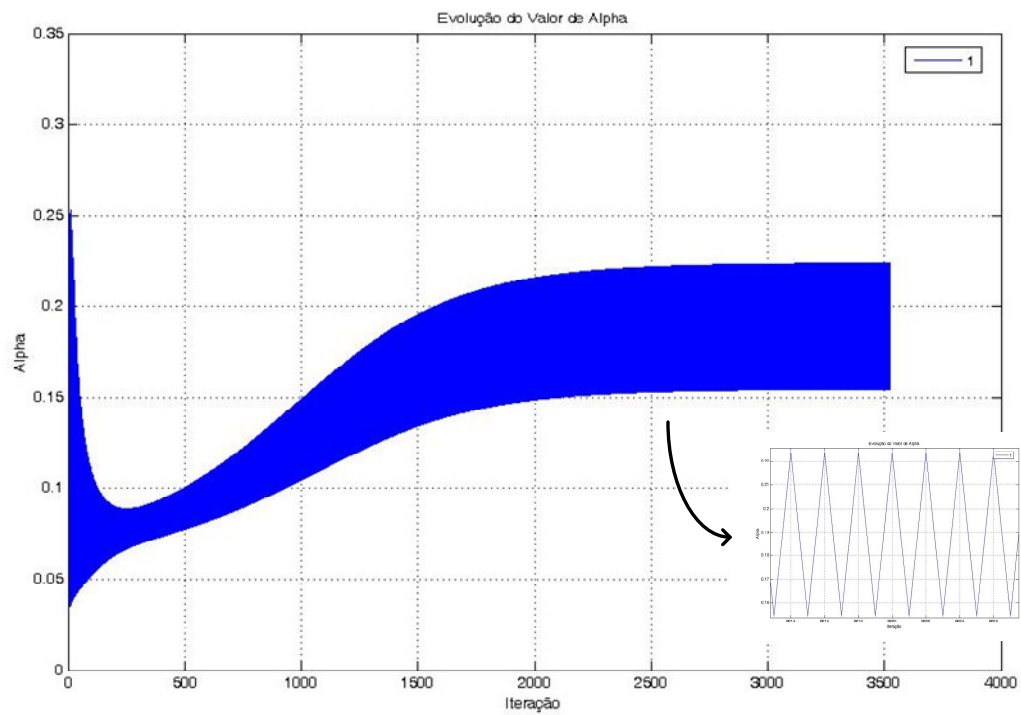


Figura 4-1: Evolução de α para a escolha número 1.

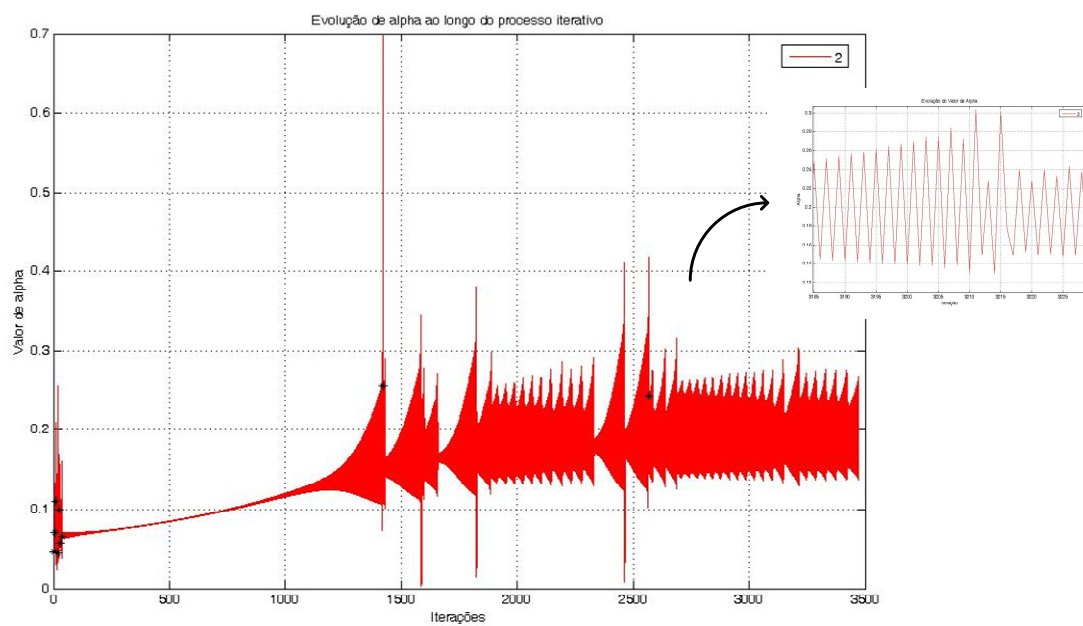


Figura 4-2: Evolução de α para a escolha número 2.

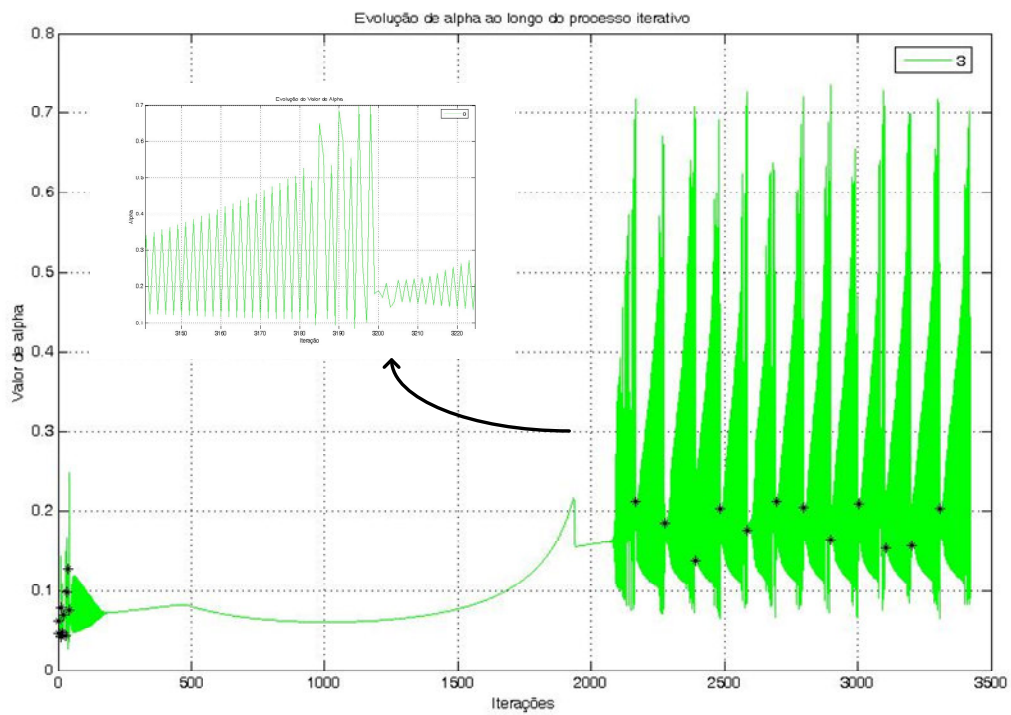


Figura 4-3: Evolução de α para a escolha número 3.

A evolução de x_k é bastante semelhante para todos os métodos de escolha, não se fazendo necessário apresentar três figuras. O gráfico com a evolução de x_k é apresentado na Figura 4-4.

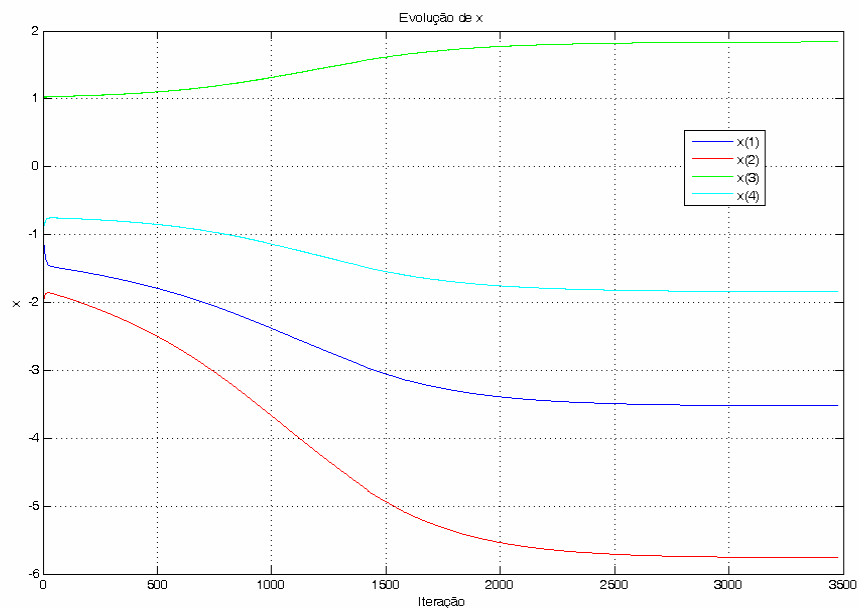


Figura 4-4: Evolução de x_k para a escolha número 2.

Outros resultados de interesse são: a norma 2 do gradiente (critério de convergência do método), o custo da função objetivo (idealmente deve ser zero) e o teste de aderência da curva obtida com os dados experimentais.

No mesmo gráfico é mostrada a evolução da norma para os diferentes métodos de escolha. Observa-se na Figura 4-5 comportamentos semelhantes. Note o movimento oscilatório nas iterações finais, característico do método do gradiente.

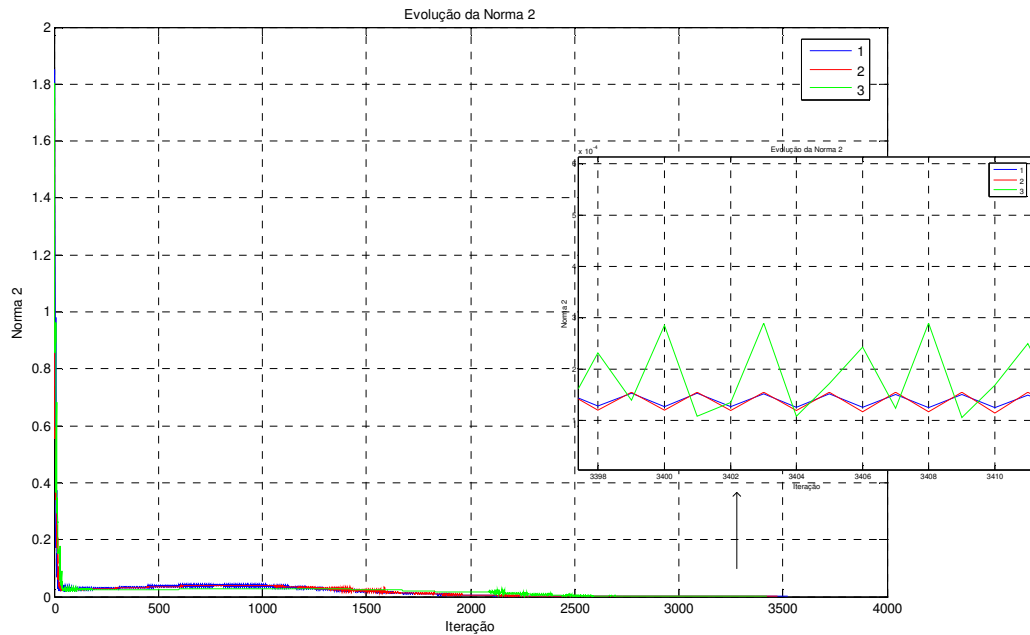


Figura 4-5: Evolução de $\|\nabla F(x_k)\|$.

De forma semelhante à norma do gradiente, é apresentada a evolução do custo da função objetivo na Figura 4-6.

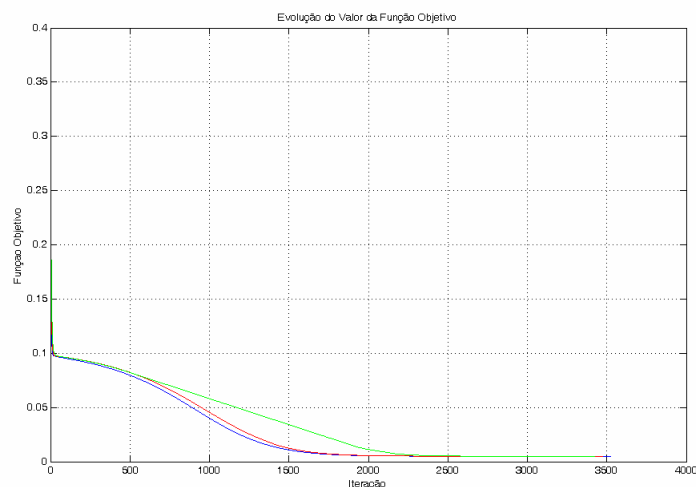


Figura 4-6: Evolução de $F(x_k)$.

O resultado de maior interesse prático é o teste de aderência das curvas ajustadas com os dados experimentais. Os ajustes obtidos para as diferentes escolhas são apresentados na Figura 4-7. As curvas dos três métodos se sobrepõem e proporcionam um ajuste adequado.

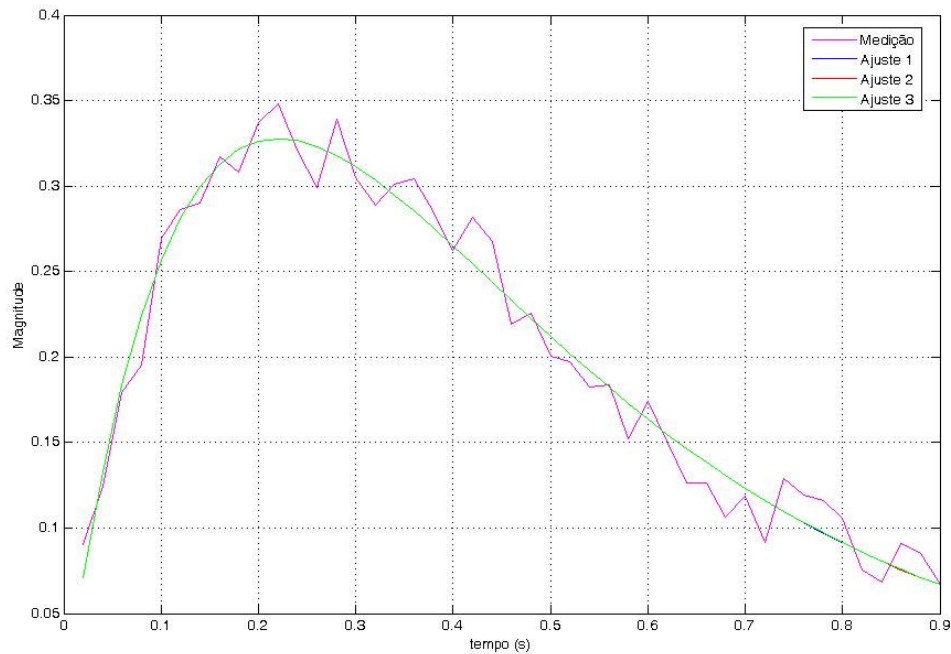


Figura 4-7: Teste de aderência para os pontos obtidos para as diferentes escolhas.

4.1.1.2 Outros Pontos Iniciais

Mantendo a tolerância em $\varepsilon=1 \times 10^{-4}$, dois novos pontos iniciais foram testados $x_0 = [-2 \ -2 \ 2 \ -2]^T$ e $x_0 = [0 \ 0 \ 0 \ 0]^T$, e os resultados são apresentados na Tabela 4-2 e Tabela 4-3, respectivamente.

Tabela 4-2: Resultados para $x_0 = [-2 \ -2 \ 2 \ -2]^T$ e $\varepsilon=1 \times 10^{-4}$

Método de Escolha do Passo Inicial	Nº de Iterações	Nº de Interpolações	Nº Máximo de Interpolações	Tempo [s]	Norma	Custo	x(1)	x(2)	x(3)	x(4)
1	3060	3060	1	6,41	9,952E-05	5,008E-03	-3,6954	-5,4511	2,3104	-2,3126
2	2920	8	1	5,19	9,927E-05	5,008E-03	-3,6949	-5,4516	2,3090	-2,3112
3	2299	20	1	4,16	9,990E-05	5,008E-03	-3,6949	-5,4506	2,3099	-2,3120

Tabela 4-3: Resultados para $x_0 = [0 \ 0 \ 0 \ 0]^T$ e $\varepsilon = 1 \times 10^{-4}$

Método de Escolha do Passo Inicial	Nº de Iterações	Nº de Interpolações	Nº Máximo de Interpolações	Tempo [s]	Norma	Custo	x(1)	x(2)	x(3)	x(4)
1	2213	2212	1	3,30	9,989E-05	1,063E-01	-0,9698	-0,9698	0,1544	0,1544
2	1592	13	3	1,70	9,970E-05	1,063E-01	-0,9698	-0,9698	0,1544	0,1544
3	1387	22	3	1,35	9,353E-05	1,063E-01	-0,9697	-0,9697	0,1544	0,1544

Da Tabela 4-2 podem-se inferir conclusões semelhantes ao caso anterior, valendo notar que a diferença de tempo e de número de iterações obtidas pela escolha número 3 foi mais expressiva.

O teste de aderência com as curvas é apresentado na Figura 4-8. Como no caso anterior as curvas se sobrepõem e proporcionam um ajuste adequado.

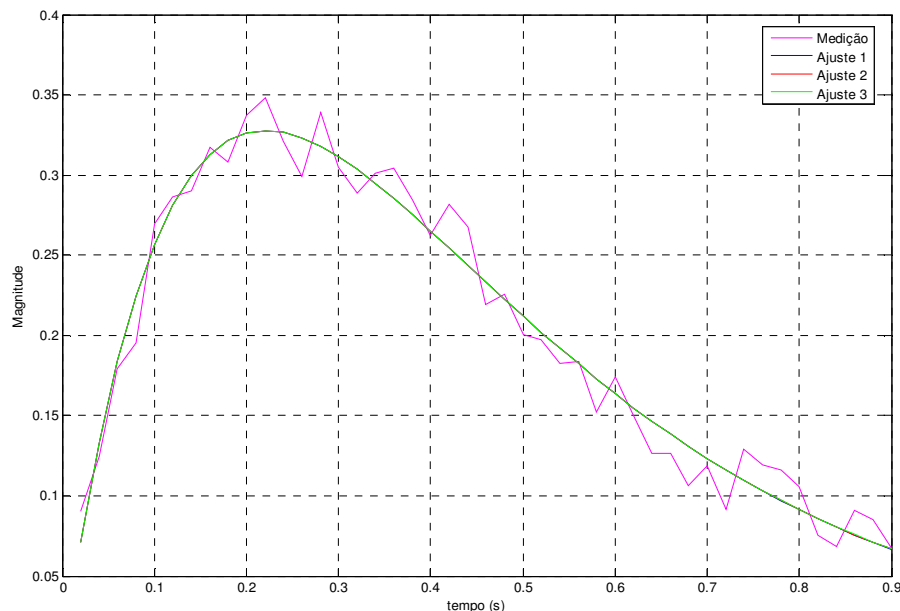


Figura 4-8: Teste de aderência para os pontos obtidos para as diferentes escolhas com $x_0 = [-2 \ -2 \ 2 \ -2]^T$.

Embora nos dados da Tabela 4-3 a performance da escolha número 3 continue sendo a melhor, observa-se na Figura 4-9 que as curvas geradas pelos pontos obtidos na otimização não passam no teste de aderência. As curvas não representam o comportamento da concentração da droga no sangue em função do tempo.

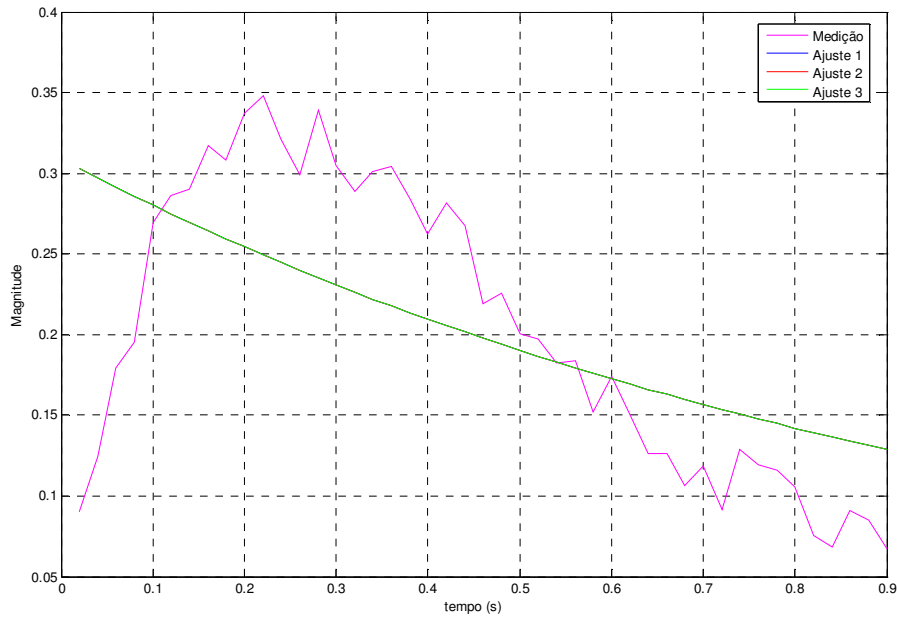


Figura 4-9: Teste de aderência para os pontos obtidos para as diferentes escolhas com $x_0 = [0 \ 0 \ 0 \ 0]^T$.

4.1.1.3 Conclusões

Para os pontos $x_0 = [-1 \ -2 \ 1 \ -1]^T$ e $x_0 = [-2 \ -2 \ 2 \ -2]^T$ foram obtidos como solução final os pontos $x_0 = [-3,52 \ -5,76 \ 1,84 \ -1,84]^T$ e $x_0 = [-3,69 \ -5,45 \ 2,31 \ -2,31]^T$, respectivamente. Embora sejam pontos distintos de \mathbb{R}^4 proporcionam valores finais de norma e custo semelhantes, e ajustes adequados como mostrado na Figura 4-7 e na Figura 4-8. Isso se deve à existência de vários pontos de mínimos locais para a função objetivo não-linear do problema.

De acordo com as análises do método do gradiente conclui-se que a melhor escolha do passo inicial de busca é a número 3. Portanto, a partir desse momento quando o método do gradiente voltar a ser utilizado será com o passo inicial escolhido dessa maneira.

4.1.2 Método Quase-Newton

4.1.2.1 Ponto Inicial $x_0 = [-1 \ -2 \ 1 \ -1]^T$

De forma semelhante ao método do gradiente será analisada a escolha que proporcione o melhor desempenho para o método Quase-Newton. Segundo [1] a melhor escolha para métodos baseados na direção de Newton é número 1.

Os resultados são apresentados na Tabela 4-4 para uma tolerância de 1×10^{-6} . Contradizendo a referência a melhor escolha é a número 2 por apresenta o menor número de iterações e menor tempo de execução. Com a escolha número 3 o algoritmo não converge ou executa mais que dez mil iterações, o que a torna inviável computacionalmente.

Tabela 4-4: Resultados para $x_0 = [-1 \ -2 \ 1 \ -1]^T$ e $\epsilon = 1 \times 10^{-6}$

Método de Escolha do Passo Inicial	Nº de Iterações	Nº de Interpolações	Nº Máximo de Interpolações	Tempo [s]	Norma	Custo	x(1)	x(2)	x(3)	x(4)
1	195	56	1	0,27	2,048E-07	5,000E-03	-4,0000	-5,0000	4,0002	-4,0002
2	184	37	2	0,26	3,961E-07	5,000E-03	-3,9999	-5,0001	3,9990	-3,9990
3	>10000	-	-	-	-	-	-	-	-	-

Os comportamentos dos α 's durante o processo iterativo são mostrados na Figura 4-10 e na Figura 4-11. Os pontos pretos indicam as iterações que ocorreram interpolações.

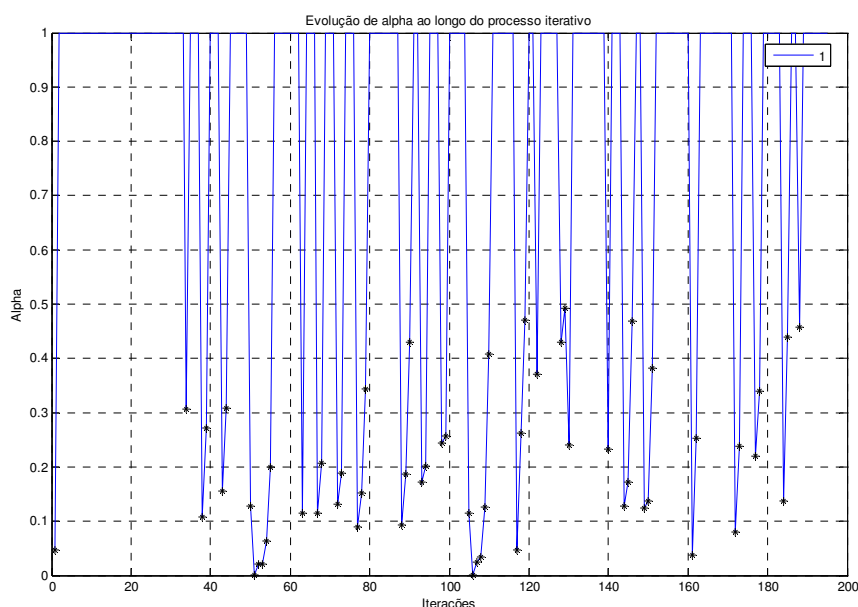


Figura 4-10: Evolução de α para a escolha número 1.

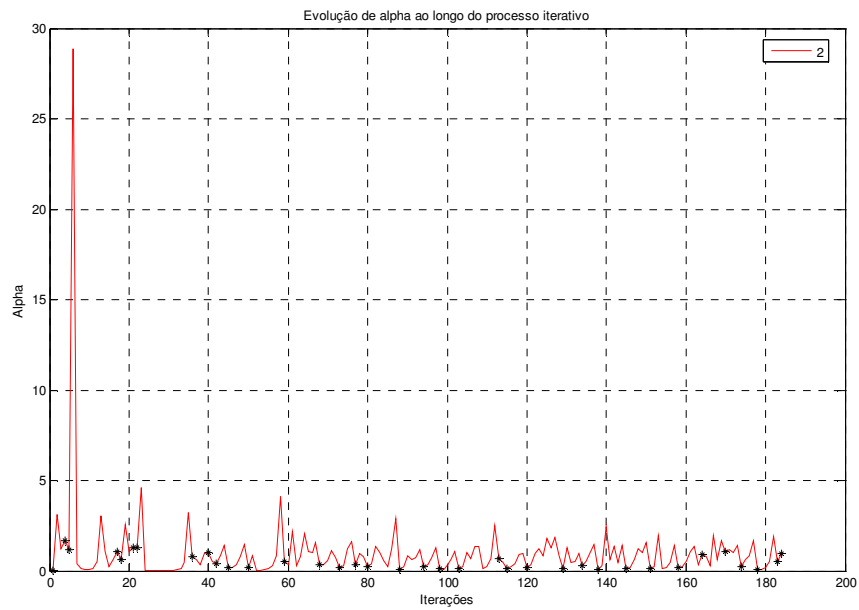


Figura 4-12: Evolução de α para a escolha número 2.

As demais variáveis apresentam uma evolução bastante semelhantes ao longo do processo iterativo, para ambos os métodos de escolha. Na Figura 4-13 tem-se o comportamento de x_k .

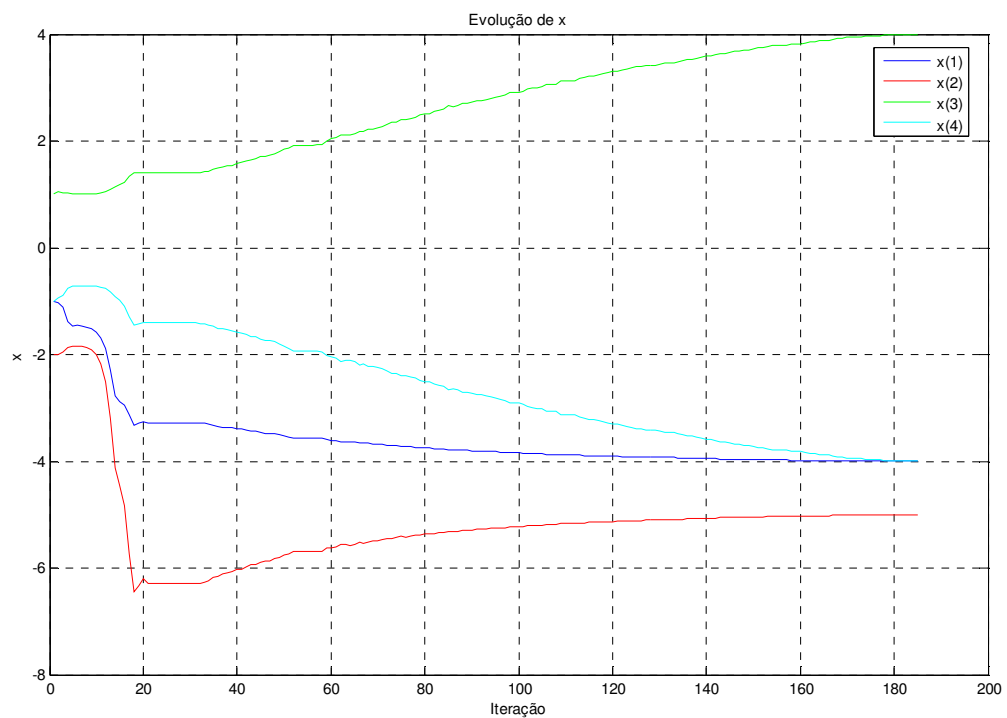


Figura 4-13: Evolução de x_k para a escolha número 2.

No primeiro momento a norma de $F(x_k)$ decresce juntamente com o valor da função objetivo, em seguida a uma elevação do seu valor até que os valores comecem a convergir para a solução final. Estes comportamentos estão ilustrados na Figura 4-14 e na Figura 4-15.

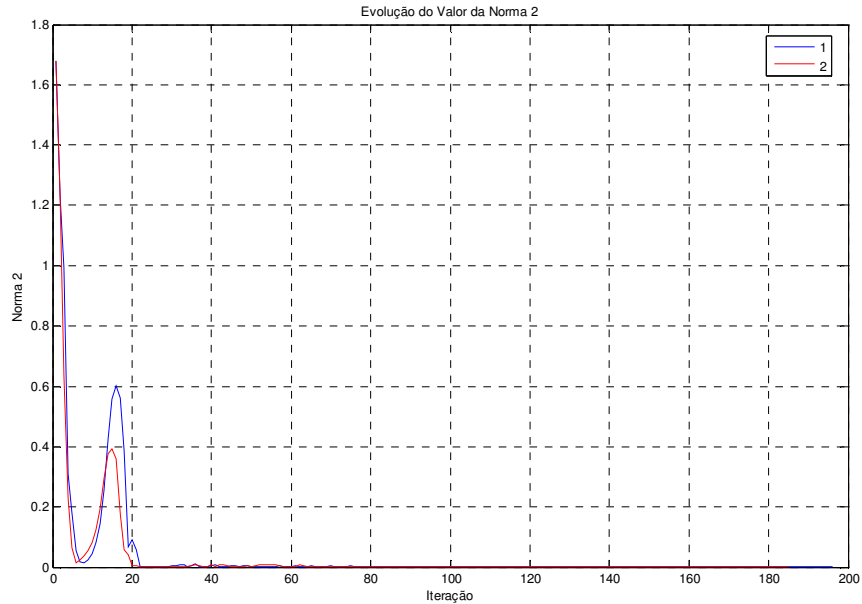


Figura 4-14: Evolução de $\|\nabla F(x_k)\|$.

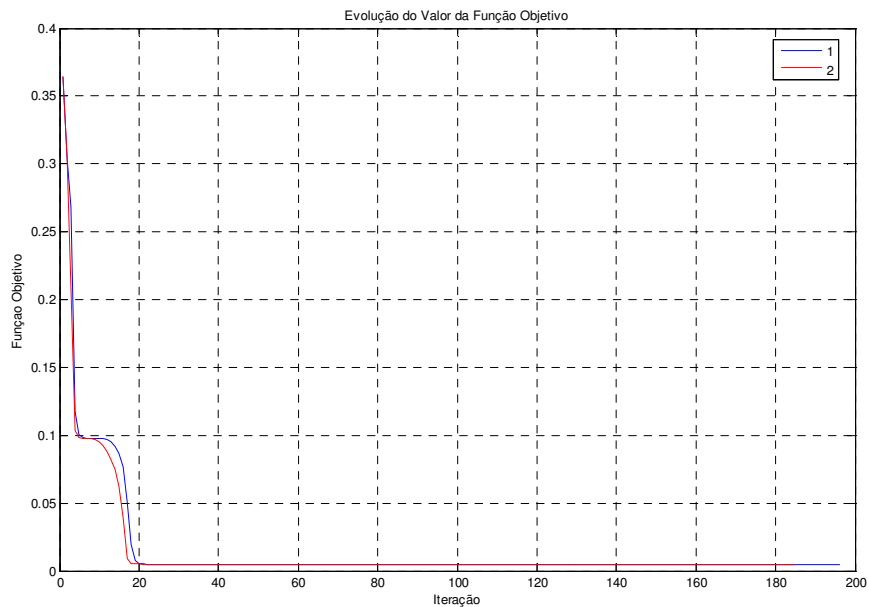


Figura 4-15: Evolução de $F(x_k)$.

Novamente, as curvas se sobrepõem e proporcionam um ajuste adequado, como pode ser visualizado na Figura 4-16.

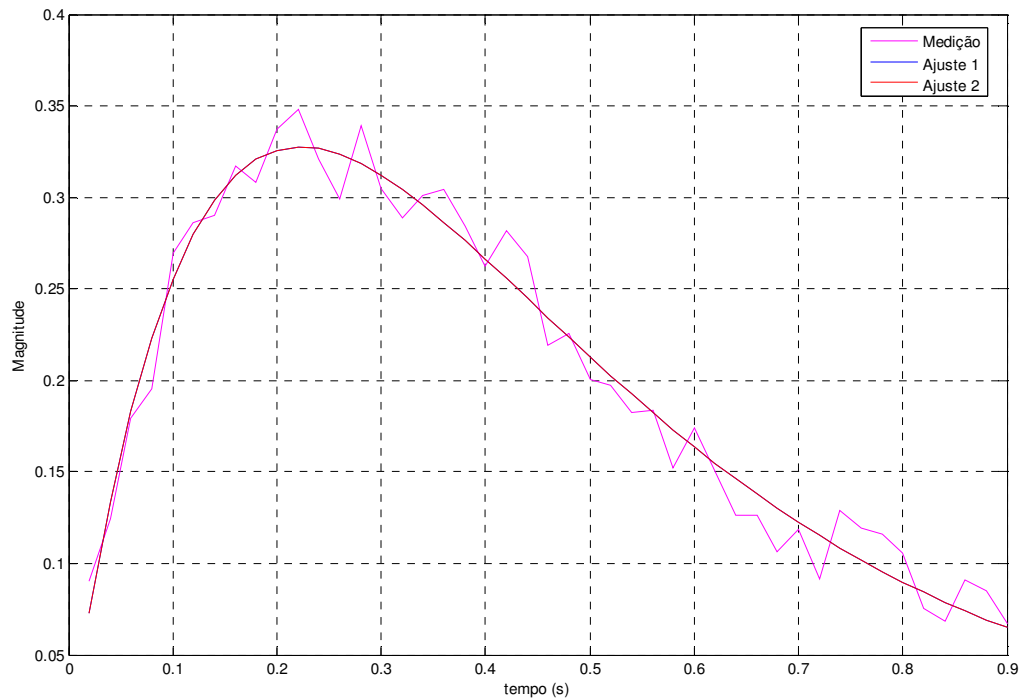


Figura 4-16: Teste de aderência para os pontos obtidos para as diferentes escolhas.

4.1.2.2 Outros Pontos Iniciais

Para os pontos iniciais $x_0 = [-2 \ -2 \ 2 \ -2]^T$ e $x_0 = [0 \ 0 \ 0 \ 0]^T$, com tolerância em $\varepsilon = 1 \times 10^{-6}$ os resultados são apresentados na Tabela 4-5 e Tabela 4-6, respectivamente.

Tabela 4-5: Resultados para $x_0 = [-2 \ -2 \ 2 \ -2]^T$ e $\varepsilon = 1 \times 10^{-6}$

Método de Escolha do Passo Inicial	Nº de Iterações	Nº de Interpolações	Nº Máximo de Interpolações	Tempo [s]	Norma	Custo	x(1)	x(2)	x(3)	x(4)
1	2815	2693	3	6,48	1,921E-07	5,000E-03	-4,0000	-5,0000	4,0002	-4,0002
2	158	32	1	0,24	1,155E-07	5,000E-03	-3,9998	-5,0003	3,9978	-3,9978
3	>10000	-	-	-	-	-	-	-	-	-

Tabela 4-6: Resultados para $x_0 = [0 \ 0 \ 0 \ 0]^T$ e $\varepsilon = 1 \times 10^{-6}$

Método de Escolha do Passo Inicial	Nº de Iterações	Nº de Interpolações	Nº Máximo de Interpolações	Tempo [s]	Norma	Custo	x(1)	x(2)	x(3)	x(4)
1	11	1	1	0,11	1,577E-07	1,063E-01	-0,9705	-0,9705	0,1545	0,1545
2	10	9	3	0,10	6,188E-08	1,063E-01	-0,9705	-0,9705	0,1545	0,1545
3	9	10	3	0,07	1,031E-09	1,063E-01	-0,9705	-0,9705	0,1545	0,1545

Dos dados da Tabela 4-5 podem-se inferir conclusões semelhantes ao caso anterior, valendo notar que a diferença de tempo e de número de iterações obtidas pela escolha número 2 foi mais expressiva. Como a solução final partindo $x_0 = [-2 \ -2 \ 2 \ -2]^T$ é a mesma com $x_0 = [-1 \ -2 \ 1 \ -1]^T$, o ajuste é igual ao da Figura 4-16.

Partindo com $x_0 = [0 \ 0 \ 0 \ 0]^T$ o ajuste não é adequado e semelhante ao da Figura 4-9.

4.1.2.3 Conclusões

De acordo com as análises do método Quase-Newton conclui-se que a melhor escolha do passo inicial de busca é a número 2. Portanto, a partir desse momento quando o método do gradiente voltar a ser utilizado será com o passo inicial escolhido dessa maneira.

4.1.3 Método Newton Gradiente Conjugado

4.1.3.1 Ponto Inicial $x_0 = [-1 \ -2 \ 1 \ -1]^T$

Os resultados são apresentados na Tabela 4-7 para uma tolerância de 1×10^{-6} . Contradizendo a referência a melhor escolha é a número 2 por apresentar o menor número de iterações e menor tempo de execução.

Tabela 4-7: Resultados para $x_0 = [-1 \ -2 \ 1 \ -1]^T$ e $\epsilon = 1 \times 10^{-6}$

Método de Escolha do Passo Inicial	Nº de Iterações	Nº de Interpolações	Nº Máximo de Interpolações	Tempo [s]	Norma	Custo	x(1)	x(2)	x(3)	x(4)
1	681	641	1	1,40	8,157E-07	5,000E-03	-3,9282	-5,0993	3,4253	-3,4257
2	176	37	1	0,38	9,247E-07	5,000E-03	-3,9273	-5,1006	3,4188	-3,4193
3	1452	10	1	2,89	6,252E-07	5,000E-03	-3,9424	-5,0793	3,5262	-3,5266

Os comportamentos dos α 's durante o processo iterativo são mostrados na Figura 4-17, na Figura 4-18 e na Figura 4-19. Os pontos pretos indicam as iterações que ocorreram interpolações.

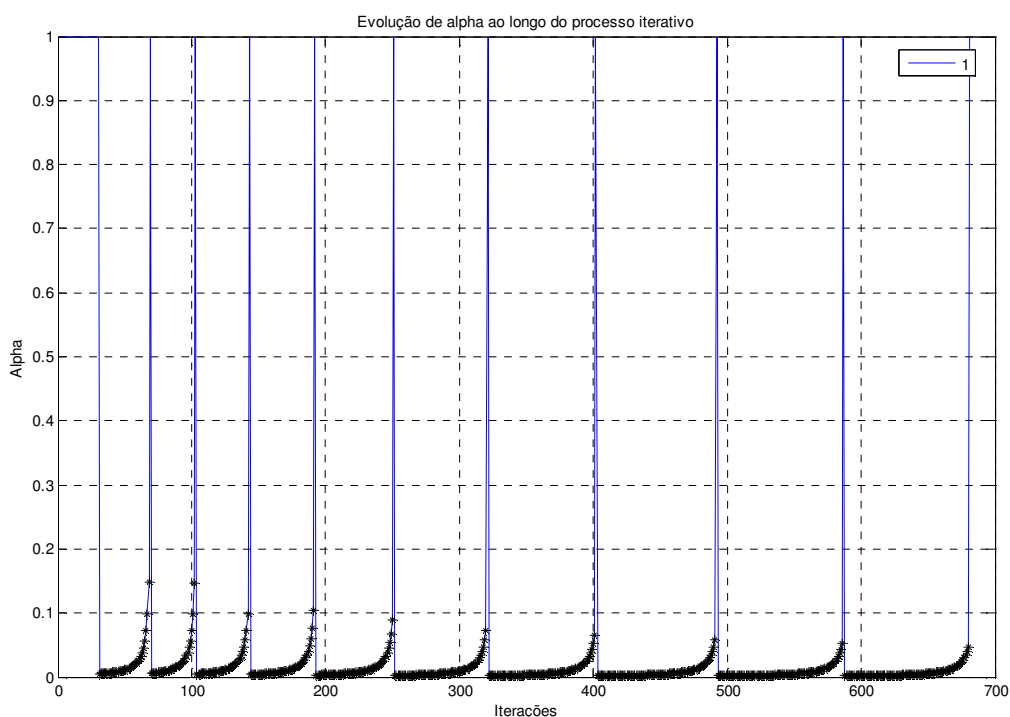


Figura 4-17: Evolução de α para a escolha número 1.

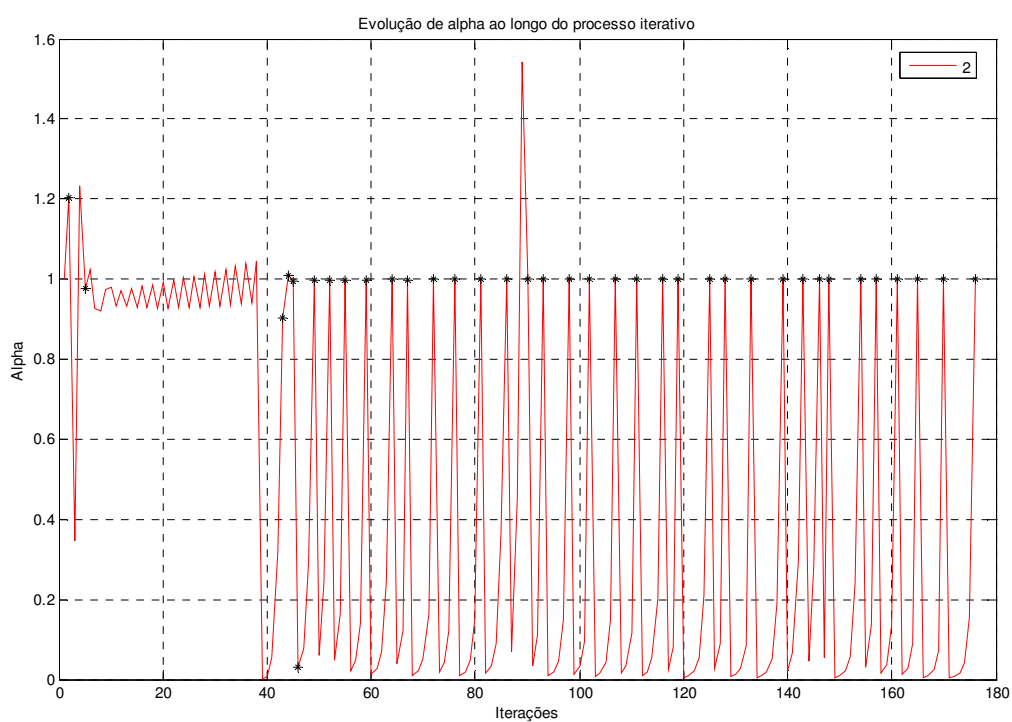


Figura 4-18: Evolução de α para a escolha número 2.

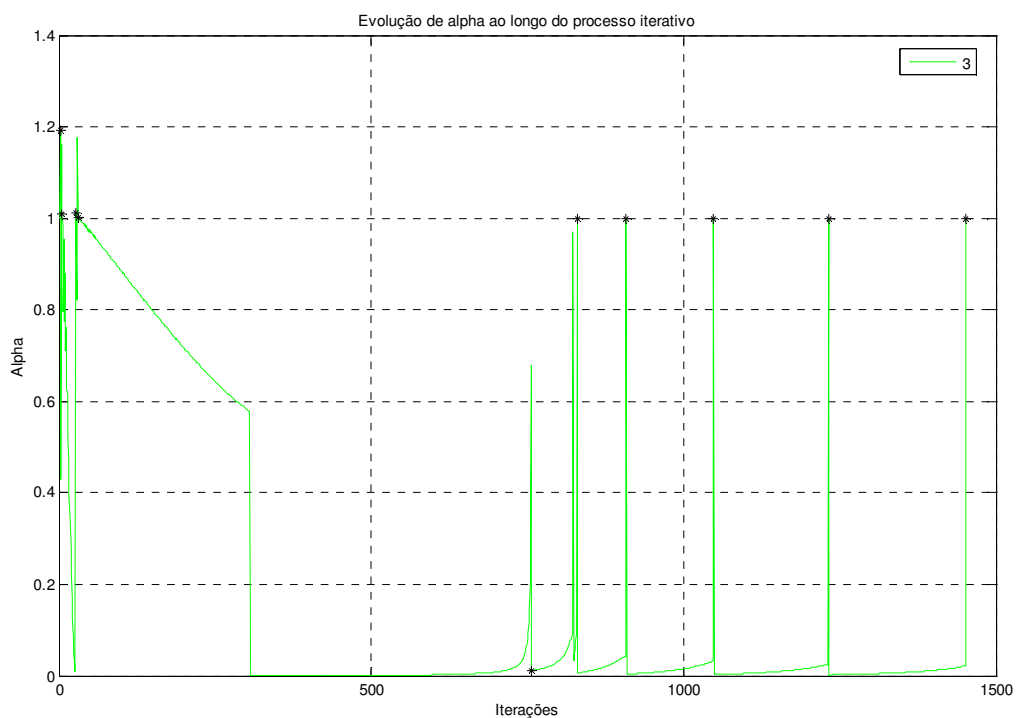


Figura 4-19: Evolução de α para a escolha número 3.

Para as escolhas 1 e 2 o comportamento de x_k é bastante semelhante e pode ser visualizado na Figura 4-20 e na Figura 4-21.

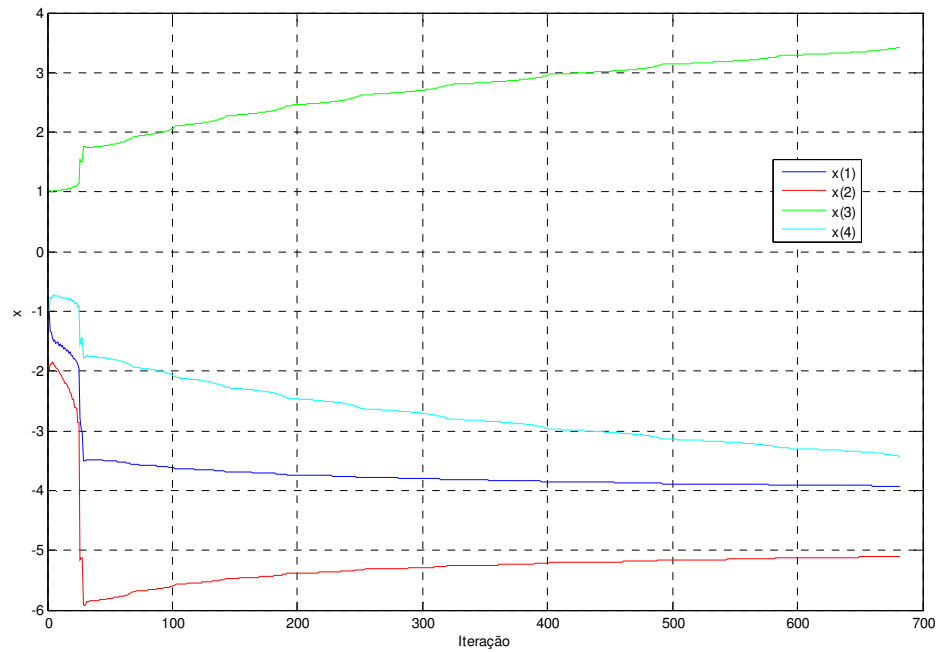


Figura 4-20: Evolução de x_k para a escolha número 1.

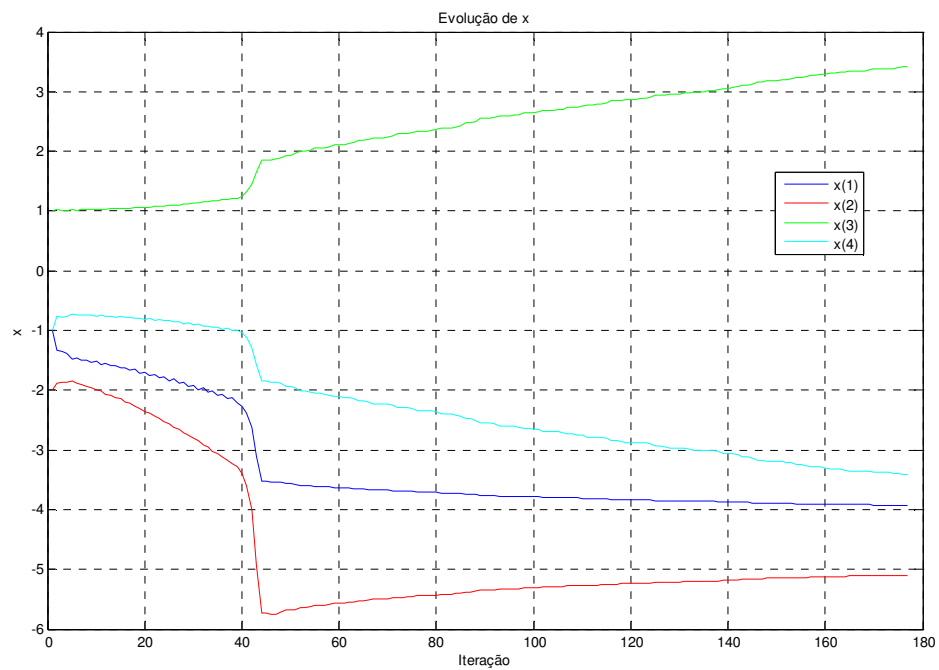


Figura 4-21: Evolução de x_k para a escolha número 2.

Com a escolha 3 a evolução de x_k é bastante lenta, tal comportamento pode ser atribuído aos baixos valores de α durante o processo (Figura 4-19).

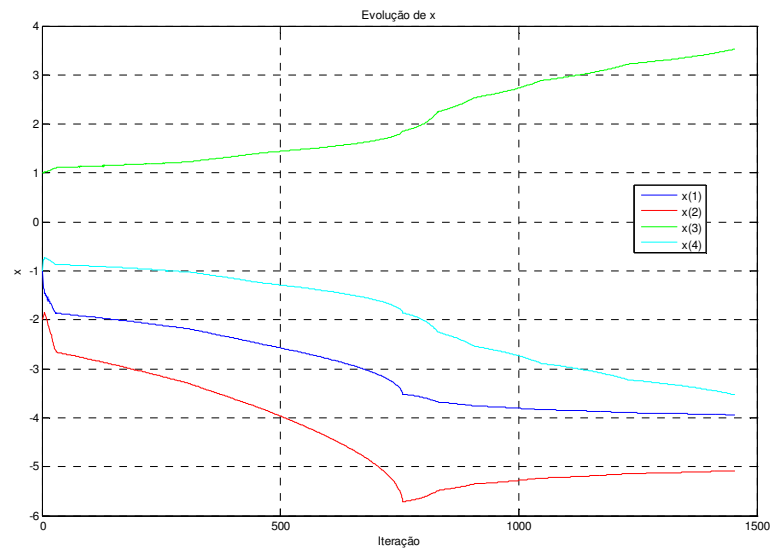


Figura 4-22: Evolução de x_k para a escolha número 3.

No mesmo gráfico é mostrada a evolução da norma para os diferentes métodos de escolha. Observa-se na Figura 4-23 os comportamentos.

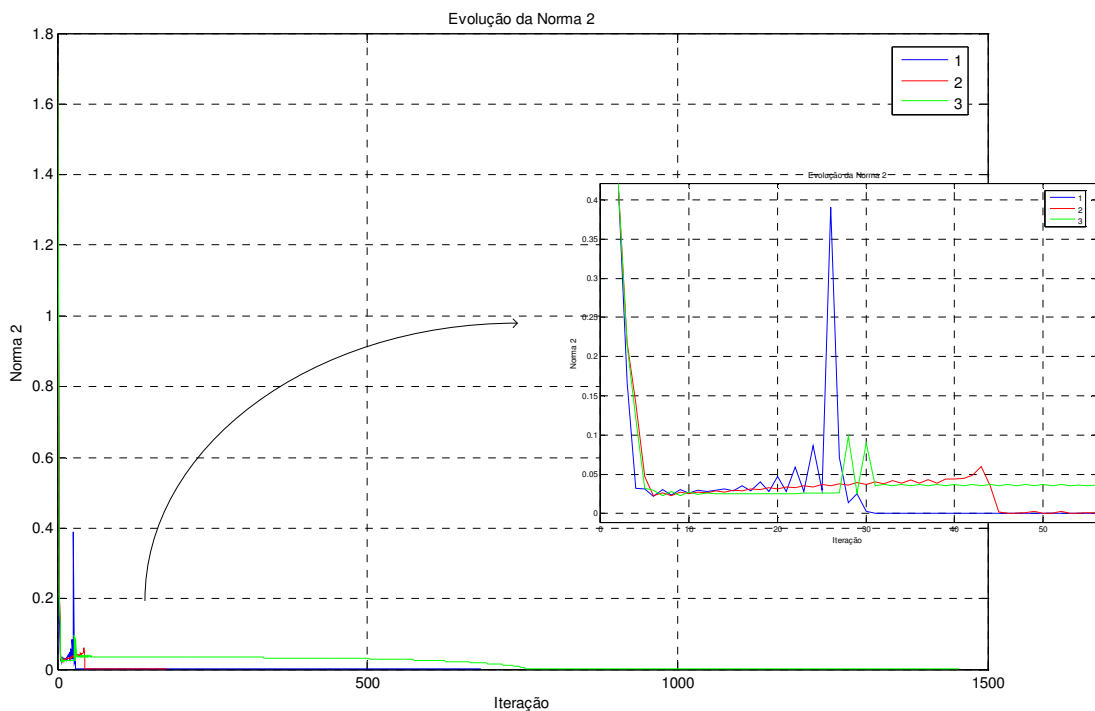


Figura 4-23: Evolução de $\|\nabla F(x_k)\|$.

De forma semelhante à norma do gradiente, é apresentada a evolução do custo da função objetivo na Figura 4-24.

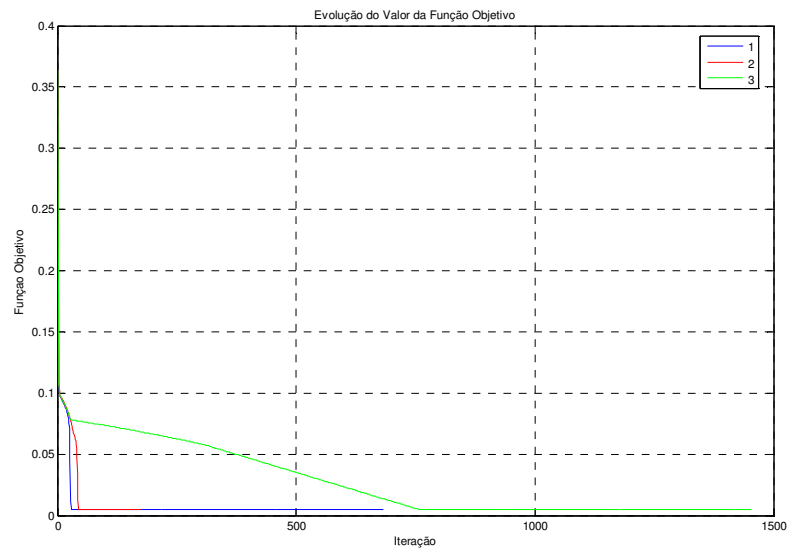


Figura 4-24: Evolução de $F(x_k)$.

Quanto ao ajuste de curva, as curvas se sobrepõem e proporcionam um ajuste adequado, como pode ser visualizado na Figura 4-25.

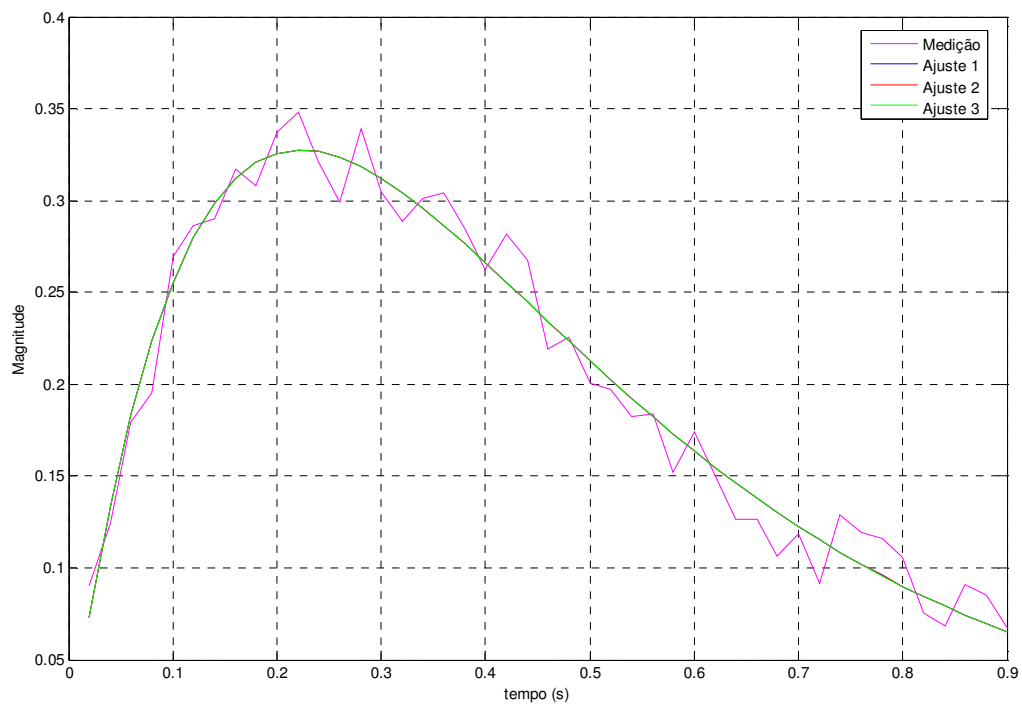


Figura 4-25: Teste de aderência para os pontos obtidos para as diferentes escolhas com $x_0 = [-1 \ -2 \ 1 \ -1]^T$.

4.1.3.2 Outros Pontos Iniciais

Mantendo a tolerância em $\varepsilon=1 \times 10^{-6}$, dois novos pontos iniciais foram testados $x_0 = [-2 \ -2 \ 2 \ -2]^T$ e $x_0 = [0 \ 0 \ 0 \ 0]^T$, e os resultados são apresentados na Tabela 4-8 e Tabela 4-9, respectivamente.

Tabela 4-8: Resultados para $x_0 = [-2 \ -2 \ 2 \ -2]^T$ e $\varepsilon=1 \times 10^{-6}$

Método de Escolha do Passo Inicial	Nº de Iterações	Nº de Interpolações	Nº Máximo de Interpolações	Tempo [s]	Norma	Custo	x(1)	x(2)	x(3)	x(4)
1	681	558	4	1,31	7,494E-07	5,000E-03	-3,9321	-5,0937	3,4525	-3,4530
2	128	31	1	0,31	5,272E-07	5,000E-03	-3,9493	-5,0695	3,5778	-3,5781
3	600	10	1	1,12	5,134E-07	5,000E-03	-3,9500	-5,0687	3,5825	-3,5828

Tabela 4-9: Resultados para $x_0 = [0 \ 0 \ 0 \ 0]^T$ e $\varepsilon=1 \times 10^{-6}$

Método de Escolha do Passo Inicial	Nº de Iterações	Nº de Interpolações	Nº Máximo de Interpolações	Tempo [s]	Norma	Custo	x(1)	x(2)	x(3)	x(4)
1	5	0	0	0,13	4,577E-11	1,063E-01	-0,9705	-0,9705	0,1545	0,1545
2	12	6	2	0,13	4,399E-08	1,063E-01	-0,9705	-0,9705	0,1545	0,1545
3	977	5	2	1,67	9,963E-07	1,063E-01	-0,9705	-0,9705	0,1545	0,1545

Da Tabela 4-8 podem-se inferir conclusões semelhantes ao caso anterior. O teste de aderência com as curvas é apresentado na Figura 4-25. Como no caso anterior as curvas se sobrepõem e proporcionam um ajuste adequado.

Partindo com $x_0 = [0 \ 0 \ 0 \ 0]^T$ o ajuste não é adequado e semelhante ao da Figura 4-9.

4.1.3.3 Conclusões

De acordo com as análises do método Newton-GC conclui-se que a melhor escolha do passo inicial de busca é a número 2. Portanto, a partir desse momento quando o método do gradiente voltar a ser utilizado será com o passo inicial escolhido dessa maneira.

4.2 Análise da Condição de Curvatura $s_k \times y_k^T$

Como mencionado anteriormente, não há garantias que a condição de curvatura seja atendida devido ao algoritmo de busca linear implementado não ser baseado nas condições de Wolfe. Uma alternativa para lidar com esse inconveniente é não atualizar a aproximação da matriz Hessiana quando a condição de curvatura for negativa ou perto de zero, isto é, fazer $H_{k+1} = H_k$.

Aplicando o critério de atualização para condição curvatura negativa (a matriz H só é atualizada se $s_k \times y_k^T$ é maior que zero) obtiveram-se os resultados apresentados na Tabela 4-10. Nota-se que não há nenhuma diferença no comportamento do algoritmo. O mesmo comportamento foi obtido para o ponto inicial $x_0 = [-2 \ -2 \ 2 \ -2]^T$.

Tabela 4-10: Resultados para $x_0 = [-1 \ -2 \ 1 \ -1]^T$ e $\epsilon = 1 \times 10^{-6}$

Testa Condição de Curvatura	Tempo [s]	Nº de Iterações	Nº de Interpolações	Nº Máximo de Interpolações	Nº de Atualizações da Matriz H
Sim	0,22	184	37	2	184
Não	0,22	184	37	2	184

Quanto à aplicação do critério com números pequenos, isso vai depender muito mais da experiência do usuário para determinar o que é um “número pequeno”. Neste caso, para números menores que 1×10^{-10} não foi obtida nenhuma diferença de desempenho. Para o valor 1×10^{-10} e maiores a aplicação do critério mostrou-se ineficiente, prejudicando o desempenho do algoritmo. Tomando como valor pequeno 1×10^{-10} e partindo dos pontos iniciais $x_0 = [-1 \ -2 \ 1 \ -1]^T$ e $x_0 = [-2 \ -2 \ 2 \ -2]^T$ têm-se os resultados mostrados na Tabela 4-11 e na Tabela 4-12, respectivamente.

Tabela 4-11: Resultados para $x_0 = [-1 \ -2 \ 1 \ -1]^T$ e $\epsilon = 1 \times 10^{-6}$

Testa Condição de Curvatura	Tempo [s]	Nº de Iterações	Nº de Interpolações	Nº Máximo de Interpolações	Nº de Atualizações da Matriz H
Sim	0,27	190	42	2	175
Não	0,22	184	37	2	184

Tabela 4-12: Resultados para $x_0 = [-2 \ -2 \ 2 \ -2]^T$ e $\epsilon = 1 \times 10^{-6}$

Testa Condição de Curvatura	Tempo [s]	Nº de Iterações	Nº de Interpolações	Nº Máximo de Interpolações	Nº de Atualizações da Matriz H
Sim	0,39	323	37	1	143
Não	0,211	158	32	1	158

Portanto, conclui-se que implementar $H_{k+1} = H_k$ quando $s_k \times y_k^T$ é negativo ou muito próximo de zero não é uma boa escolha, como também em [1] não se recomenda.

Por último, a comparação de H_k na última iteração com a matriz Hessiana no ponto final mostra como a metodologia de cálculo é eficiente. Na Tabela 4-13 são mostradas as matrizes finais sem a aplicação e com a aplicação do critério de curvatura. Nota-se que sem testar a condição as matrizes apresentam valores similares. Mas, com o teste da condição nota-se que os valores das duas matrizes não são parecidos. Isto é devido à perda de informação quando não há a atualização da matriz H_k .

Tabela 4-13: Matrizes para $x_0 = [-1 \ -2 \ 1 \ -1]^T$ e $\epsilon = 1 \times 10^{-6}$

Sem testar a condição	$H_k^{-1} = \begin{bmatrix} 3,0557 & -2,1716 & 3,1014 & 2,4556 \\ -2,1716 & 1,5934 & -2,4614 & -1,9953 \\ 3,1014 & -2,4614 & 5,9120 & 5,2207 \\ 2,4556 & -1,9953 & 5,2207 & 4,6670 \end{bmatrix}$
	$\nabla^2 F(x_k) = \begin{bmatrix} 3,0489 & -2,1677 & 3,0999 & 2,4556 \\ -2,1677 & 1,5899 & -2,4556 & -1,9905 \\ 3,0999 & -2,4556 & 5,7592 & 5,0690 \\ 2,4556 & -1,9905 & 5,0690 & 4,5160 \end{bmatrix}$
Teste de condição com critério de 1×10^{-10}	$H_k^{-1} = \begin{bmatrix} 3,0075 & -2,1501 & 3,1829 & 2,5410 \\ -2,1501 & 1,6003 & -2,6349 & -2,1668 \\ 3,1829 & -2,6349 & 6,8308 & 6,0999 \\ 2,5410 & -2,1668 & 6,0999 & 5,5073 \end{bmatrix}$
	$\nabla^2 F(x_k) = \begin{bmatrix} 3,0469 & -2,1660 & 3,0990 & 2,4546 \\ -2,1660 & 1,5885 & -2,4546 & -1,9896 \\ 3,0990 & -2,4546 & 5,7594 & 5,0690 \\ 2,4546 & -2,1668 & 5,0690 & 4,5157 \end{bmatrix}$

4.3 Análise da Variação do Parâmetro da Seqüência Forçante

No método Newton-GC a solução inexata da Equação 3-1 feita pelo método do Gradiente Conjugado Linear tem dois critérios de parada: a curvatura negativa e a norma do resíduo.

A norma do resíduo é comparada com o produto da norma da função objetivo por η_k (com $0 < \eta_k < 1$ para todo k). O parâmetro η_k é chamado de seqüência forçante e é dado pela Equação 4-1.

$$\eta_k = \min \left(SF, \sqrt{\|\nabla F(x_k)\|} \right) \quad (4-1)$$

Em que SF é o parâmetro da seqüência forçante. Com valor de 0,5 recomendado por [1].

Com o intuito de avaliar o desempenho do método Newton-GC para diferentes valores do parâmetro SF, foi construída a Tabela 4-14 com o valor de SF entre 0,25 e 0,75. Nota-se que o melhor resultado foi obtido para SF=0,5.

Outra conclusão interessante pode ser obtida observando o número de saídas do GCL. Quando se aumenta ou diminui o valor de SF o número de saídas pela condição curvatura aumenta e há uma diminuição no desempenho do algoritmo.

Tabela 4-14: Resultados para $x_0 = [-1 \ -2 \ 1 \ -1]^T$ e $\varepsilon = 1 \times 10^{-6}$

Parâmetro da Seq. Forçante	Tempo [s]	Nº de Iterações	Nº de Interpolações	Nº Máximo de Interpolações	Nº de Saídas do GCL pela Curvatura	Nº de Saída do GCL pelo Resíduo
0,75	1,30	660	38	1	511	149
0,60	0,43	176	37	1	38	138
0,50	0,42	176	37	1	38	138
0,40	2,05	1016	42	2	843	173
0,25	2,02	1016	42	2	843	173

4.4 Comparação entre os Métodos de Otimização

Com relação ao tempo de execução o método que apresentou o melhor desempenho foi o método Quase-Newton. Embora, o método Newton-GC tenha apresentado o menor número de iterações, o tempo de cálculo por iteração foi muito maior. Isso pode ser devido a dois motivos: cálculo da matriz hessiana e cálculo da direção de busca via GCL. Em apenas uma situação o método Newton-GC teve um desempenho melhor com relação ao tempo de execução.

Os resultados como tempo de execução e número de iterações para diferentes valores de tolerâncias são mostrados da Tabela 4-15 a Tabela 4-18.

Vale lembrar que todos os cálculos executados foram realizados numericamente, sem o uso de programas simbólicos que deixariam a execução do programa excessivamente lenta.

Tabela 4-15: Resultados para $x_0 = [-1 \ -2 \ 1 \ -1]^T$ e $\varepsilon = 1 \times 10^{-4}$

Método	Tempo [s]	Número de Iterações	Número de Interpolações	Número Máximo de Interpolações	Norma	Custo	x(1)	x(2)	x(3)	x(4)
Gradiente	6,99	3424	22	1	9,672E-05	5,029E-03	-3,5238	-5,7588	1,8389	-1,8433
Qnewton	0,19	92	23	1	6,476E-05	5,003E-03	-3,8073	-5,2782	2,7431	-2,7445
Newton-GC	0,23	45	5	1	5,059E-05	5,028E-03	-3,5325	-5,7525	1,8534	-1,8582

Tabela 4-16: Resultados para $x_0 = [-1 \ -2 \ 1 \ -1]^T$ e $\varepsilon = 5 \times 10^{-5}$

Método	Tempo [s]	Número de Iterações	Número de Interpolações	Número Máximo de Interpolações	Norma	Custo	x(1)	x(2)	x(3)	x(4)
Gradiente	8,23	3860	24	1	4,977E-05	5,029E-03	-3,5274	-5,7599	1,8431	-1,8479
Qnewton	0,17	124	37	1	2,712E-05	5,001E-03	-3,9016	-5,1373	3,2499	-3,2505
Newton-GC	0,18	49	7	1	3,807E-05	5,022E-03	-3,5668	-5,6860	1,9353	-1,9395

Tabela 4-17: Resultados para $x_0 = [-1 \ -2 \ 1 \ -1]^T$ e $\varepsilon = 1 \times 10^{-5}$

Método	Tempo [s]	Número de Iterações	Número de Interpolações	Número Máximo de Interpolações	Norma	Custo	x(1)	x(2)	x(3)	x(4)
Gradiente	-	>20000	-	-	-	-	-	-	-	-
Qnewton	0,43	182	53	1	6,62E-06	5,000E-03	-3,9936	-5,0087	3,9416	-3,9416
Newton-GC	0,32	102	20	1	6,826E-06	5,003E-03	-3,7946	-5,2985	2,6851	-2,6867

Tabela 4-18: Resultados para $x_0 = [-1 \ -2 \ 1 \ -1]^T$ e $\varepsilon = 1 \times 10^{-5}$

Método	Tempo [s]	Número de Iterações	Número de Interpolações	Número Máximo de Interpolações	Norma	Custo	x(1)	x(2)	x(3)	x(4)
Gradiente	-	>20000	-	-	-	-	-	-	-	-
Qnewton	0,26	190	42	2	9,01E-07	5,000E-03	-3,9997	-5,0004	3,9975	-3,9975
Newton-GC	0,46	176	37	1	9,247E-07	5,003E-03	-3,9273	-5,1006	3,4188	-3,4193

5 Conclusão

O estudo de métodos computacionais é sem dúvidas um dos aspectos importantes para fixar os conceitos da teoria de otimização. Com certeza essa tarefa não é fácil. É necessário dedicação para analisar os resultados e os códigos à procura de erros de programação e de modelagem.

Os programas computacionais implementados têm a vantagem de funcionar para diferentes funções objetivos, necessitando apenas mudar os arquivos relacionados com a função objetivo.

Os objetivos do trabalho foram atingidos, sendo possível o ajuste de curva adequado com os três métodos. Foram determinadas as melhores escolhas do passo inicial de busca para os três métodos, de forma a obter o menor custo computacional. Analisou-se questões específicas dos métodos como o impacto de se implementar $H_{k+1} = H_k$, quando $s_k \times y_k^T$ é negativo ou muito próximo de zero no método Quase-Newton, sendo essa implementação inadequada. E, para o método Newton-GC verificou-se o desempenho do algoritmo para diferentes valores do parâmetro da sequência forçante, concluindo-se que a recomendação de [1] é a mais adequada.

Quanto ao desempenho dos métodos, os menores tempos de execução foram obtidos com o método Quase-Newton. Com este também foram obtidos os menores valores de norma e custo.

Cabe ressaltar que não há um método melhor que o outro, e sim mais adequado para a resolução de um problema específico.

6 Referências Bibliográficas

- [1] NOCEDAL, J.; WRIGHT, S. J. "Numerical Optimization ". 1ª Edição. 175 Fifth Avenue, New York, NY 10010, USA: Springer, 1999.
- [2] FINARDI, E. C. Notas de aula da disciplina EEL6000 – Métodos Numéricos de Otimização. 2008.

Apêndice

Apêndice I - Listagem do código do arquivo de dados

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% dados.m: Dados das medições
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

global Y
Y = [
0.090542
0.124569
0.179367
0.195654
0.269707
0.286027
0.289892
0.317475
0.308191
0.336995
0.348371
0.321337
0.299423
0.338972
0.304763
0.288903
0.300820
0.303974
0.283987
0.262078
0.281593
0.267531
0.218926
0.225572
0.200594
0.197375
0.182440
0.183892
0.152285
0.174028
0.150874
0.126220
0.126266
0.106384
0.118923
0.091868
0.128926
0.119273
0.115997
0.105831
0.075261
0.068387
0.090823
0.085205
0.067203
];
```

Apêndice II – Listagem do código: Método do Gradiente

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% gradiente.m: Otimização via Método do Gradiente
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

toler=1e-4; % tolerância

% inicialização
dados;
x = input('Entre com o ponto inicial x0: ');
k=1;
f(k)=fx(x);
dfxk=dfx(x);
norma(k)=norm(dfxk,2);

% laço
while ~(norma(k)<=toler)
    X(:,k)=x;

    p=-dfxk; % direção de descida

    f(k)=fx(x); % valor atual da função objetivo

    if k~=1
        alpha0=alpha(k-1)*(dfxkm1'*pkml)/(dfxk'*p);
    else
        alpha0=1;
    end
    dfxkm1=dfxk;
    pkml=p;

    [alpha(k),i(k)]=btls(x,alpha0,p,dfxk,f(k)); % backtracking line search

    x=x+alpha(k)*p; % atualização de x

    k=k+1;

    dfxk=dfx(x);
    norma(k)=norm(dfxk,2); % norma 2 do gradiente
end

% etapa final
X(:,k)=x;
f(k)=fx(x); % valor da função objetivo
k=k-1; % número total de iterações
```

Apêndice III - Listagem do código: Método Quase-Newton

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% qnewton.m: Otimização via Método Quase-Newton
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

toler=1e-6; % tolerância

% inicialização
dados;
x = input('Entre com o ponto inicial x0: ');
k=1;
h=1;
I=eye(4,4);
H=I;
f(k)=fx(x);
dfxk=dfx(x);
norma(k)=norm(dfxk,2);

% laço
while ~(norma(k)<=toler)
    X(:,k)=x;

    p=-H*dfxk; % direção de descida

    f(k)=fx(x); % valor atual da função objetivo

    if k~=1
        alpha0=2*(f(k)-f(k-1))/(dfxk'*p);
    else
        alpha0=1;
    end

    [alpha(k),i(k)]=btls(x,alpha0,p,dfxk,f(k)); % backtracking line search

    x=x+alpha(k)*p; % atualização de x

    % cálculo da matriz H
    s=alpha(k)*p;
    dfxk_old=dfxk;
    dfxk=dfx(x);
    y=dfxk-dfxk_old;
    ro(k)=1/(y'*s);
    if k~=1 %&& s'*y>toler/10000
        H=(I-ro(k)*s*y')*H*(I-ro(k)*y*s')+ro(k)*s*s';
        h=h+1; % contador de atualizações de H
    elseif k==1
        H=((y'*s)/(y'*y))*I;
        H=(I-ro(k)*s*y')*H*(I-ro(k)*y*s')+ro(k)*s*s';
    end

    k=k+1;

    norma(k)=norm(dfxk,2); % norma 2 do gradiente
end

% etapa final
X(:,k)=x;
```

```
f(k)=fx(x); % valor da função objetivo
k=k-1; % número total de iterações
```

Apêndice IV - Listagem do código: Método Newton-GC

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% gcnewton.m: Otimização via Método Newton-GC
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

toler=1e-6; % tolerância do método

% inicialização
dados;
x = input('Entre com o ponto inicial x0: ');
k=1;
dfxk=dfx(x);
norma(k)=norm(dfxk,2);
f(k)=fx(x);
r=0; c=0;

% laço
while ~(norma(k)<=toler)
    X(:,k)=x;

    Hes=hessiana(x); % matriz hessiana em x
    [p,kr,kc]=gcl(Hes,-dfxk,norma(k)); % direção de descida via GCL
    r=r+kr; % contador de saídas pelo resíduo
    c=c+kc; % contador de saídas pela curvatura

    f(k)=fx(x); % valor atual da função objetivo

    if k~=1
        alpha0=2*(f(k)-f(k-1))/(dfxk'*p);
    else
        alpha0=1;
    end

    [alpha(k),i(k)]=btls(x,alpha0,p,dfxk,f(k)); % backtracking line search

    x=x+alpha(k)*p; % atualização da variável

    k=k+1;

    dfxk=dfx(x);
    norma(k)=norm(dfxk,2); % norma 2 do gradiente
end

% etapa final
X(:,k)=x;
f(k)=fx(x); % valor da função objetivo
k=k-1; % número total de iterações
```

Apêndice V - Listagem do código: “Backtracking Line Search”

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% btls.m: Backtracking Line Search com interpolação quadrática
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [alpha, i] = btls(x, alpha0, p, dfxk, fxk)

c1=1e-4;
i=0; % contador de interpolações
fxka=fx(x+alpha0*p);

while fxka>(fxk+c1*alpha0*dfxk'*p) % Condição de Decréscimo Suficiente
    alpha0=-dfxk'*p*alpha0^2/(2*(fxka-fxk-dfxk'*p*alpha0));
    fxka=fx(x+alpha0*p);
    i=i+1;
end

alpha=alpha0;
```

Apêndice VI - Listagem do código: Método do GCL

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% gcl.m: Cálculo da direção de busca via GCL
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [x, kr, kc] = gcl(A, b, norma)

% iniciliazação
x=zeros(4,1);
r=A*x-b;
p0=-r;
SF=0.5; % parâmetro da sequência forçante

if p0'*A*p0<=0
    alpha=-(r'*p0)/(p0'*A*p0);
    x=x+alpha*p0;
else
    % critério de parada: convergência ou direção de descida encontrada
    while ~(norm(r,2)<=min(SF, sqrt(norma))*norma || p0'*A*p0<=0)
        alpha=-(r'*p0)/(p0'*A*p0);
        x=x+alpha*p0;
        r=A*x-b;
        beta=(r'*A*p0)/(p0'*A*p0);
        p0=-r+beta*p0;
    end
end

if p0'*A*p0<=0
    kc=1; % contador de saídas pela condição de curvatura
    kr=0;
else
    kc=0;
    kr=1; % contador de saídas pelo resíduo
end
```

Apêndice VII - Listagem do código: Função Objetivo

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% fx.m: Cálculo do valor da função objetivo
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function f = fx(x)

global Y
f=0;
t=0.02*[1:45]';

f = sum(0.5*(Y-(x(3)*exp(x(1)*t)+x(4)*exp(x(2)*t))).^2);
```

Apêndice VIII - Listagem do código: Gradiente da Função Objetivo

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% dfx.m: Cálculo do gradiente da função objetivo
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function df = dfx(x)

global Y
df=zeros(4,1);
t=0.02*[1:45]';

df=[ (Y-(x(3)*exp(x(1)*t)+x(4)*exp(x(2)*t)))*(-x(3)*t.*exp(x(1)*t))
      (Y-(x(3)*exp(x(1)*t)+x(4)*exp(x(2)*t)))*(-x(4)*t.*exp(x(2)*t))
      (Y-(x(3)*exp(x(1)*t)+x(4)*exp(x(2)*t)))*(-exp(x(1)*t))
      (Y-(x(3)*exp(x(1)*t)+x(4)*exp(x(2)*t)))*(-exp(x(2)*t))
    ];
```

Apêndice XI - Listagem do código: Hessiana da Função Objetivo

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% hessiana.m: Hessiana da função objetivo
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function H = hessiana(x)

global Y
H=zeros(4,4);
t=0.02*[1:45]';

aux=Y-(x(3)*exp(x(1)*t)+x(4)*exp(x(2)*t));

H(1,1)=sum((x(3)*t.*exp(x(1)*t)).^2+aux.*(-x(3)*t.^2.*exp(x(1)*t)));
H(1,2)=sum(x(3)*x(4)*t.^2.*exp(x(1)*t+x(2)*t));
H(1,3)=sum(x(3)*t.*exp(2*x(1)*t)+aux.*(-t.*exp(x(1)*t)));
H(1,4)=sum(x(3)*t.*exp(x(1)*t+x(2)*t));
H(2,2)=sum((x(4)*t.*exp(x(2)*t)).^2+aux.*(-x(4)*t.^2.*exp(x(2)*t)));
H(2,3)=sum(x(4)*t.*exp(x(1)*t+x(2)*t));
H(2,4)=sum(x(4)*t.*exp(2*x(2)*t)+aux.*(-t.*exp(x(2)*t)));
H(3,3)=sum(exp(x(1)*t).^2);
H(3,4)=sum(exp(x(1)*t+x(2)*t));
H(4,4)=sum(exp(x(2)*t).^2);

H(2,1)=H(1,2);    H(3,1)=H(1,3);
H(3,2)=H(2,3);    H(4,1)=H(1,4);
H(4,2)=H(2,4);    H(4,3)=H(3,4);
```