Universidade Federal do Rio de Janeiro
Programa de Engenharia Elétrica

Alunos: Bernardo Bouzan
Elly Fonseca Bouzan
Leonardo Santoro
Professor: Afonso Celso Del Nero
Data: 04/11/2010

Otimização – Aspectos Teóricos e Métodos Numéricos:

Trabalho 2

1 – Introdução

O relatório a seguir tem por objetivo apresentar brevemente o conceito utilizado nos métodos vistos em sala de aula para a escolha da direção de descida e, posteriormente, detalhar as construções de seus algoritmos a partir do software Matlab versão 7.8.0. Nesse relatório, foram utilizados como métodos de busca linear os algoritmos da Seção Áurea, Fibonacci e Interpolação Polinomial, todos apresentados no trabalho anterior.

2 – Revisão Conceitual

2.1 – Método do Gradiente

Este método tem por objetivo o cálculo da direção de descida que minimiza a função objetivo. O princípio utilizado é o de que a derivada direcional de uma função será mínima se a direção do passo tiver mesma direção e sentido oposto ao vetor gradiente da função. Dessa forma, deseja-se identificar em que direção a função deve ser percorrida para que seja minimizada. Depois de identificada a direção, deve-se saber o tamanho do passo a ser dado, e para isso serão utilizados os algoritmos de busca linear vistos anteriormente. Em seguida, deve-se calcular novamente a direção e assim sucessivamente até que os critérios de parada sejam satisfeitos.

2.1.1 – Algoritmo: Método do Gradiente

A - $x_k = x_0$

B - $\nabla f(x_k) = g(x_k)$

C – $\|g(x_k)\| < \varepsilon \longrightarrow x^* = x_k \longrightarrow$ FIM

D – $d_k = -g(x_k)\,\|g(x_k)\|^{-1}$

E - $\tilde{f}(\alpha) = f(x_k + \alpha d_k)$

F – $\alpha_k = min\{\tilde{f}(\alpha)\}$

G - $x_{k+1} = x_k + \alpha_k d_k$

H – Critérios de Parada são satisfeitos ?

- Sim: $x = x_{k+1} \longrightarrow$ FIM.

- Não: $x_k = x_{k+1} \longrightarrow$ Retornar para o passo B.

Deve ser destacado que, utilizando o Método do Gradiente, a direção de descida é sempre a direção de descida máxima e, dadas duas direções de descida consecutivas quaisquer, elas serão perpendiculares, ou seja, os avanços ocorrem em zig-zag com ângulos retos a partir do ponto inicial x_0. Além disso, a convergência desse algoritmo está relacionada ao condicionamento da Hessiana de f. Hessianas bem condicionadas nos pontos de interesse são excelentes para o método e, caso contrário, a convergência pode ser demasiadamente lenta.

2.2 – Método do Gradiente Conjugado

Os problemas de convergência encontrados, principalmente para funções do tipo quadrática (de vasta importância no estudo de otimização), utilizando-se o Método do Gradiente devem-se às bruscas mudanças de direção citadas anteriormente. Para minimizar esse efeito, o Método do Gradiente Conjugado estabelece uma nova expressão para o cálculo da direção de descida, que não será mais a direção máxima, mas sim uma composição da direção de descida máxima com a considerada anteriormente. Dessa forma, espera-se suavizar as mudanças de direção ocorridas e melhorar a convergência. Assim, será considerado que:

$$d_{k+1} = -g(x_{k+1}) + \beta_k d_k.$$

Resta apenas estabelecer como é feito o cálculo de β_k. Nesse trabalho, foram adotadas duas maneiras de estabelecer tal parâmetro, que conduz ao Método do Gradiente Conjugado na forma de Polak-Rebière e na forma de Fletcher-Reeves. São eles:

- Polak-Rebière: $\beta_k = \frac{\|g_{k+1}\| - [g_{k+1}]^T g_k}{\|g_k\|}$.

- Fletcher-Reeves: $\beta_k = \frac{\|g_{k+1}\|}{\|g_k\|}$.

2.2.1 – Algoritmo: Método do Gradiente Conjugado (para funções objetivo na forma $f(x) = \frac{1}{2}x^T Ax + b^T x, sendo\ A > 0$.

A - $g_0 = Ax_0 + b$; $d_0 = -g_0$

B - $\alpha_0 = \frac{-d_0^T g_0}{d_0^T A d_0}$

C - $x_1 = x_0 + \alpha_0 d_0$

D - $g_1 = Ax_1 + b$

E - β_0 = Polak-Rebière ou Fletcher-Reeves

F - $d_1 = -g_1 + \beta_0 d_0$

G - $\alpha_1 = \frac{-d_1^T g_1}{d_1^T A d_1}$

Assim, após n iterações, o método alcançará a solução exata do problema, sendo n a dimensão do vetor x.

2.3 – Métodos de Newton

São algoritmos de segunda ordem cujo objetivo, assim como os métodos anteriores, é calcular a direção de descida que deve ser adotada para minimizar uma função objetivo f(x). A cada passo, deve ser encontrada uma aproximação quadrática para a função objetivo f(x), cujo mínimo será o próximo passo, utilizando o Teorema de Taylor. Assim, a função aproximada calculada dependerá do gradiente e da Hessiana da função original f(x).

O método original, por ser de segunda ordem, converge rapidamente, quando converge. Porém, por muitas vezes, problemas associados à Hessiana fazem com que a convergência não seja alcançada. Deve ser observado que uma condição necessária para a convergência é que a Hessiana calculada a cada iteração seja positiva definida. Assim, nota-se que situações de falha na convergência são comuns ao utilizar o Método de Newton.

Muitas variações surgiram para corrigir os problemas vinculados à Hessiana calculada e garantir a convergência do método. Esses algoritmos são conhecidos por Métodos de Newton Modificados. Nesse trabalho, foi implementado um algoritmo que considera uma diminuição do tamanho do passo na direção da descida através do uso de um algoritmo de busca linear, ao invés de um passo unitário adotado no método original. Além disso, a condição necessária de uma Hessiana definida positiva a cada iteração se constituía a maior dificuldade do método. A correção desse problema pode ser alcançada ao substituir a Hessiana calculada pela expressão abaixo:

$F_k = G_k + \delta I_n$, sendo:

G_k: a Hessiana calculada na iteração k.

I_n: a matriz identidade de orden n.

δ: escalar calculado de forma que os autovalores de F_k sejam positivos.

2.3.1 – Algoritmo: Método de Newton Modificado.

A - $x_k = x_0$

B - $g_k = \nabla f(x_k)$; $G_k = \nabla^2 f(x_k)$

C – $\delta \epsilon \mathbb{R}\ tal\ que\ \lambda_i(G_k + \delta I_n) > 0$

D - $F_k = G_k + \delta I_n$

E - $d_k = -[F_k]^{-1} g_k$

F - $\tilde{f}(\alpha) = f(x_k + \alpha d_k)$

G - $\alpha_k = \min \tilde{f}(\alpha)$

H - $x_{k+1} = x_k + \alpha_k d_k$

I – Critérios de Parada são satisfeitos ?

- Sim: $x = x_{k+1}$ $\longrightarrow$ FIM.
- Não: $x_k = x_{k+1}$ $\longrightarrow$ Retornar para o passo B.

2.4 – Métodos de Quase Newton

Mais uma vez, o objetivo dos métodos será o de encontrar a direção de descida para minimizar uma função objetivo f(x). Contudo, os algoritmos de Quase Newton visam facilitar o cálculo da Hessiana, encontrando-se a mesma sem a necessidade de derivação por duas vezes da função objetivo f(x). A idéia a ser explorada aqui é a de que deve ser possível fazer a construção recursiva da estimativa da Hessiana (ou de sua inversa), durante o decorrer de um processo de otimização. A estimativa parcial da Hessiana deve ser utilizada no decorrer desse processo. Isso é particularmente útil na otimização de funções não-quadráticas, em que a Hessiana não é constante: esse procedimento permite a adaptação contínua da estimativa da Hessiana ao seu valor localmente válido.

Foram considerados dois métodos distintos, porém, a única diferença entre ambos ocorre no cálculo da estimativa da Hessiana:

- Davidon-Fletcher-Powell (DFP):

$$H_{k+1} = H_k - \frac{H\,\gamma\gamma^T H}{\gamma^T H\gamma} + \frac{\delta\delta^T}{\delta^T\gamma}$$

-Broyden-Fletcher-Goldfarb-Shanno (BFGS):

$$H_{k+1} = H_k - \frac{\delta\gamma^T H + H\gamma\delta^T}{\delta^T\gamma} + \left(1 + \frac{\gamma^T H\gamma}{\delta^T\gamma}\right) \frac{\delta\delta^T}{\delta^T\gamma}$$

2.4.1 – Algoritmo: Método de Newton Modificado.

A - $x_k = x_0$; $H_k = H_0 = I$

B - $g_k = \nabla f(x_k)$

C - $d_k = -H_k\, g_k$

D - $\tilde{f}(\alpha) = f(x_k + \alpha\, d_k)$

E - $\alpha_k = \min \tilde{f}\,(\alpha)$

F - $x_{k+1} = x_k + \alpha_k\, d_k$

G – Critérios de Parada são satisfeitos ?

- Sim: $x = x_{k+1}$ ⟶ FIM.

- Não: Prosseguir ao passo H

H - $\delta_k = x_{k+1} - x_k$

I - $g_{k+1} = \nabla f(x_{k+1})$

J – $\gamma_k = g_{k+1} - g_k$

K - $H_{k+1} = DFP\ ou\ BFGS$

L – k = k + 1 ⟶ Retornar para C.

3 – Manual de Execução

Para inicializar um dos métodos, a função *optimize* deve ser executada a partir da linha de comando do Matlab. Após isso, deverá aparecer o conteúdo exibido na Figura 1. Toda interface do programa foi desenvolvido através da ferramenta GUIDE do Matlab.

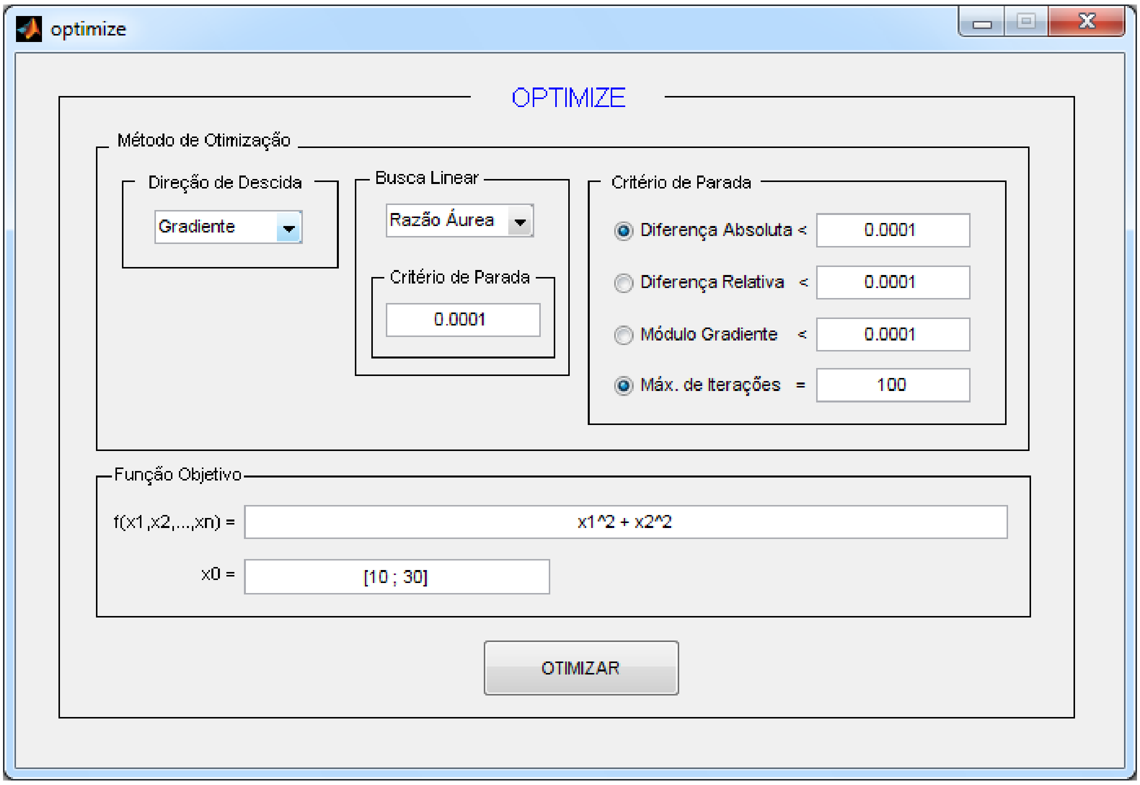

Figura 1 – Inicializando a função Optimize

De acordo com as Figuras 2 e 3, nota-se que diversos métodos para cálculo de direção de descida e busca linear foram implementados e estão disponíveis para uso. São eles:

- Algoritmos de Busca Linear: Fibonacci, Razão Áurea, Interpolação Quadrática.

- Algoritmos de Cálculo de Direção de Descida: Gradiente, Gradiente Conjugado (PR e FR), Newton, Newton Modificado, Quase Newton (DFP e BFGS).

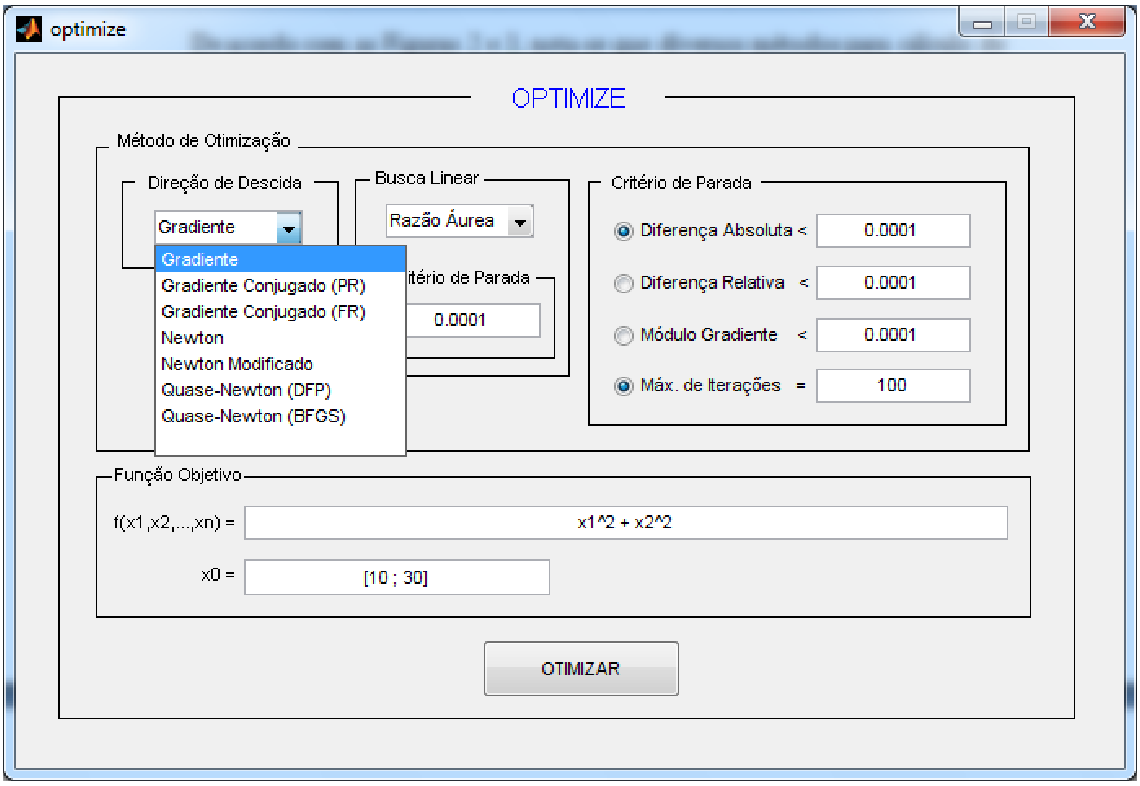

Figura 2 – Optimize: Algoritmos de Direção de Descida

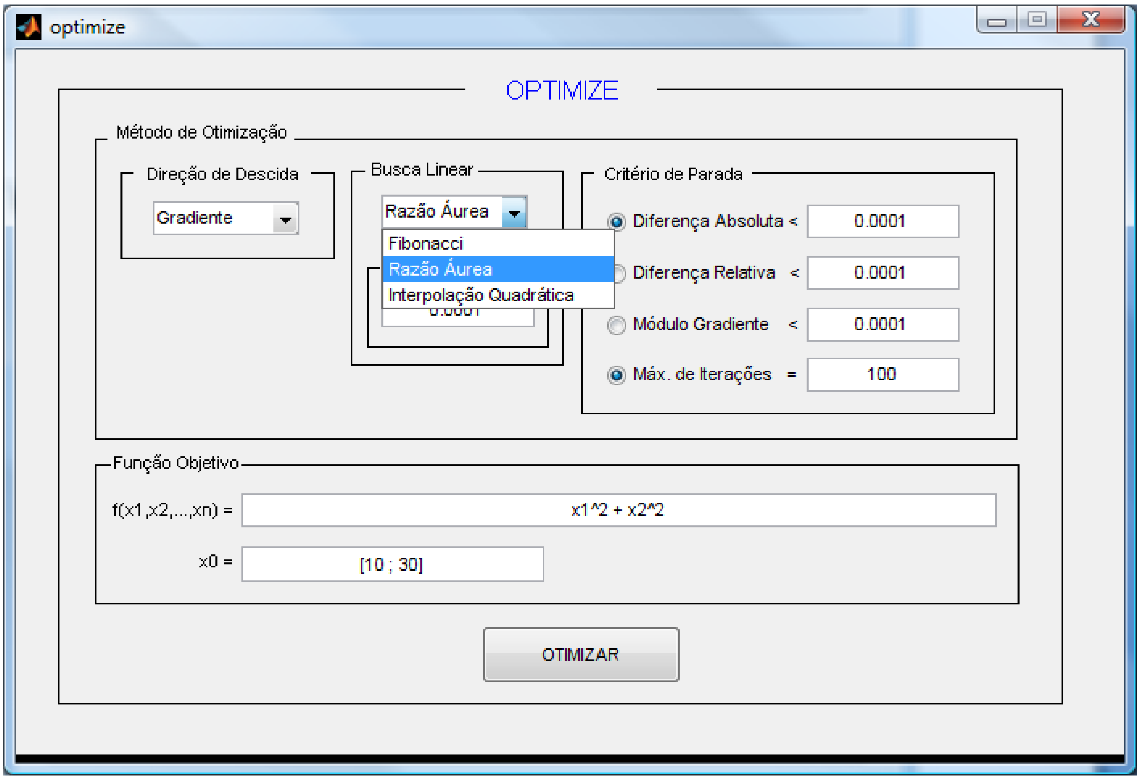

Figura 3 – Optimize: Algoritmos de Busca Linear disponíveis

O primeiro passo para a utilização da ferramenta é definir qual será a função objetivo que deverá ser minimizada, conforme Figura 4. O usuário deve observar que a função objetivo deve ser uma função de $x_1, x_2, \dots, x_n$. Caso contrário a rotina utilizada para reconhecimento da função objetivo não será capaz de fazê-lo e o processo de otimização sequer será iniciado. Além da função a ser minimizada, o usuário deverá definir qual será o ponto a partir do qual os métodos deverão partir até chegar ao ponto de mínimo. Deve ser observado que o vetor x_0 deverá possuir dimensão compatível com a função objetivo inserida.

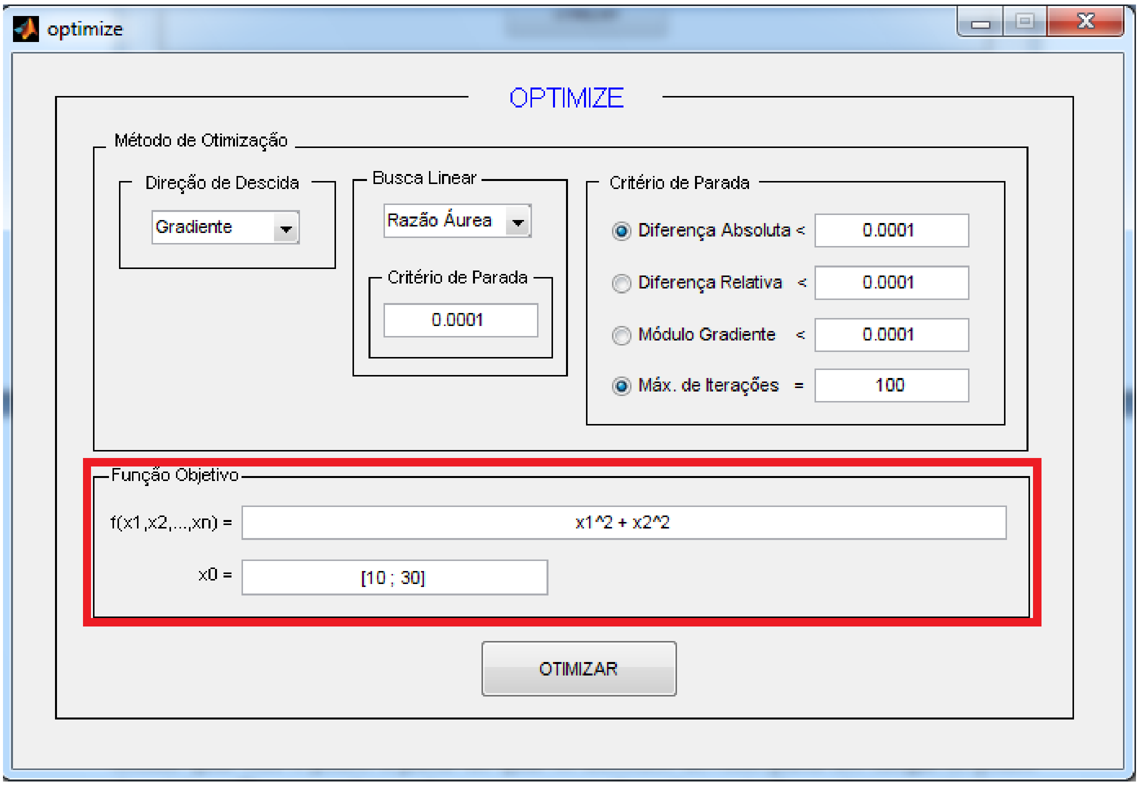

Figura 4 – Optimize: Função Objetivo e Ponto Inicial

A etapa seguinte deve ser a escolha dos métodos que serão utilizados para o cálculo da Direção de Descida e para Busca Linear, conforme mostrado anteriormente. No caso da Busca Linear, o usuário deve, além de selecionar um algoritmo, definir qual será o valor absoluto da tolerância utilizada no cálculo do tamanho do passo. A forma como essa tolerância é aplicada aos métodos disponibilizados foram anteriormente definidas no Trabalho 1.

Por último, o usuário deverá definir qual(is) critério(s) de parada o programa utilizará para definir que um dado ponto x é um ponto de mínimo x_{min}, conforme mostrado na Figura 5. Nota-se também que é possível a utilização de até 4 critérios simultaneamente.

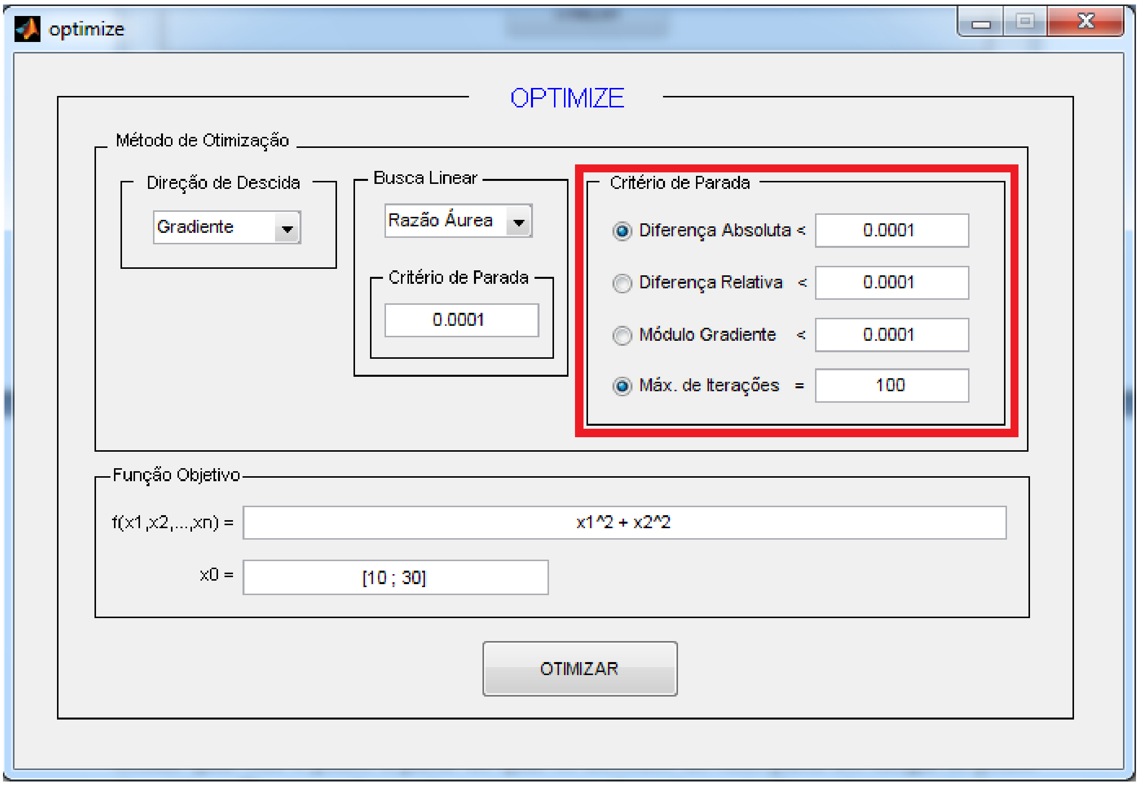

Figura 5 – Optimize: Critérios de Parada

4 – Resolução de um Exemplo através da Optimize

Para que não haja dúvidas na utilização da ferramenta criada, será resolvido um exemplo prático retirado da literatura indicada para o curso. Além disso, espera-se, através de um exemplo prático, apresentar todos os resultados exibidos pelo programa.

Será resolvido, a seguir, o problema descrito na página 80 da apostila base do curso, cujo objetivo é minimizar a função objetivo $f(x) = x_1^2 + 3x_2^2 + 2x_1 - 12x_2$ através do algoritmo do Gradiente Conjugado (PR), utilizando ponto inicial $x_0 = [4 \;\; 4]^T$.

Após executar a função optimize a partir da tela inicial do Matlab, devem-se inserir os dados exigidos no problema descrito acima. A Figura 6 mostra os dados inseridos no programa.

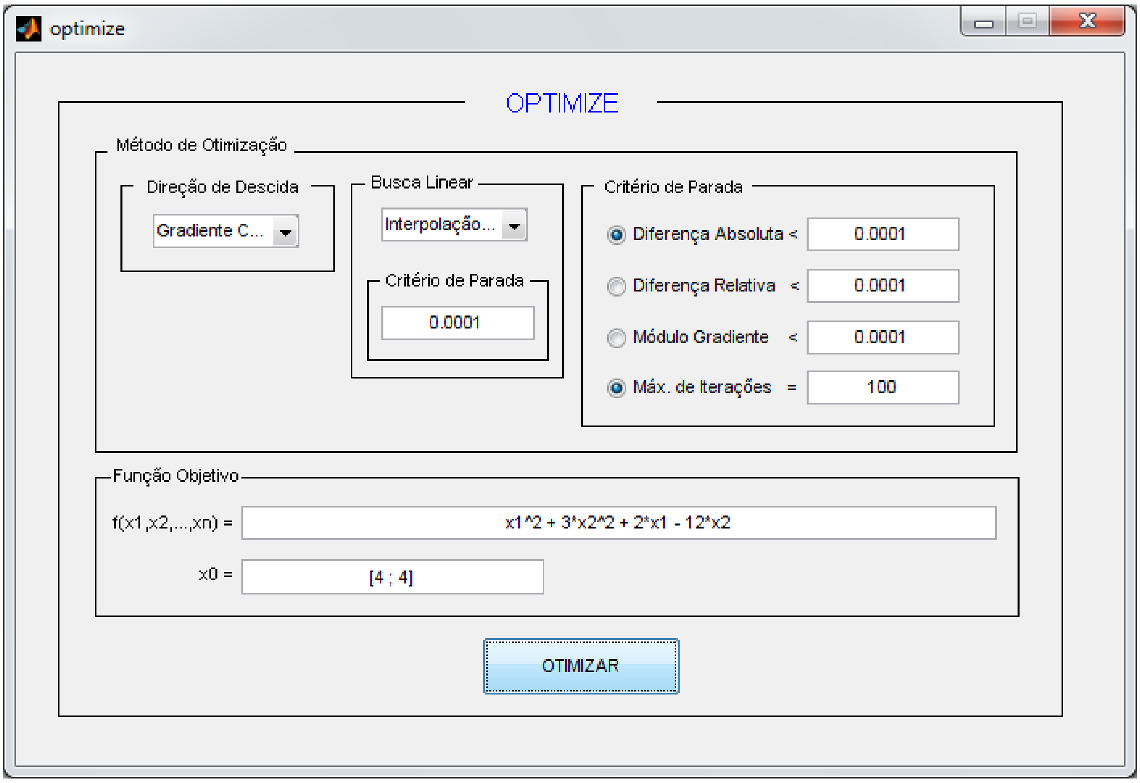

Figura6 – Optimize: Exemplo de Uso

Após a execução do programa, automaticamente é gerado um relatório contendo a função objetivo que foi minimizada, o método de direção de descida escolhido, o algoritmo de busca linear utilizado, o ponto de partida, o ponto mínimo encontrado, a figura plotada para a função objetivo mostrando-se a evolução dos pontos a cada iteração e uma Tabela de iterações. No Anexo A deste relatório é possível observar, na íntegra, o conteúdo do relatório gerado pelo programa Optimize para este exemplo.

A Figura 7 e a Tabela 1 são mostradas a seguir.

Núm. Iterações	**|d|**	**grad(f)**	**Passo**	**Núm. Iterações Busca Linear**
1	15.6205	15.6205	0.22932	1
2	110.3005	7.0468	0.00088566	1
3	6.748	6.8463	0.29947	1
4	26.6058	3.8413	0.005296	1
5	3.3769	3.5188	0.25958	1
6	6.3802	1.7314	0.01482	1
7	1.4063	1.5042	0.37451	1
8	1.5406	0.83351	0.059329	1
9	0.47691	0.53832	0.427	1
10	0.29998	0.24926	0.12186	1
11	0.10084	0.10678	0.52519	1
12	0.017122	0.017239	0.16765	1

Tabela 1 – Optimize: Tabela criada após minimizar a função objetivo f(x)

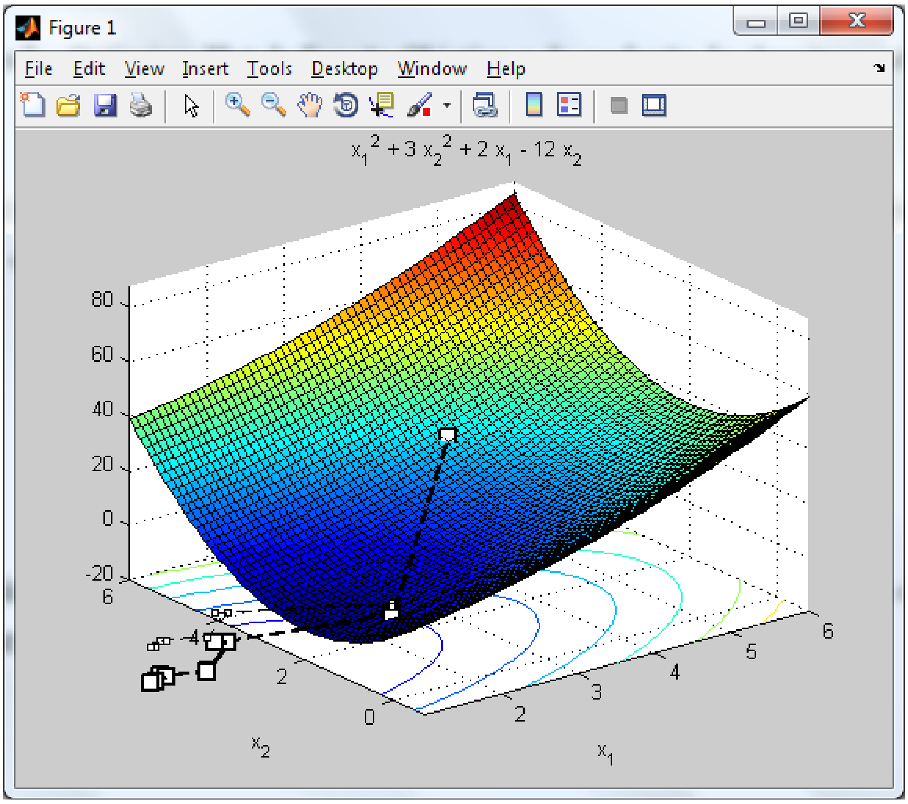

Figura7 – Optimize: Plot da Função Objetivo e da evolução do algoritmo sobre a mesma

Calculando-se analiticamente o ponto de mínimo da função objetivo $f(x) = x_1^2 + 3x_2^2 + 2x_1 - 12x_2$, chega-se ao seguinte resultado:

$$\frac{\partial f(x)}{\partial x1} = 2x_1 + 2$$

$$\frac{\partial f(x)}{\partial x2} = 6x_2 - 12$$

Assim, o ponto extremo analítico é: $x^* = [\ -1 \quad 2\ ]^T$. A Hessiana de f(x) pode ser escrita da seguinte forma:

$$H = \begin{bmatrix} 2 & 0 \\ 0 & 6 \end{bmatrix}$$

Dada a Hessiana acima, pode-se afirmar que o ponto calculado é um ponto de mínimo, uma vez que seus autovalores são positivos.

Portanto, o ponto $x = [\ -0.9996061 \quad 1.99999\ ]^T$ encontrado numericamente corresponde ao valor esperado.

Anexo A