



UFRJ

Trabalho Final

INTRODUÇÃO À OTIMIZAÇÃO

Alunos:

Cayo Valsamis

Gabriel Pelielo

Rafael Accácio

Rodrigo Moysés

Professor:

Afonso Celso del Nero

17 de Março de 2016

Conteúdo

1	Introdução	3
2	Mínimos Quadrados	3
3	Algoritmo Genético	5
3.1	Primeira geração	5
3.2	População intermediária	6
3.3	Segunda geração	7
4	Simplex	9
4.1	Ordenação	9
4.2	Centróide	9
4.3	Reflexão	10
4.4	Expansão	10
4.5	Contração	11
4.6	Encolhimento	12
4.7	Crítérios de Parada	13
5	Interface Gráfica	14
6	Resultados e Discussão	15
7	Conclusão	18

1 Introdução

O trabalho detalhado neste relatório se baseia em analisar métodos numéricos “pobres” para realizar busca de mínimos de funções vetoriais e compará-los aos métodos “nobres”. Os métodos implementados foram:

- Simplex ou Nelder-Mead
- Algoritmo Genético

Existem diversas variações do método genético para minimização de funções, e por isso foi escolhido um deles e detalhado na sua respectiva seção.

Os métodos implementados são chamados de “pobres” porque não necessitam de cálculos de derivadas ou hessianas, mas trabalham utilizando a força computacional dos computadores modernos. O objetivo deste trabalho é implementar tais métodos e compará-los aos métodos “nobres”, que realizam cálculos rebuscados para alcançar o mínimo das funções e mostrar quais são as vantagens e desvantagens de cada método.

Além disso, foram testados os métodos construídos de acordo com um problema real, para que fosse possível compará-los entre si e entre os métodos construídos em trabalhos passados.

2 Mínimos Quadrados

O *Método dos Mínimos Quadrados* é uma ferramenta matemática que procura encontrar a melhor configuração, a melhor distribuição de um conjunto de dados tentando minimizar a soma dos quadrados das diferenças entre o valor estimado e os dados observados (também conhecidos como erro ou resíduos) e consiste em um estimador que minimiza a soma dos quadrados dos resíduos da regressão, de forma a maximizar o grau de ajuste do modelo aos dados observados.

Como já é conhecido, as medições podem apresentar erro, o que pode interferir na identificação. Por isso tomamos como premissa que o erro de medição é aleatório de variância e média constantes, e o mais importante, independente do que queremos medir, ou seja, seja uma variável aleatória não correlatada. Desse modo, o valor de minimização encontrado seja o mais próximo do ótimo possível. Caso alguma dessas considerações não seja verdadeira encontraremos valores um pouco distantes do esperado.

Havendo dito que a ideia é minimizar o quadrado dos erros. Como neste trabalho queremos fazer o “fit” de uma função que conhecemos com os valores encontrados experimentalmente podemos seguir utilizando a equação a seguir:

$$\min\left(\sum_{i=1}^k (\mathbf{f}(t_k, \omega_n, \zeta) - \mathbf{y}_k)\right) \quad (2.1)$$

Onde k é a amostra, $\mathbf{f}(t_k, \omega_n, \zeta)$ é a expressão da função de resposta no tempo para o sistema proposto utilizando o instante de tempo t_k da k -ésima amostra, \mathbf{y}_k é o valor medido na amostra k e ω_n e ζ são os parâmetros do sistema que queremos encontrar.

Dessa forma dividimos o problema em dois.

1. Calcular a expressão $\sum_{i=1}^k (\mathbf{f}(t_k, \omega_n, \zeta) - \mathbf{y}_k)$
2. Minimizar a expressão através de algum método de minimização

Para primeira parte foi utilizada o toolbox de expressões simbólicas do matlab. E depois de calculada a expressão, foi passada como argumento para funções de minimização de 2 variáveis, já que a expressão fica toda em função de ω_n e ζ .

Foram feitos testes com os métodos de minimização feitos no trabalho anterior [4] e também para mais dois que serão apresentados nas próximas seções.

3 Algoritmo Genético

O *Algoritmo Genético* é uma família de modelos computacionais inspirados pelo evolucionismo. Esses algoritmos são utilizados para encontrar soluções aproximadas de problemas de otimização.

Para ser construído, esse algoritmo necessita de uma **função-objetivo**, que é a expressão matemática que será minimizada. A grande vantagem desse algoritmo é que não é necessário conhecer inteiramente a função, como as suas derivadas, por exemplo. O único conhecimento necessário é saber o valor da função para cada indivíduo.

O **indivíduo** é outro componente necessário para realizar o algoritmo. O indivíduo representa um dos valores no espaço de busca, ou seja, é uma das opções para o mínimo da função-objetivo ser alcançado. O conjunto de todos esses indivíduos representa a **população**, que é o grupo total de opções possíveis para a minimização [1].

O algoritmo construído pode ser dividido em etapas, que foram divididas de acordo com as gerações de populações. A figura 1 ilustra o processo do algoritmo genético feito neste trabalho com todas as etapas do processo de construção do algoritmo [5].

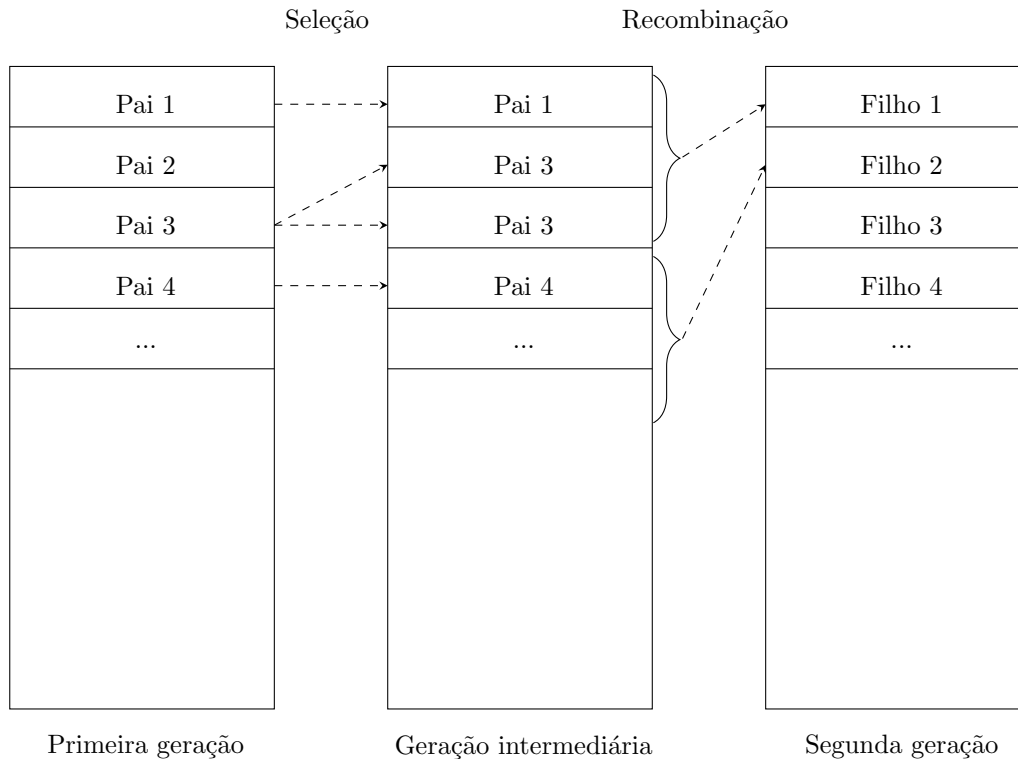


Figura 1: Infográfico ilustrando o algoritmo genético

3.1 Primeira geração

O algoritmo é inicializado com a escolha **aleatória** de uma população, com um número certo de indivíduos, que é denominada de primeira geração. Esses indivíduos são espalhados (novamente de forma aleatória) em um espaço quadrado, com lados

bem definidos e centrado no ponto inicial x_0 da otimização, como mostra a figura 2. Os pontos azuis da figura representam os indivíduos da primeira população, e são o conjunto de pontos que será utilizado para otimizar a função-objetivo e encontrar o seu mínimo.

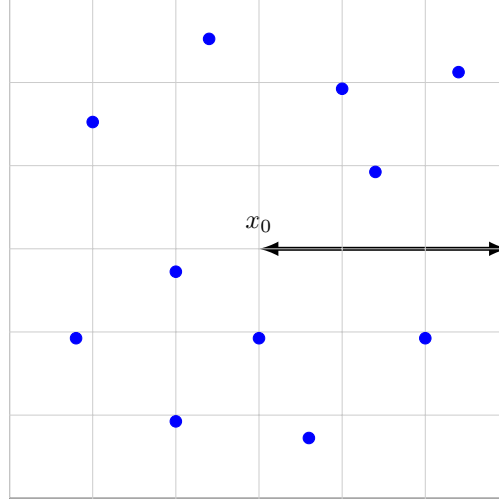


Figura 2: Território da primeira geração

O passo seguinte é calcular o valor de cada indivíduo com relação à função-objetivo. Como o algoritmo é baseado no evolucionismo, então o nosso objetivo é selecionar os indivíduos mais aptos dentro de uma dada população. Para isso, passaremos para a próxima etapa.

3.2 População intermediária

Para selecionar os indivíduos mais aptos dentro de uma população, o procedimento feito foi calcular a aptidão de cada indivíduo. De acordo com o algoritmo genético canônico, a **aptidão** ϕ_i é definida como a relação entre a média \bar{f} dos valores da população com relação a função-objetivo e o valor f_i do indivíduo i segundo a equação 3.1 [5].

$$\phi_i = \frac{\bar{f}}{f_i} \quad (3.1)$$

Então, se um indivíduo está longe do mínimo da função-objetivo, sua aptidão é pequena em relação à média, e se o indivíduo está perto do mínimo da função, sua aptidão é grande.

Dessa forma, podemos selecionar os indivíduos que são mais aptos para sobreviver nesse ambiente. A forma utilizada para realizar essa seleção foi a seguinte:

- Se $\phi < 1$, então a parte decimal indica a probabilidade que esse indivíduo terá de entrar na população intermediária. Por exemplo, se um indivíduo i possui $\phi_i = 0.745$, então ele terá uma chance de 74.5% de ser selecionado.

- Se $\phi > 1$, então a parte inteira indica o número de cópias que indivíduo terá na população intermediária e a parte decimal restante indica a chance de uma outra cópia ser selecionada. Como exemplo, se um indivíduo i possui $\phi_i = 8.163$, então ele terá 8 cópias do seu código genético garantido na população intermediária, e uma chance extra de 16.3% de conseguir uma nona cópia.

3.3 Segunda geração

Com a população intermediária selecionada, o próximo passo é realizar a recombinação. **Recombinação** é o processo de combinar os genes dos pais para gerar os filhos, que serão os integrantes da segunda geração.

Existem diversos tipos de recombinação para algoritmos genéticos, e nesse trabalho escolhemos fazer a recombinação utilizando a **média dos valores de 3 pais**. Isso significa que utilizamos as informações genéticas de três pontos aleatórios da população intermediária e fizemos a média entre esses pontos, resultando no primeiro filho, que é o primeiro indivíduo da segunda geração, e assim foi feito até recombinar toda a população. Como pode acontecer de o número de indivíduos não ser múltiplo de 3, então foi feita uma média com o resto da população, sendo de 1 ou 2 pontos [3].

A escolha de fazer a média de três pais e não de dois como acontece normalmente na natureza se baseou no fato de que, com mais pais, a população diminui mais aceleradamente, o que se traduz em um ganho de tempo para a convergência da busca do mínimo da função-objetivo.

Depois de realizados todos os passos, o algoritmo é refeito, utilizando dessa vez a população da segunda geração como a nova primeira geração, fechando o ciclo. O critério de parada do algoritmo genético construído foi feito da forma que, se o melhor indivíduo da geração atual (o que tem a melhor aptidão) tiver uma melhora de aptidão menor que a tolerância desejada na próxima geração, o ciclo é quebrado e o programa mostra o ponto mínimo encontrado.

A figura 3 ilustra um exemplo da primeira iteração da minimização da função $x^2 + y^2$, em que os pontos azuis representam a primeira geração, os vermelhos, a população intermediária e a segunda geração está na cor verde. É possível ver como os indivíduos começam a convergir para o ponto $(0, 0)$, que é o mínimo da função escolhida.

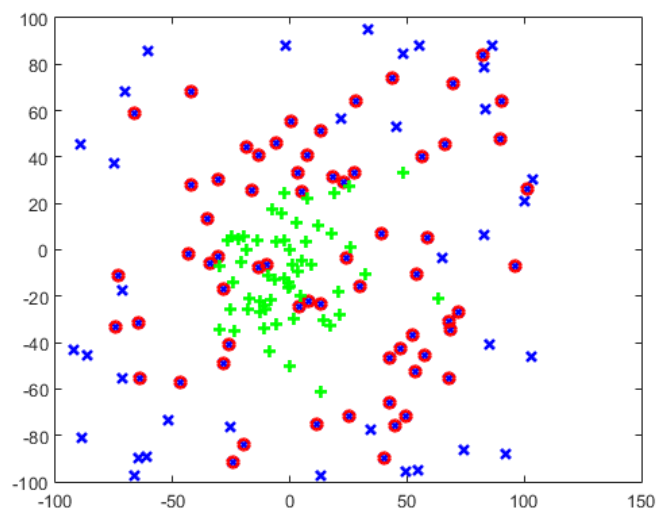


Figura 3: Exemplo de minimização da função $x^2 + y^2$

4 Simplex

O algoritmo *Simplex* é um método de minimização multidimensional sem restrições criado por John A. Nelder e Roger Mead, e publicado em 1965 na revista “The Computer Journal”. O método se baseia em utilizar um Simplex para minimizar uma função de n variáveis. Mas o que é um simplex? Um Simplex é o menor politopo possível para um espaço de n variáveis. Como exemplo, vemos a figura 4. Para \mathbb{R}^2 o menor politopo é um Triângulo, e para \mathbb{R}^3 é um Tetraedro.

Este é um método bem generalista que serve para diversos problemas pela sua facilidade, e por não necessitar de Gradientes e Hessianas, como veremos a seguir, é considerado um método de *ordem 0*.

Para facilitar a visualização, representação e entendimento, será explicado o método para \mathbb{R}^2 e fica a critério do leitor imaginar para situações com n variáveis.

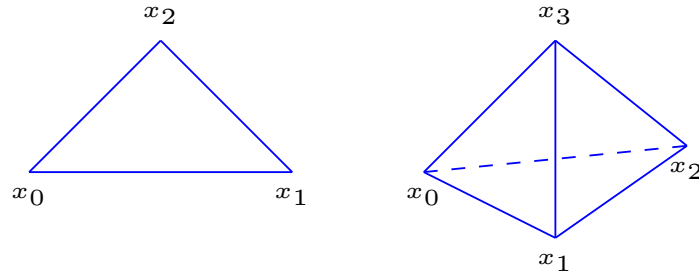


Figura 4: Exemplos de Simplex para \mathbb{R}^2 e \mathbb{R}^3 .

4.1 Ordenação

Com o (novo) Triângulo encontrado, a primeira coisa a se fazer é ordená-lo usando como critério o valor da função $f(x)$ para cada um de seus vértices. Utilizando a mesma nomenclatura de pontos que Nelder e Mead utilizaram em sua publicação, o vértice de menor valor será chamado x_l (de “low”), o com maior valor será chamado x_h (de “high”) e o outro ponto iremos chamar arbitrariamente de x_s .

4.2 Centróide

Conhecendo x_l , x_h e x_s , encontra-se o centróide entre todos os pontos excluindo o x_h . O centróide será calculado a partir de

$$c = \frac{x_l + x_s}{2} \quad (4.1)$$

4.3 Reflexão

Conhecendo x_l , x_h , x_s e c . Calcula-se a reflexão do ponto através da fórmula a seguir:

$$x_r = c + \alpha(c - x_h) \quad (4.2)$$

Onde α é chamado índice de reflexão, normalmente igual a 1. Podemos ver na figura 5 como realizar a operação graficamente.

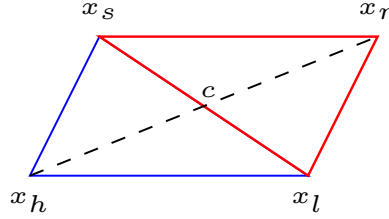


Figura 5: Representação da Reflexão.

Caso $f(x_l) < f(x_r) < f(x_s)$, x_h é substituído por x_r e a iteração é encerrada. Desse modo verifica-se se algum dos critérios de parada é correspondido e volta-se para a etapa de ordenação.

Caso difira cai-se em algum dos casos das subseções seguintes.

4.4 Expansão

Se $f(x_r) < f(x_l)$ faz-se a transformação de expansão através da seguinte equação

$$x_e = c + \gamma(x_r - c) \quad (4.3)$$

Onde γ é chamado coeficiente de expansão, normalmente igual a 2.

Podemos ver como realizar a transformação de expansão graficamente na figura 6.

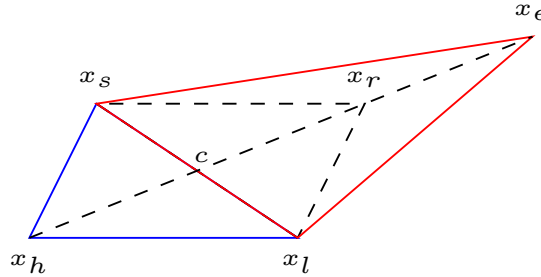


Figura 6: Representação da Expansão.

Para terminar a iteração x_h é substituído pelo que possui menor valor de função. Se $f(x_r) < f(x_e)$, aceita-se x_r , caso contrário x_e é aceito. Por sempre pegar o menor

valor entre os dois, acaba-se eventualmente aumentando o tamanho do triângulo, por isso as vezes o método é chamado de minimização gananciosa, sempre querendo mais e mais. Depois verifica-se se algum dos critérios de parada é correspondido e volta-se para a etapa de ordenação.

4.5 Contração

Se $f(x_r) \geq f(x_l)$ faz-se a transformação de Contração através das seguintes equações

$$x_c = \begin{cases} c + \beta(x_r - c) & \text{Se } f(x_s) \leq f(x_r) < f(x_h) \\ c + \beta(x_h - c) & \text{Se } f(x_r) > f(x_h) \end{cases} \quad (4.4a)$$

$$(4.4b)$$

Onde β é chamado coeficiente de Contração, normalmente igual a $\frac{1}{2}$. A contração onde x_c fica entre x_r e c será chamada de *Contração para fora* e a contração onde x_c fica entre x_h e c será chamada de *Contração para dentro*. Cada uma destas transformações podem ser vistas respectivamente da esquerda para direita na figura 7.

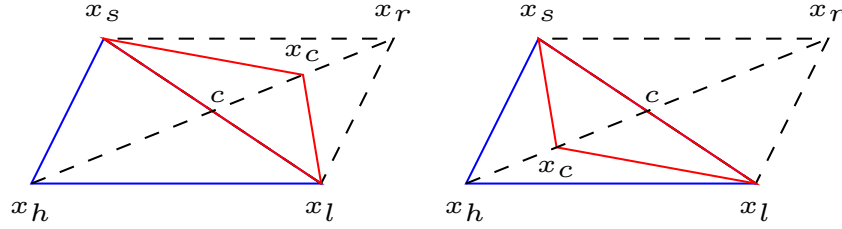


Figura 7: Representação das Contrações.

Para encontrar quem ira substituir x_h , faz-se algumas comparações pelo ponto mostrado nas equações

$$\text{Contração para fora} \begin{cases} \text{Substituir } x_h \text{ por } x_c & \text{Se } f(x_c) \leq f(x_r) \\ \text{Realizar encolhimento} & \text{Se } f(x_c) > f(x_r) \end{cases}$$

$$\text{Contração para dentro} \begin{cases} \text{Substituir } x_h \text{ por } x_c & \text{Se } f(x_c) < f(x_h) \\ \text{Realizar encolhimento} & \text{Se } f(x_c) \geq f(x_h) \end{cases}$$

Caso haja a substituição do ponto termina-se a iteração, verifica-se se algum dos critérios de parada é correspondido e volta-se para a etapa de ordenação.

4.6 Encolhimento

O encolhimento é muito raro de acontecer e é usado para quando todas as outras transformações anteriores resultam em pontos cujo valor correspondente na função é pior que os anteriores. O que só acontece em casos específicos. Pode ser visto um extrato da publicação original de 1965, [2]

A failed contraction is much rarer, but can occur when a valley is curved and one point of the simplex is much farther from the valley bottom than the others; contraction may then cause the reflected point to move away from the valley bottom instead of towards it. Further contractions are then useless. The action proposed contracts the simplex towards the lowest point, and will eventually bring all points into the valley.

Dessa forma a transformação é simplesmente trazer todos os pontos para mais perto do menor, o que pode ser descrito pela equação a seguir

$$x_s := x_s + \delta(x_s - x_l) \quad (4.5a)$$

$$x_h := x_h + \delta(x_h - x_l) \quad (4.5b)$$

Onde $:=$ indica uma substituição do ponto e δ é chamado coeficiente de encolhimento, normalmente igual a $\frac{1}{2}$.

A transformação pode ser vista na figura 8, onde em vermelho é o novo triângulo formado.

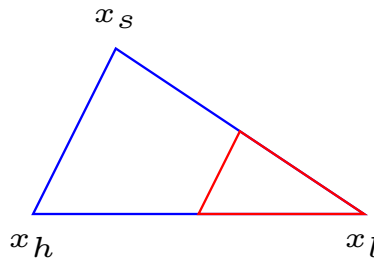


Figura 8: Representação do Encolhimento.

Depois de substituídos os pontos termina-se a iteração, verifica-se se algum dos critérios de parada é correspondido e volta-se para a etapa de ordenação.

4.7 Critérios de Parada

O primeiro critério de parada mais simples é o número de iterações realizadas, o que não precisa de explicações. O segundo critério, foi uma maneira de distinguir o quanto os pontos do triângulo estão próximos um do outro, assim encontra-se o circuncentro do triângulo e o raio da circunferência circunscrita denota a distância entre os vértices, quando ele for menor que um certo valor termina-se a minimização.

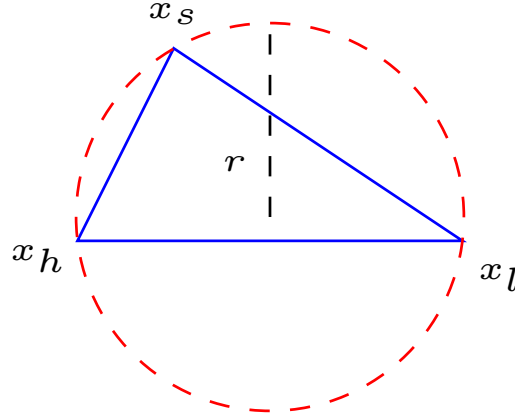


Figura 9: Circunferência circunscrita ao simplex.

Já o terceiro critério foi o mesmo utilizado na publicação original. Procura-se o desvio padrão de todos os valores de $f(x)$ encontrados de modo a ter uma pequena ideia de como é a superfície. Quando o desvio chega a zero significa que a superfície se comporta como um plano que não possui mínimo. então como eles dizem em [2]

The success of the criterion depends on the simplex not becoming too small in relation to the curvature of the surface until the final minimum is reached.

Escolhe-se um desvio tão pequeno quanto se queira para ser referência e uma vez que o desvio da superfície decresça e se torne menor que ele, termina-se a minimização

5 Interface Gráfica

Após desenvolver os métodos descritos nas seções 3 e 4 como funções do MATLAB, foram feitas algumas alterações na interface desenvolvida anteriormente [4] para englobar esses novos métodos, conforme ilustrado na figura 10

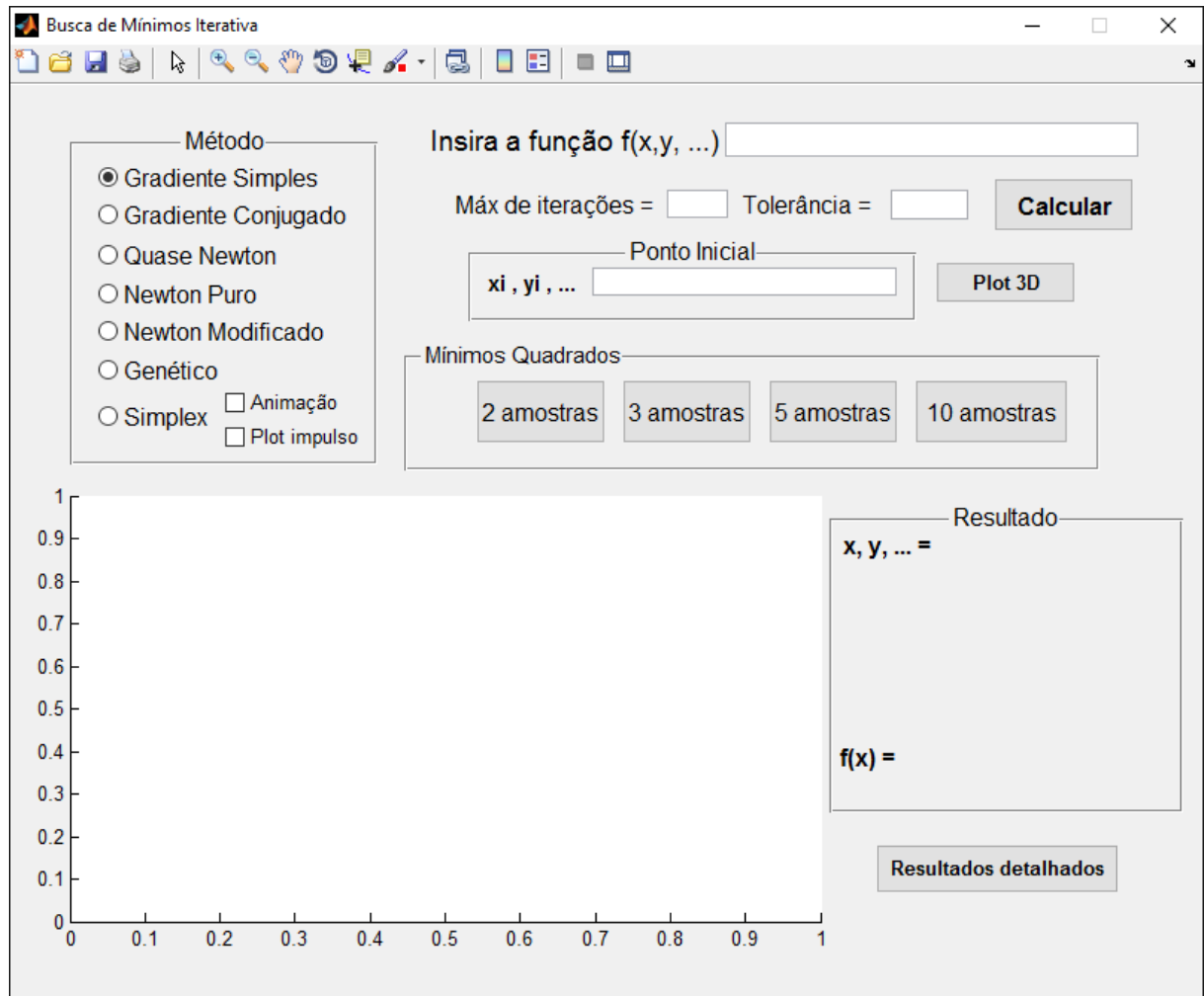


Figura 10: Janela do programa de Busca de Mínimos Iterativa.

Há um grupo de 4 novos botões, que servem para a resolução do problema de mínimos quadrados apresentado nesse trabalho. Ao clicar em um desses botões, a função de erro quadrático com 2, 3, 5 ou 10 amostras é automaticamente colocada na caixa de texto *Insira a função $f(x,y,...)$* . Assim, o usuário pode facilmente tentar encontrar os parâmetros desejados da função de transferência, testando para os diferentes métodos implementados.

Na aba *Método* houve a adição das 2 novas funções, e o método simplex também apresenta dois parâmetros extras. Ao marcar a opção *Animação*, o programa mostra cada passo da minimização pelo algoritmo simplex e o usuário consegue ver bem cada

etapa do processo, passando para a próxima com um clique do mouse ou pressionando qualquer tecla (incluindo efeitos sonoros). Já a opção *Plot Impulso* serve para quando a função de mínimos quadrados for a função minimizada, assim que o mínimo for encontrado, o programa mostra um gráfico da resposta ao impulso do sistema proposto nesse trabalho, com os coeficientes que acabaram de ser calculados.

6 Resultados e Discussão

Após implementados todos os métodos, foram testados para realizar a minimização quadrática segundo o enunciado:

2.) A resposta ao impulso unitário de um sistema foi medida em laboratório, resultando na tabela abaixo

t→	0.50	1.00	1.50	2.00	2.50	3.00	3.50	4.00	4.50	5.00
m(t)→	1.65	-1.30	0.50	-0.10	-0.15	0.15	-0.05	0.05	0.01	0.00

Deseja-se atribuir a este sistema um modelo LIT caracterizado por $H(s) = \omega_n^2 / (s^2 + 2\zeta\omega_n s + \omega_n^2)$ cuja resposta ao impulso unitário é $h(t) = (\omega_n^2 / \omega_d) e^{-\zeta\omega_n t} \sin(\omega_d t)$, onde $\omega_d = \omega_n \sqrt{1 - \zeta^2}$. Pelo método dos mínimos quadrados, encontrar a melhor estimativa para ζ e ω_n , usando (a) 03 das amostras acima, (b) 05 delas, (c) todas. Comente os resultados obtidos.

Fizemos as operações de encontrar a função objetivo usando os dados t e $m(t)$ da tabela e $h(t)$ e o método mostrado na seção 2, para 3, 5 e 10 amostras.

Cada uma das três funções objetivo foram testadas em todos os métodos de minimização vetorial implementados durante a realização da matéria de Introdução a Otimização, tantos os métodos “nobres” (Newton e Gradiente por exemplo) quanto nos métodos “pobres” mostrados neste documento. Infelizmente, todos os métodos “nobres” possuíam grande dificuldade, obtendo iterações que duravam minutos tornando assim impossível de computar a minimização.

Nos métodos pobres, O algoritmo genético utilizado neste trabalho mostrou-se eficiente para funções bem comportadas (quadráticas por exemplo), mas em funções menos comportadas, como a função objetivo para 3, 5 e 10 amostras, obteve semelhantemente dificuldades na hora de computar a minimização. Por isso tentou-se utilizar somente duas amostras, e mesmo assim as iterações demoravam dezenas de minutos cada, devido a explosão combinatória na hora da “reprodução”.

A seguir vemos os resultados obtidos para o simplex. As figuras 11, 12 e 13 mostram a convergência do método e as figuras 14, 15 e 16, mostram a resposta ao impulso unitário do sistema proposto pela minimização.

Para 3 amostras os valores de ω_n e ζ encontrados foram 4.8207 e 0.30037

Para 5 amostras os valores de ω_n e ζ encontrados foram 2.0521 e 0.30193

Para 10 amostras os valores de ω_n e ζ encontrados foram 4.9804 e 0.26358

Fim pelo número de Iterações

Iterações: 120/200

Tempo de simulação: 5.518

Coordenadas do mínimo: (4.8207,0.30037,0.0019568)

Fim pelo tamanho do Raio da Circunferência Circunscrita

Iterações: 69/200

Tempo de simulação: 3.628

Coordenadas do mínimo: (2.0521,0.30193,0.42709)

Fim pelo tamanho do Raio da Circunferência Circunscrita

Iterações: 71/200

Tempo de simulação: 3.837

Coordenadas do mínimo: (4.9804,0.26358,0.011948)

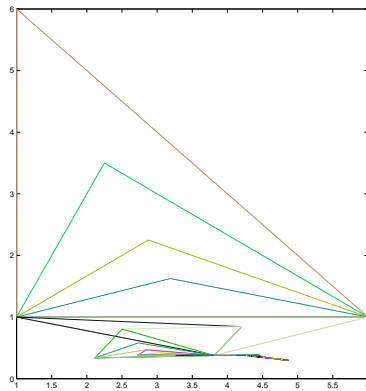


Figura 11: Convergência para 3 amostras

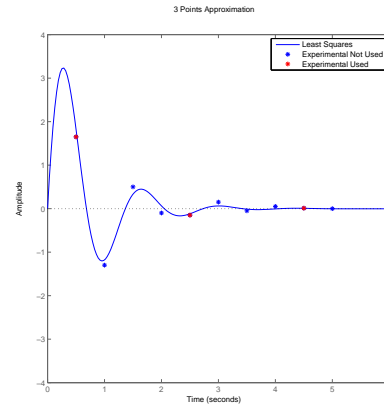


Figura 14: Resposta do impulso (3 amostras)

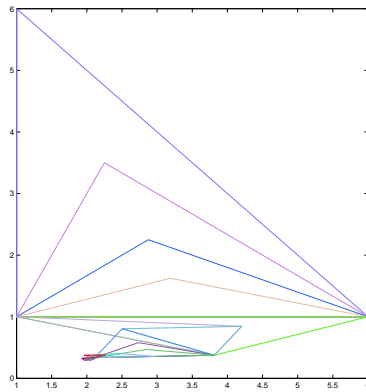


Figura 12: Convergência para 5 amostras

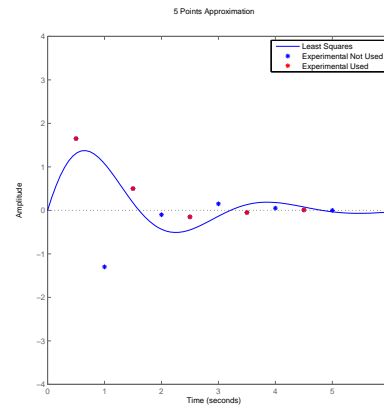


Figura 15: Resposta do impulso (5 amostras)

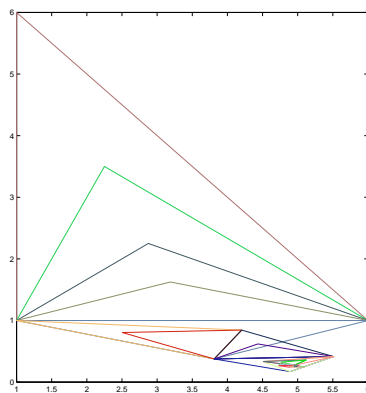


Figura 13: Convergência para 10 amostras

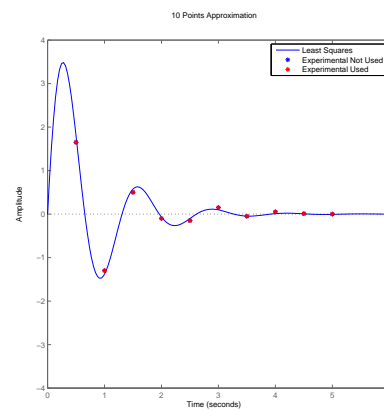


Figura 16: Resposta do impulso (10 amostras)

7 Conclusão

Os valores encontrados na seção 6 para a minimização do exercício proposto, quando comparados com os valores reais ($\omega_n = 5.0$ e $\zeta = 0.28$) encontram-se bem próximos, com discrepâncias menores que 10%.

Esses resultados evidenciam o poder dos métodos pobres, que apesar de geralmente exigirem um alto poder computacional, não mostram dificuldade em resolver problemas que os métodos nobres não conseguem, por necessitar de um cálculo simbólico complexo de derivadas e hessianas, o que resulta em um tempo de processamento muito grande.

Referências

- [1] R. LINDEN. *Algoritmos Genéticos (2a edição)*. BRASPORT.
- [2] John A Nelder and Roger Mead. A simplex method for function minimization. *The computer journal*, 7(4):308–313, 1965.
- [3] Chuan-Kang Ting. *Advances in Artificial Life: 8th European Conference, ECAL 2005, Canterbury, UK, September 5-9, 2005. Proceedings*, chapter On the Mean Convergence Time of Multi-parent Genetic Algorithms Without Selection, pages 403–412. Springer Berlin Heidelberg, Berlin, Heidelberg, 2005.
- [4] Cayo Valsamis, Gabriel Pelielo, Rodrigo Moysés, and Rafael Accácio. Trabalho 2 - introdução à otimização.
- [5] Darrell Whitley. A genetic algorithm tutorial. *Statistics and Computing*, 4(2):65–85.