



UNIVERSIDADE FEDERAL DE SANTA CATARINA  
CAMPUS BLUMENAU  
MESTRADO PROFISSIONAL EM MATEMÁTICA EM REDE NACIONAL

Matheus dos Santos Modesti

**OTIMIZAÇÃO SEM DERIVADAS:** Discussão e aplicabilidade no ensino básico

Blumenau  
2024

Matheus dos Santos Modesti

**OTIMIZAÇÃO SEM DERIVADAS:** Discussão e aplicabilidade no ensino básico

Dissertação submetida ao Mestrado Profissional em Matemática em Rede Nacional da Universidade Federal de Santa Catarina para a obtenção do título de Mestre em Matemática.

Orientador: Prof. Rafael Aleixo de Carvalho, Dr.

Blumenau  
2024

Ficha de identificação da obra elaborada pelo autor,  
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Modesti, Matheus dos Santos

Otimização Sem Derivadas : Discussão e aplicabilidade no ensino básico / Matheus dos Santos Modesti ; orientador, Rafael Aleixo de Carvalho, 2024.

79 p.

Dissertação (mestrado) - Universidade Federal de Santa Catarina, Campus Blumenau, Programa de Mestrado Profissional em Matemática em Rede Nacional - PROFMAT, Blumenau, 2024.

Inclui referências.

1. Matemática. 2. Otimização sem derivadas. 3. Matemática Aplicada. 4. Modelagem Matemática. I. Carvalho, Rafael Aleixo de. II. Universidade Federal de Santa Catarina. Programa de Mestrado Profissional em Matemática em Rede Nacional - PROFMAT. III. Título.

Matheus dos Santos Modesti

**OTIMIZAÇÃO SEM DERIVADAS:** Discussão e aplicabilidade no ensino básico

O presente trabalho em nível de mestrado foi avaliado e aprovado por banca examinadora composta pelos seguintes membros:

Prof. Rafael Aleixo de Carvalho, Dr.  
Universidade Federal de Santa Catarina

Prof. Gabriel Haeser, Dr.  
Universidade de São Paulo

Prof. Luiz Rafael dos Santos, Dr.  
Universidade Federal de Santa Catarina

Certificamos que esta é a **versão original e final** do trabalho de conclusão que foi julgado adequado para obtenção do título de Mestre em Matemática.

---

Coordenação do Programa de  
Pós-Graduação

---

Prof. Rafael Aleixo de Carvalho, Dr.  
Orientador

Blumenau, 2024.



Para minha mãe.

## **AGRADECIMENTOS**

Primeiramente, gostaria de expressar minha profunda gratidão a Deus pelo dom da vida, sabedoria e paciência que Ele me concedeu.

À minha querida mãe, que, lá do Céu, continua a zelar por mim e a me capacitar a cada dia, e ao meu pai, cujos esforços incansáveis fizeram com que eu pudesse concluir todas as etapas dos meus estudos.

Não posso deixar de agradecer a todos os dedicados professores do meu curso. Em particular, quero expressar minha sincera gratidão ao Professor Rafael Aleixo, que generosamente aceitou ser meu orientador e me apoiou durante toda esta jornada.

Agradeço a todos os meus colegas de mestrado. Nossa convivência foi repleta de momentos inesquecíveis e aprendizados valiosos.

E, é claro, aos meus amigos, quero expressar meu eterno agradecimento por estarem sempre ao meu lado, compartilhando os altos e baixos da vida. Sua amizade é um presente inestimável.

## RESUMO

A otimização sem derivadas é a área matemática aplicada que busca otimizar funções com técnicas que não façam uso da diferenciação. Seus estudos iniciaram na década de 1960, e evoluíram nos anos posteriores, dando origem a vários métodos e aplicações. São estudados os métodos de otimização de  $\mathbb{R}^2$  em  $\mathbb{R}^2$  baseados em simplex (Nelder-Mead e Busca Multidirecional) e de Busca Direta Direcionais (Busca Direta e Busca Direta Direcional), e alguns métodos de otimização unidimensional (Busca pela Seção Áurea e Interpolação Parabólica Sucessiva). Sua teoria é visitada, com a listagem dos principais teoremas que garantem sua convergência e resultados, com algumas demonstrações e a explicação de seus algoritmos. Os métodos de Nelder-Mead e da Busca Direta são testados em algumas funções, bem como o método da Busca pela Seção Áurea e da Interpolação Parabólica Sucessiva, e seus resultados de iterações necessárias, erro e tempo de máquina, comparados. Por fim, traça-se um paralelo entre a modelagem matemática, a resolução de problemas, a otimização e a Base Nacional Comum Curricular (BNCC), e propõe-se uma sequência didática para sua utilização no ensino médio, explorando as potencialidades e interações significativas entre estes agentes do ensino da matemática.

**Palavras-chave:** Otimização sem derivadas. Matemática Aplicada. Modelagem Matemática.

## ABSTRACT

Derivative-free optimization is the applied mathematical field that seeks to optimize functions using techniques that do not rely on differentiation. Its studies began in the 1960s and evolved in the subsequent years, giving rise to various methods and applications. The methods for optimizing  $\mathbb{R}^2$  in  $\mathbb{R}^2$  based on simplex (Nelder-Mead and Multidirectional Search) and Directional Direct Search (Direct Search and Directional Direct Search) are examined, as well as some one-dimensional optimization methods (Golden Section Search and Successive Parabolic Interpolation). Its theory is explored, listing the main theorems guaranteeing its convergence and results, along with some demonstrations and an explanation of its algorithms. The Nelder-Mead and Direct Search methods are compared, as well as the Golden Section Search and Successive Parabolic Interpolation methods, in terms of required iterations, error, and machine time. Finally, a connection is drawn between mathematical modeling, problem solving, optimization, and the Common National Curriculum Base (BNCC), and a didactic sequence is proposed for its use in high school education, exploring the potential and meaningful interactions among these elements in the teaching of mathematics.

**Keywords:** Derivative-free optimization. Applied Mathematics. Mathematical Modeling.

## LISTA DE FIGURAS

Figura 2.1 – Funcionamento do Método de Nelder Mead . . . . .	16
Figura 2.2 – Funcionamento do Método de Nelder Mead (1) . . . . .	18
Figura 2.3 – Funcionamento do Método de Nelder Mead (2) . . . . .	20
Figura 2.4 – Funcionamento do Método de Nelder Mead (3) . . . . .	21
Figura 2.5 – Funcionamento do Método de Nelder Mead (4) . . . . .	22
Figura 2.6 – Funcionamento do Método de Nelder Mead (5) . . . . .	23
Figura 2.7 – Possibilidades da Busca Multidirecional . . . . .	25
Figura 3.1 – Método da Busca Direta . . . . .	29
Figura 3.2 – Aproximação do Mínimo pelas Iterações no Método da Busca pela Seção Áurea . . . . .	39
Figura 3.3 – Aproximação do Mínimo pelas Iterações no Método da Interpolação Parabólica Sucessiva . . . . .	41
Figura 4.1 – Função de Himmelblau . . . . .	44
Figura 4.2 – Curvas de Nível da Função de Himmelblau . . . . .	45
Figura 4.3 – Função de Beale . . . . .	46
Figura 4.4 – Curvas de Nível da Função de Beale . . . . .	46
Figura 4.5 – Função de Booth . . . . .	47
Figura 4.6 – Curvas de Nível da Função de Booth . . . . .	48
Figura 4.7 – Função Three-Hump Camel . . . . .	48
Figura 4.8 – Curvas de Nível da Função Three-Hump Camel . . . . .	49
Figura 4.9 – Primeira Função . . . . .	51
Figura 4.10 – Segunda Função . . . . .	52

## LISTA DE QUADROS

Quadro 4.1 – Métodos de Nelder-Mead e Busca Coordenada aplicados à Função de Himmelblau . . . . .	45
Quadro 4.2 – Métodos de Nelder-Mead e Busca Coordenada aplicados à Função de Beale . . . . .	47
Quadro 4.3 – Métodos de Nelder-Mead e Busca Coordenada aplicados à Função de Booth . . . . .	47
Quadro 4.4 – Métodos de Nelder-Mead e Busca Coordenada aplicados à Função Three-Hump Camel . . . . .	49
Quadro 4.5 – Métodos Busca pela Seção Áurea e da Interpolação Parabólica Sucessiva aplicados à primeira função . . . . .	51
Quadro 4.6 – Métodos Busca pela Seção Áurea e da Interpolação Parabólica Sucessiva aplicados à segunda função . . . . .	53

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>12</b>
<b>2</b>	<b>MÉTODOS BASEADOS EM SIMPLEX</b>	<b>15</b>
2.1	ALGORITMO DE NELDER-MEAD	15
2.2	BUSCA MULTIDIRECIONAL	25
<b>3</b>	<b>MÉTODOS DE BUSCA DIRETA DIRECIONAIS</b>	<b>27</b>
3.1	BUSCA COORDENADA	28
3.2	BUSCA DIRETA DIRECIONAL	30
3.3	CONVERGÊNCIA DOS MÉTODOS DE BUSCA DIRETA DIRECIONAIS	32
<b>3.3.1</b>	<b>Comportamento do parâmetro de passos</b>	<b>32</b>
<b>3.3.2</b>	<b>Conjunto Arbitrário de Bases Positivas</b>	<b>33</b>
<b>3.3.3</b>	<b>Conjunto Finito de Bases Positivas</b>	<b>36</b>
3.4	OTIMIZAÇÃO UNIDIMENSIONAL	37
<b>3.4.1</b>	<b>Busca pela Seção Áurea</b>	<b>37</b>
<b>3.4.2</b>	<b>Interpolação Parabólica Sucessiva</b>	<b>40</b>
<b>4</b>	<b>ALGUMAS IMPLEMENTAÇÕES E COMPARAÇÕES ENTRE OS MÉTODOS</b>	<b>43</b>
4.1	FUNÇÕES DE $\mathbb{R}^2$ EM $\mathbb{R}$	43
<b>4.1.1</b>	<b>Função de Himmelblau</b>	<b>44</b>
<b>4.1.2</b>	<b>Função de Beale</b>	<b>45</b>
<b>4.1.3</b>	<b>Função de Booth</b>	<b>46</b>
<b>4.1.4</b>	<b>Função Three-Hump Camel</b>	<b>48</b>
<b>4.1.5</b>	<b>Análise Geral</b>	<b>49</b>
4.2	FUNÇÕES DE $\mathbb{R}$ EM $\mathbb{R}$	50
<b>4.2.1</b>	<b>Primeira Função</b>	<b>51</b>
<b>4.2.2</b>	<b>Segunda Função</b>	<b>52</b>
<b>5</b>	<b>ENSINO, A BNCC E OTIMIZAÇÃO</b>	<b>54</b>
5.1	OTIMIZAÇÃO E A BNCC	55
<b>6</b>	<b>PROPOSTA DE SEQUÊNCIA DIDÁTICA</b>	<b>59</b>
6.1	AULA 1 E 2: INTRODUÇÃO AO CONCEITO DE FUNÇÃO	59
6.2	AULA 3 E 4: GRÁFICOS DE FUNÇÕES	60
6.3	AULA 5 E 6: FUNÇÕES INVERSA, INJETIVIDADE, SOBREJETIVIDADE E BIJETIVIDADE	61
6.4	AULA 7: OTIMIZAÇÃO COM FUNÇÕES	61
6.5	AULA 8: TÉCNICAS DE OTIMIZAÇÃO DE $\mathbb{R}$ EM $\mathbb{R}$	62
6.6	AULA 9 E 10: TÉCNICAS DE OTIMIZAÇÃO DE $\mathbb{R}^2$ EM $\mathbb{R}$	62
6.7	AULA 11: EXERCÍCIOS PRÁTICOS DE OTIMIZAÇÃO	63
6.8	AULA 12: PROGRAMAÇÃO DOS MÉTODOS ESTUDADOS	64

6.9	AULA 13 E 14: REVISÃO E AVALIAÇÃO . . . . .	65
6.10	CONCLUSÃO DA SEQUÊNCIA DIDÁTICA . . . . .	66
<b>7</b>	<b>CONCLUSÃO . . . . .</b>	<b>68</b>
	<b>REFERÊNCIAS . . . . .</b>	<b>70</b>
	<b>APÊNDICE A – CÓDIGOS DOS ALGORITMOS . . . . .</b>	<b>72</b>
A.1	ALGORITMO DE NELDER MEAD APLICADO À FUNÇÃO DE HIM- MELBLAU . . . . .	72
A.2	ALGORITMO DA BUSCA DIRETA APLICADO À FUNÇÃO DE HIM- MELBLAU . . . . .	74
A.3	ALGORITMO DA BUSCA PELA SEÇÃO ÁUREA APLICADO À PRI- MEIRA FUNÇÃO . . . . .	76
A.4	ALGORITMO DA INTERPOLAÇÃO PARABÓLICA SUCESSIVA APLI- CADO À PRIMEIRA FUNÇÃO . . . . .	77



## 1 INTRODUÇÃO

A otimização é uma área de estudo da matemática aplicada que se concentra na busca da melhor solução possível, chamada de solução ótima, para um problema específico, dadas as restrições e objetivos estabelecidos. Em essência, a otimização envolve a tarefa de encontrar o valor mínimo (ou máximo) de uma função, chamada de função objetivo, sujeita a limitações ou restrições.

Dada uma função  $f$ , temos o problema:

$$\min f(x), \quad \text{sujeita a } x \in S \quad (1.1)$$

onde  $S \subset \mathbb{R}^n$  e  $f : S \rightarrow \mathbb{R}$ . Um ponto  $x^*$  é dito *minimizador local* de  $f$  em  $S$  se e somente se existir um  $\varepsilon > 0$  de modo que  $f(x) > f(x^*)$ , para todo  $x \in S \cap B(x^*, \varepsilon)$ . Particularmente,  $x^*$  será um *minimizador global* de  $f$  em  $S$  se e somente se  $f(x) \geq f(x^*)$ , para todo  $x \in S$ , e dizemos que  $x^*$  é uma *solução ótima* de (1.1) (DINIZ-EHRHARDT; LOPES; PEDROSO, 2012).

Os métodos de otimização com derivadas, como o método do gradiente, baseiam-se no cálculo das derivadas da função a ser minimizada. Elas fornecem informações sobre a taxa de variação da função em diferentes pontos do espaço de busca, permitindo direcionar a busca para regiões onde se espera encontrar o minimizador. No entanto, esses métodos pressupõe a existência e o cálculo das derivadas da função, o que pode ser uma tarefa complexa e computacionalmente intensiva, especialmente em problemas em altas dimensões, com muitas variáveis ou com funções não diferenciáveis.

No entanto, nesses mesmos contextos, quando existe um problema a ser otimizado, pode-se recorrer aos métodos de otimização sem derivadas, também chamados de métodos de otimização sem derivadas. Estes métodos, que não dependem do cálculo das derivadas da função, exploram a função objetivo apenas avaliando-a em diferentes pontos do espaço de busca e tomando decisões com base nos resultados obtidos. Essa área tem recebido bastante atenção dos pesquisadores desde a década de 1960, quando iniciaram seus estudos. Sua teoria é robusta e apresenta resultados sólidos, comparáveis aos métodos tradicionais (DINIZ-EHRHARDT; LOPES; PEDROSO, 2012).

Alguns exemplos de situações em que as derivadas não estão disponíveis, conforme mencionados por Conn, Scheinberg e Vicente (2009), incluem problemas de engenharia, como o design das asas de um rotor de helicóptero, de uma plataforma voadora, de uma forma aeroacústica e problemas de hidrodinâmica. Outros exemplos são o design de circuitos, geometria molecular, problemas relacionados a água subterrânea, registro de imagens médicas e dinâmica de preços.

A justificativa de Diniz-Ehrhardt, Lopes e Pedroso (2012) para o estudo e o uso de métodos sem derivada é a possibilidade de aplicação rápida, sem a necessidade

de cálculos extensos e complexos, como gradientes ou hessianas. Geralmente, os métodos de otimização não diferenciáveis são mais simples de implementar e não exigem conhecimento prévio das propriedades matemáticas da função. No entanto, a desvantagem desses métodos é que eles podem exigir um número maior de avaliações da função para encontrar o mínimo, tornando-os menos eficientes em termos computacionais, especialmente em problemas com alta dimensionalidade.

Ainda segundo os autores, o estudo dessa classe de métodos de otimização teve início na década de 1960 com várias publicações importantes, como as de Hooke e Jeeves em 1961, Spendley, Hext e Himsforth em 1962, e Nelder e Mead em 1965. Nas décadas seguintes, foram realizadas diversas outras publicações relevantes, incluindo os trabalhos de Polak em 1971, Torczon em 1997, McKinnon em 1998, Powell em 2002 e Audet e Dennis em 2006 (DINIZ-EHRHARDT; LOPES; PEDROSO, 2012). As duas principais referências bibliográficas deste texto são Conn, Scheinberg e Vicente em 2009, e Diniz-Ehrhardt, Lopes e Pedroso em 2012.

Embora tenham sido realizadas várias pesquisas sobre métodos de otimização sem derivadas, até o momento, pouca atenção foi dada às possibilidades de aplicação desses métodos no contexto do ensino regular nacional. A maioria dos estudos concentra-se na teoria ou em aplicações específicas, como o trabalho de Giuliani (2013), que utiliza otimização sem derivadas no estudo da maximização de campos de petróleo. Portanto, existe uma oportunidade promissora para investigar e explorar o potencial dessa modalidade de otimização no ensino básico, especialmente no ensino médio.

Adicionalmente, é importante ressaltar que os problemas de otimização abordados neste trabalho são irrestritos, ou seja, não envolvem limitações específicas que restrinjam as soluções viáveis. A otimização com restrições é uma área mais complexa, onde as soluções devem satisfazer uma série de condições adicionais, tornando o processo de busca pela solução ótima ainda mais desafiador para os métodos empregados. No entanto, ao focarmos em problemas irrestritos, podemos explorar os conceitos fundamentais da otimização de forma mais acessível e ampla, atingindo os objetivos desejados.

A Base Nacional Comum Curricular (BNCC) estabelece, no âmbito da área de Matemática e Suas Tecnologias, a importância de abordar o tema da otimização de funções e incentiva o uso de métodos computacionais ou softwares para resolver esses problemas. No entanto, no ensino médio, não são abordados as derivadas, que são comumente utilizados para resolver problemas de otimização de funções. Nesse sentido, este trabalho destaca a relevância dos métodos de otimização sem o uso de derivadas, abrangendo exatamente essa temática e suprimindo a proposta no currículo atual.

Desse modo, o objetivo principal deste trabalho é estudar as possibilidades de

utilização de métodos de otimização sem derivadas no contexto do ensino médio no Brasil e, ao final, apresentar uma Proposta de Sequência Didática para sua implementação.

Para atingir esse objetivo, a dissertação foi estruturada em diferentes capítulos, a saber:

- Capítulo 2: neste capítulo, é realizado um estudo aprofundado dos métodos de otimização sem derivadas baseados em simplex. São exploradas as principais características e é estudada sua convergência;
- Capítulo 3: neste momento, o foco é direcionado aos métodos de otimização sem derivadas de busca direta direcionais. São examinados em detalhes de seus princípios fundamentais e provada sua convergência para determinados casos;
- Capítulo 4: este capítulo concentra-se na implementação computacional dos métodos estudados. São realizados testes e experimentos para avaliar o desempenho e a eficiência dos métodos de otimização sem derivadas em diferentes funções;
- Capítulo 5: aqui, são discutidas as metodologias de modelagem matemática e resolução de problemas, além de abordar a relação entre os tópicos da BNCC e a otimização matemática. São exploradas possibilidades de como introduzir os conceitos de otimização sem derivadas no ensino médio de maneira relevante e acessível;
- Capítulo 6: nesta etapa, é apresentada uma proposta de sequência didática para o estudo de funções e otimização no ensino médio, utilizando os conceitos e métodos de otimização sem derivadas. São sugeridas abordagens e recursos educacionais que podem auxiliar no processo de ensino-aprendizagem desses temas;
- Capítulo 7: no último capítulo, são apresentadas as conclusões do trabalho, recapitulando os principais pontos abordados ao longo da dissertação e destacando as contribuições e possíveis direções futuras de pesquisa nessa área.

A estrutura dos capítulos foi pensada de forma a fornecer uma visão abrangente e detalhada dos métodos de otimização sem derivadas, permitindo uma compreensão aprofundada do tema, e sua aplicabilidade educacional.

## 2 MÉTODOS BASEADOS EM SIMPLEX

Nesta seção, serão apresentados dois métodos de otimização sem derivadas que se baseiam em modelos simplex: o método de Nelder-Mead e a Busca Multidirecional.

Dado um conjunto de pontos  $\{x_1, x_2, \dots, x_{n+1}\} \subset \mathbb{R}^n$ , de  $n + 1$  pontos afavelmente independentes, isto é, que o conjunto  $\{x_1 - x_{n+1}, x_2 - x_{n+1}, \dots, x_n - x_{n+1}\}$  seja linearmente independente, definimos o simplex associado a esses pontos como o poliedro formado por eles. Esses pontos são chamados de vértices do simplex (BOYD; VANDENBERGHE, 2004).

Por exemplo, um simplex de dimensão zero é um ponto, de dimensão um, um segmento de reta, de dimensão dois, um triângulo, de dimensão três um tetraedro, e assim por diante.

A seguir, detalharemos o funcionamento de dois métodos: o Método de Nelder-Mead e o método da Busca Multidirecional. Explicaremos como esses métodos operam, apresentaremos seus algoritmos e discutiremos sua convergência. A implementação e as aplicações práticas serão abordadas em capítulos posteriores.

### 2.1 ALGORITMO DE NELDER-MEAD

O algoritmo de Nelder-Mead (NELDER; MEAD, 1965), um dos primeiros métodos de otimização sem derivadas desenvolvido, é amplamente utilizado por toda a comunidade. De acordo com Diniz-Ehrhardt, Lopes e Pedroso (2012, p. 31), sua popularidade se deve à sua “simplicidade e à sua capacidade de se adaptar à curvatura da função a ser minimizada”.

O método consiste em avaliar a função em um número finito de pontos e, com base nesses valores, tomar decisões sobre os próximos passos. Cada iteração no espaço  $\mathbb{R}^n$  é baseada nos  $n+1$  vértices  $X = \{x_1, x_2, \dots, x_{n+1}\}$  de um simplex, ordenados de acordo com os valores crescentes da função  $f$ .

Em seguida, o método busca substituir o pior vértice do simplex (o que possui o maior valor de  $f$ ) refletindo-o em relação ao centroide da face oposta. Para isso, são utilizados quatro coeficientes escalares:

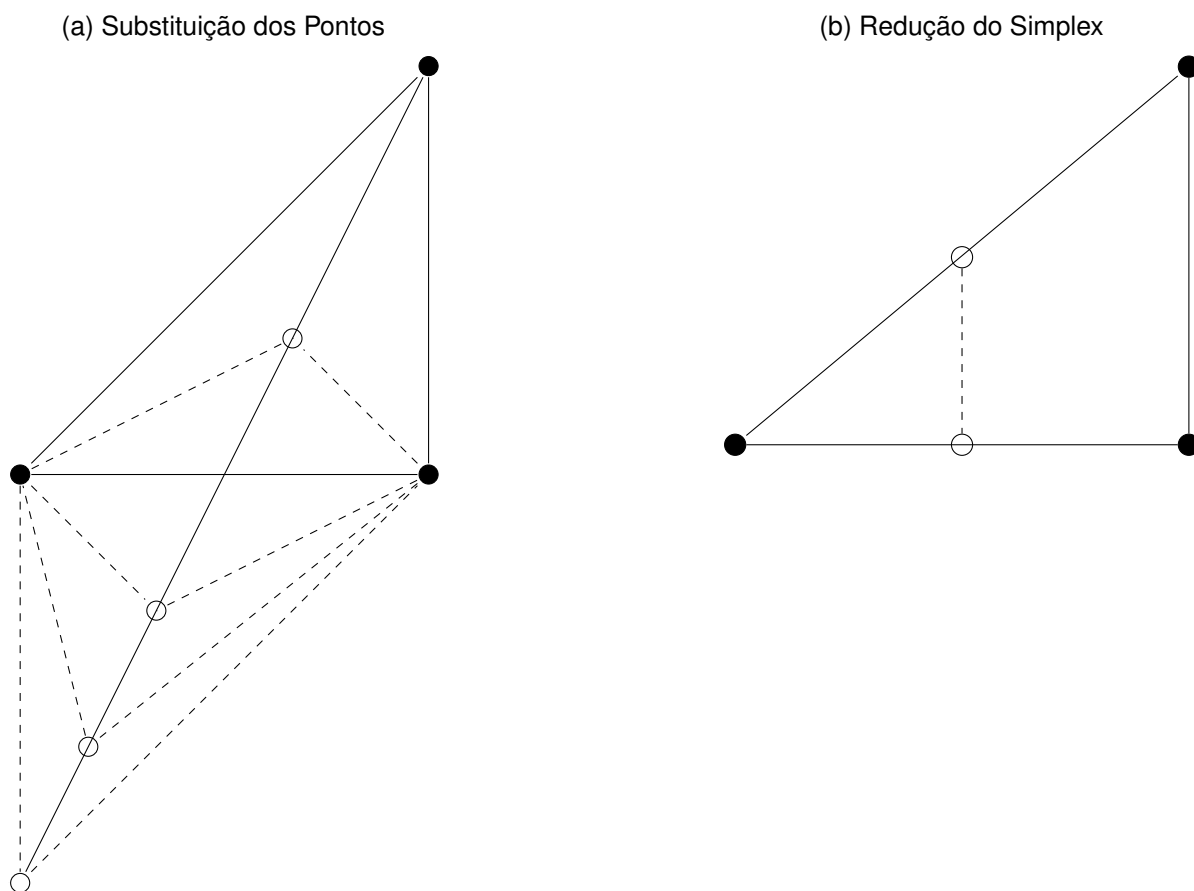
- Coeficiente de reflexão:  $\rho > 0$ ;
- Coeficiente de expansão:  $\chi > 1$  com  $\chi > \rho$ ;
- Coeficiente de retração:  $0 < \gamma < 1$  e;
- Coeficiente de redução:  $0 < \sigma < 1$ .

Na prática, é comum definir  $\rho = 1$ , o que resulta em uma reflexão exata;  $\chi = 2$ , dobrando a distância entre o novo vértice e o centroide da face oposta;  $\gamma = 0,5$ , reduzindo as distâncias nas contrações pela metade; e  $\sigma = 0,5$ , reduzindo pela metade

a distância do melhor vértice em caso de redução do simplex (DINIZ-EHRHARDT; LOPES; PEDROSO, 2012).

Na Figura 2.1a, pode-se observar o processo de substituição do pior ponto pelos métodos de redução, expansão e contração interna e externa. Já na Figura 2.1b, é apresentada a possibilidade de redução do simplex. Os pontos em preto representam os pontos da iteração atual, enquanto os pontos não pintados representam as possibilidades de novos pontos.

Figura 2.1 – Funcionamento do Método de Nelder Mead



Fonte: Adaptado de Conn, Scheinberg e Vicente (2009) e Diniz-Ehrhardt, Lopes e Pedroso (2012).

Para proceder com as iterações, os vértices do simplex são ordenados em ordem crescente, com base no valor da função objetivo. Em seguida, é calculado o centroide da face oposta do simplex, denotado por  $\bar{x}$ , que é a média aritmética dos vértices.

O método inicia tentando refletir o pior vértice do simplex em relação ao centroide, gerando o ponto de reflexão  $x_r$ , calculada por

$$x_r = (1 + \rho)\bar{x} - \rho x_{n+1}.$$

Se o valor da função avaliada nesse ponto for menor (podemos dizer melhor) do que o do pior vértice, uma expansão é realizada para explorar ainda mais essa direção.

O ponto de expansão  $x_e$  é obtido da seguinte forma:

$$x_e = (1 + \rho\chi)\bar{x} - \rho\chi x_{n+1}.$$

Caso a expansão não resulte em uma melhoria significativa, é feita uma contração para buscar um ponto mais próximo ao centroide. Os pontos da contração externa  $x_{ce}$  e interna  $x_{ci}$  são calculados, respectivamente por:

$$x_{ce} = (1 + \rho\gamma)\bar{x} - \rho\gamma x_{n+1}, \quad \text{e} \quad x_{ci} = (1 - \gamma)\bar{x} + \gamma x_{n+1}.$$

Se nenhuma melhoria for alcançada com as operações anteriores, é realizada uma redução do simplex. Nessa etapa, são calculados pontos  $v_i$  a partir do vértice melhor (primeiro vértice) e do pior vértice, utilizando um fator de redução  $\sigma$  da seguinte forma:

$$v_i = x_1 + \sigma(x_i - x_1), \quad i = \{2, \dots, n + 1\}.$$

Esses pontos são adicionados à lista de vértices para a próxima iteração.

O critério de parada do algoritmo é atingir um tamanho de diâmetro específico do simplex, denotado como  $\Delta_{tol}$ . O diâmetro do simplex é calculado como a maior distância entre dois vértices dele (CONN; SCHEINBERG; VICENTE, 2009).

O Algoritmo 2.1, adaptado de Diniz-Ehrhardt, Lopes e Pedroso (2012) mostra o funcionamento do método.

#### Algoritmo 2.1: Algoritmo de Nelder-Mead

**Dados:** Seja  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  e um simplex no  $\mathbb{R}^n$ . Escolha  $n$  pontos  $x_k$ , para  $k \in \{1, 2, \dots, n + 1\}$ . Além disso, defina os parâmetros  $\rho$ ,  $\chi$ ,  $\gamma$  e  $\sigma$ .

**Resultado:** Minimizar  $f$

```

1 enquanto  $\Delta > \Delta_{tol}$  faça
2   Ordenar os vértices e calcule o centroide dos  $n$  melhores pontos;
3   Calcular o ponto de reflexão;
4   se  $f(x_1) \leq f(x_r) < f(x_n)$  então
5     Aceite o ponto  $x_r$  e termine a iteração;
6   senão
7     Caso  $f(x_r) < f(x_1)$ , calcular o ponto de expansão. Se  $f(x_e) < f(x_r)$ ,
       aceite  $x_e$  e termine a iteração. Caso contrário, aceite  $x_r$  e termine a
       iteração.;
8     Caso  $f(x_r) \geq f(x_n)$  calcular o ponto de contração: se
        $f(x_n) \leq f(x_r) < f(x_{n+1})$ , calcular o ponto de contração externa. Se
        $f(x_c) < f(x_{n+1})$ , aceite  $x_{ce}$  e termine a iteração. Se  $f(x_r) \geq f(x_{n+1})$ ,
       calcule o ponto de contração interna. Se  $f(x_c) < f(x_{n+1})$ , aceite  $x_{ci}$  e
       termine a iteração;
9     Caso contrário, realizar uma redução do simplex.
10  fim
11 fim

```

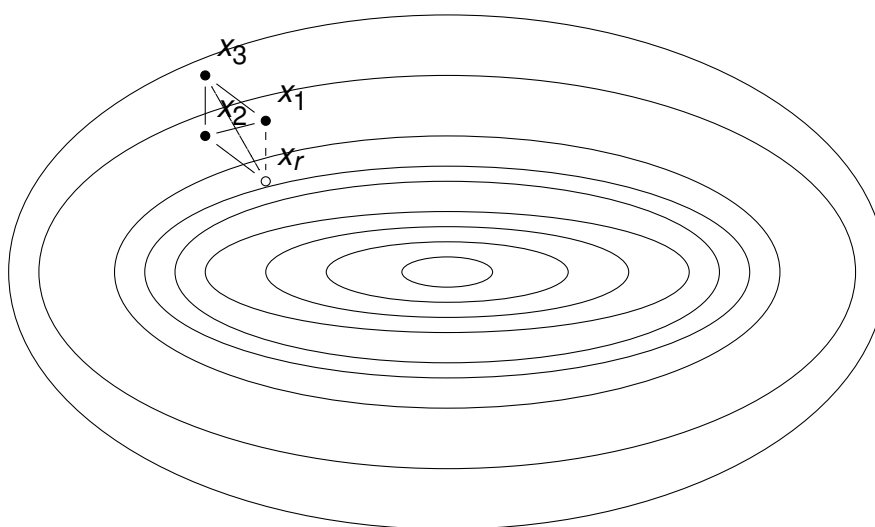
A sequência de figuras a seguir ilustra o funcionamento de algumas iterações do modelo. As elipses representam curvas de nível de uma função arbitrária, respeitando as hipóteses do algoritmo.

Para iniciar as iterações, procedemos ao cálculo da reflexão do pior ponto no simplex. Neste caso específico, o ponto  $x_3$  é identificado como o pior ponto.

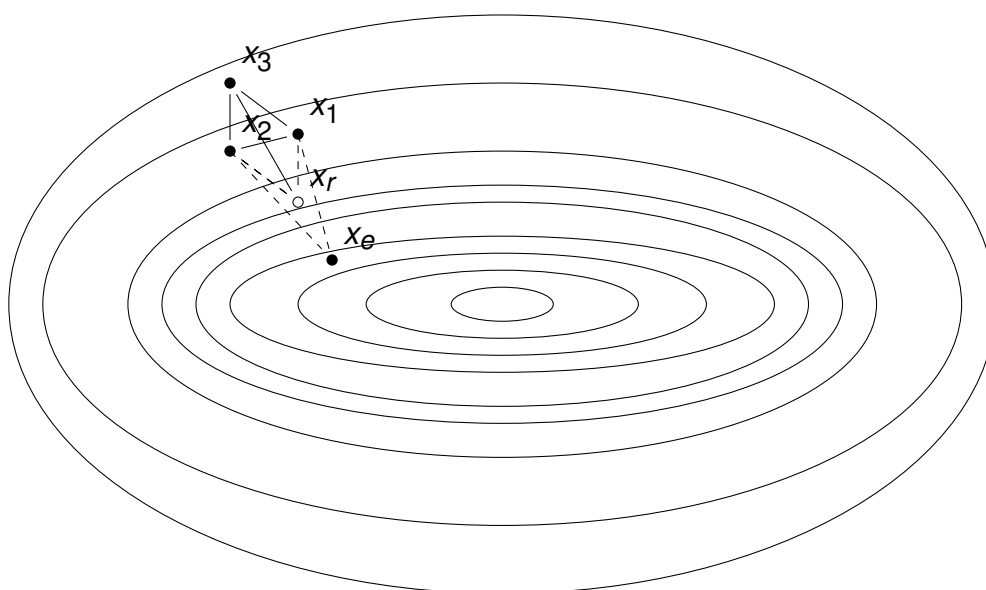
Para realizar a reflexão, utilizamos o centroide da face oposta como ponto de referência. Essa reflexão é representada na Figura 2.2a, em que podemos observar o ponto resultante,  $x_r$ , obtido após a reflexão do ponto  $x_3$ .

Figura 2.2 – Funcionamento do Método de Nelder Mead (1)

(a) Reflexão do pior ponto do simplex



(b) Expansão do pior ponto do simplex



Fonte: Elaborado pelo autor.

Ao realizar a reflexão do pior ponto, obtemos um novo ponto,  $x_r$ , que se revela melhor em relação ao melhor ponto anteriormente avaliado,  $x_1$ . Essa reflexão

é ilustrada na Figura 2.2b. Dessa forma, buscando explorar ainda mais o espaço de soluções, experimentaremos uma expansão.

A expansão consiste em ampliar a distância entre o ponto  $x_r$  e o centroide da face oposta, na esperança de encontrar um ponto ainda mais favorável. A Figura 2.2b ilustra o resultado dessa expansão, apresentando o ponto  $x_e$ , que se mostra ainda melhor do que  $x_r$ , o ponto obtido pela reflexão.

Essa estratégia de expansão visa aumentar a exploração do espaço de busca, permitindo que o modelo explore regiões mais amplas e potencialmente encontre soluções melhores.

Após a expansão, o próximo simplex a ser utilizado na iteração seguinte será composto pelos pontos  $x_1$ ,  $x_2$  e  $x_e$ . Esses pontos são novamente ordenados em ordem crescente de valor funcional, a fim de determinar a configuração do simplex na próxima iteração. A Figura 2.3a apresenta visualmente essa nova configuração do simplex, em que podemos observar a disposição dos pontos  $x_1$ ,  $x_2$  e  $x_e$ .

Após esse reordenarmos, identificamos  $x_3$  como o pior ponto. Procedemos então com uma nova etapa de reflexão, utilizando o centroide da face oposta como ponto de referência. A Figura 2.3b ilustra a reflexão do ponto  $x_3$ , resultando no ponto  $x_r$ .

Ao compararmos o valor funcional de  $x_1$  com o valor funcional de  $x_r$ , notamos que  $f(x_1) < f(x_r)$ . Portanto, aceitamos a iteração atual e prosseguimos para a próxima etapa.

Na próxima iteração, reorganizamos os pontos do simplex com base na nova configuração. A Figura 2.4a ilustra essa reorganização, apresentando a disposição dos pontos  $x_1$ ,  $x_2$  e  $x_r$ .

A nova reflexão do vértice  $x_3$  é evidenciada na Figura 2.4b. Nessa reflexão, podemos observar que o ponto  $x_r$  é melhor que  $x_3$ , porém, não é melhor que  $x_1$ . Com base nessa avaliação, mais uma vez aceitamos a iteração.

Na sequência, temos o simplex reconfigurado, conforme ilustrado na Figura 2.5a. Essa reorganização envolve a disposição dos pontos  $x_1$ ,  $x_2$  e  $x_r$ .

Avançando para mais uma iteração, procedemos com a reflexão do ponto  $x_3$ , conforme pode ser observado na Figura 2.5b.

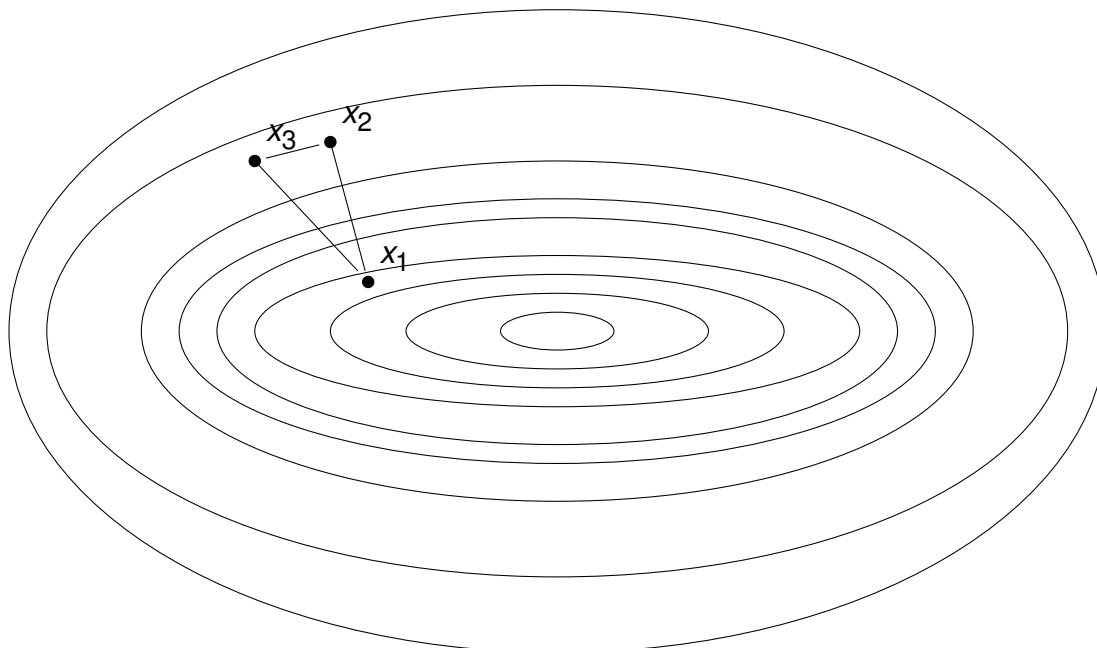
No entanto, ao avaliarmos o ponto resultante,  $x_r$ , constatamos que ele é pior que o ponto  $x_3$  refletido. Ou seja, a reflexão não proporcionou uma solução melhor. Diante dessa constatação, prosseguimos com uma contração, que pode ser realizada de forma interna ou externa, com o intuito de explorar alternativas e ajustar a direção da busca. Essa contração é representada visualmente na Figura 2.6a.

A contração interna envolve a redução da distância entre o ponto  $x_3$  e o centroide da face oposta, permitindo que o modelo explore uma região mais restrita do espaço de busca. Essa estratégia busca corrigir possíveis desvios e ajustar o direcionamento

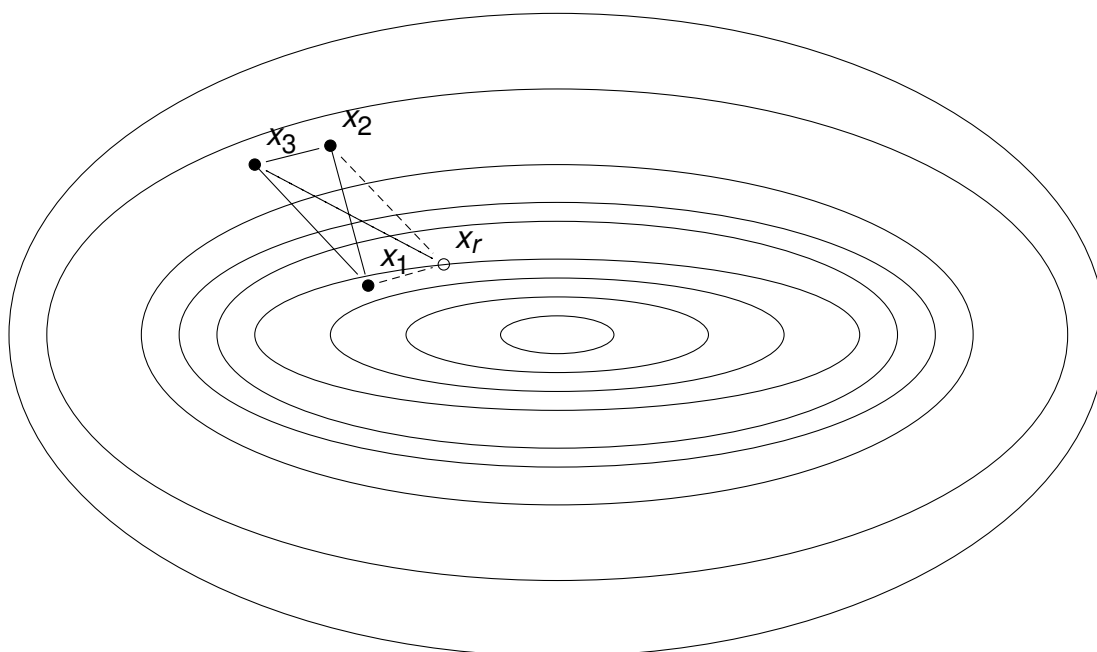


Figura 2.3 – Funcionamento do Método de Nelder Mead (2)

(a) Organização após a expansão



(b) Reflexão na nova iteração



Fonte: Elaborado pelo autor.

do processo iterativo.

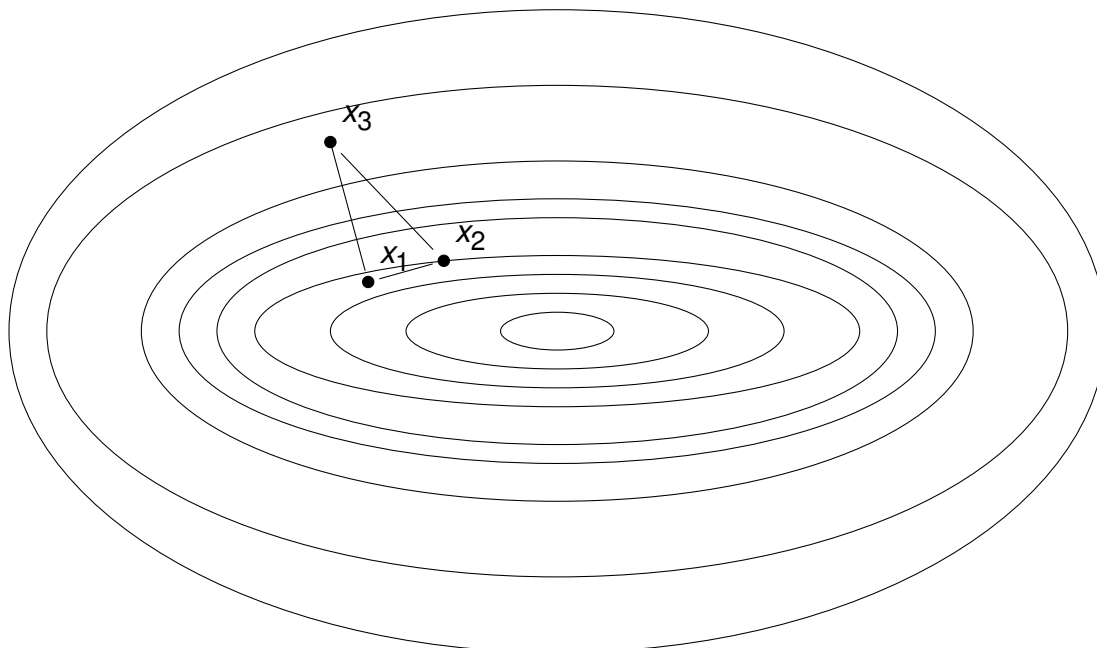
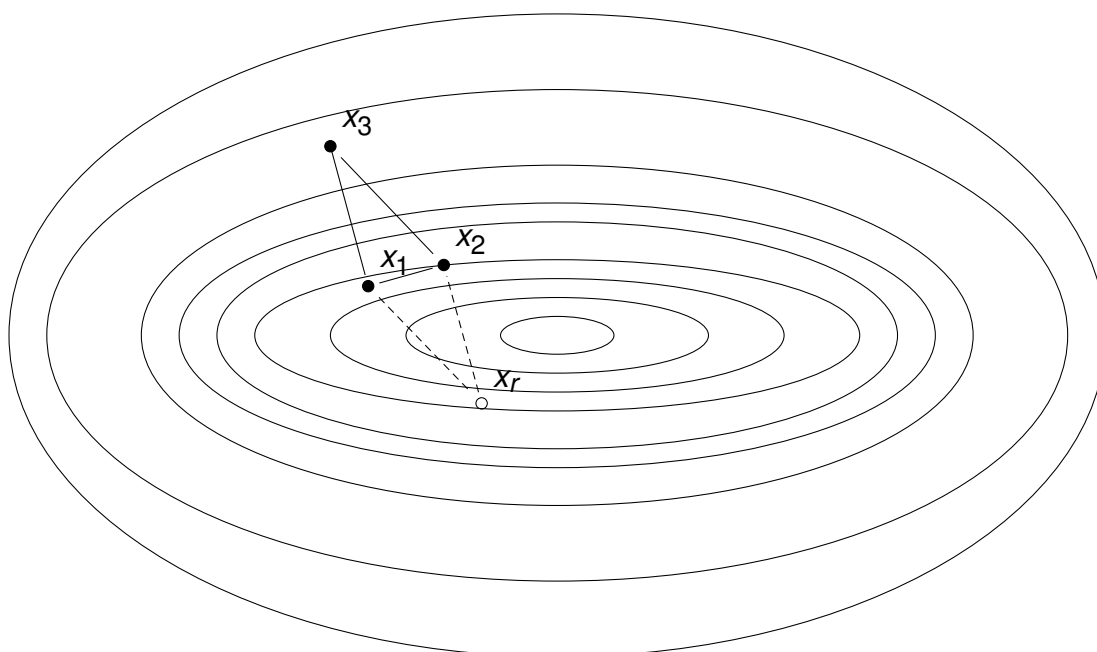
Após a realização das contrações, identificamos que o ponto  $x_{ci}$ , proveniente da contração interna, é o melhor ponto encontrado até o momento.

Como resultado, temos o simplex reconfigurado na Figura 2.6b, com os índices dos pontos organizados em ordem crescente de valor funcional.

Observando a Figura 2.6b, é possível visualizar a disposição dos pontos  $x_1$ ,

Figura 2.4 – Funcionamento do Método de Nelder Mead (3)

(a) Organização na nova iteração

(b) Reflexão do vértice  $x^3$ 

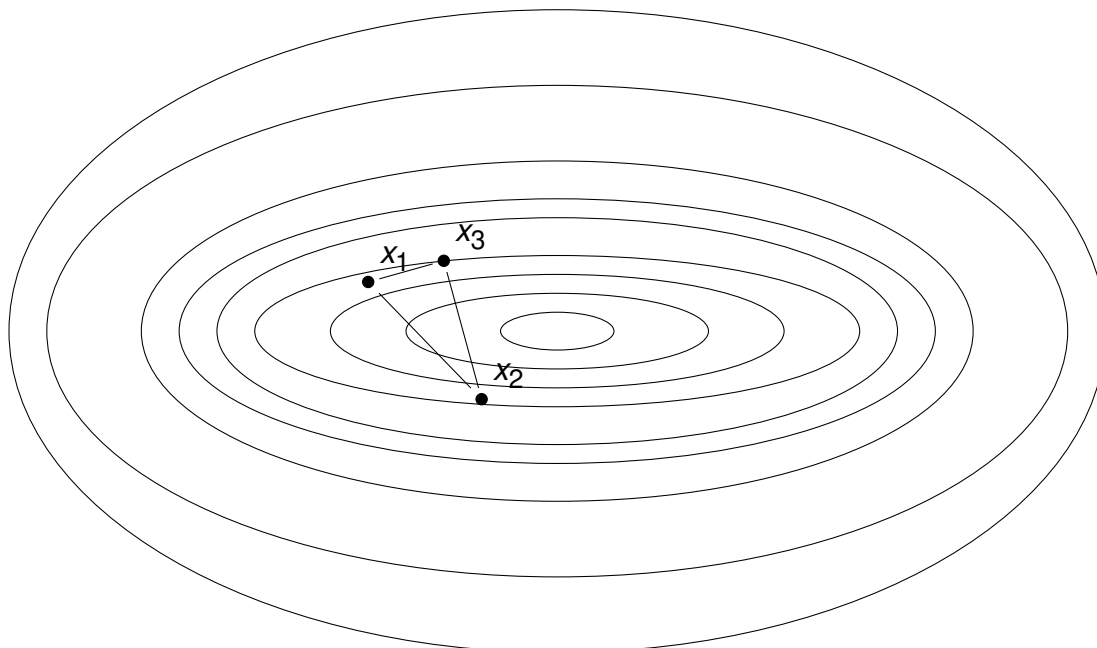
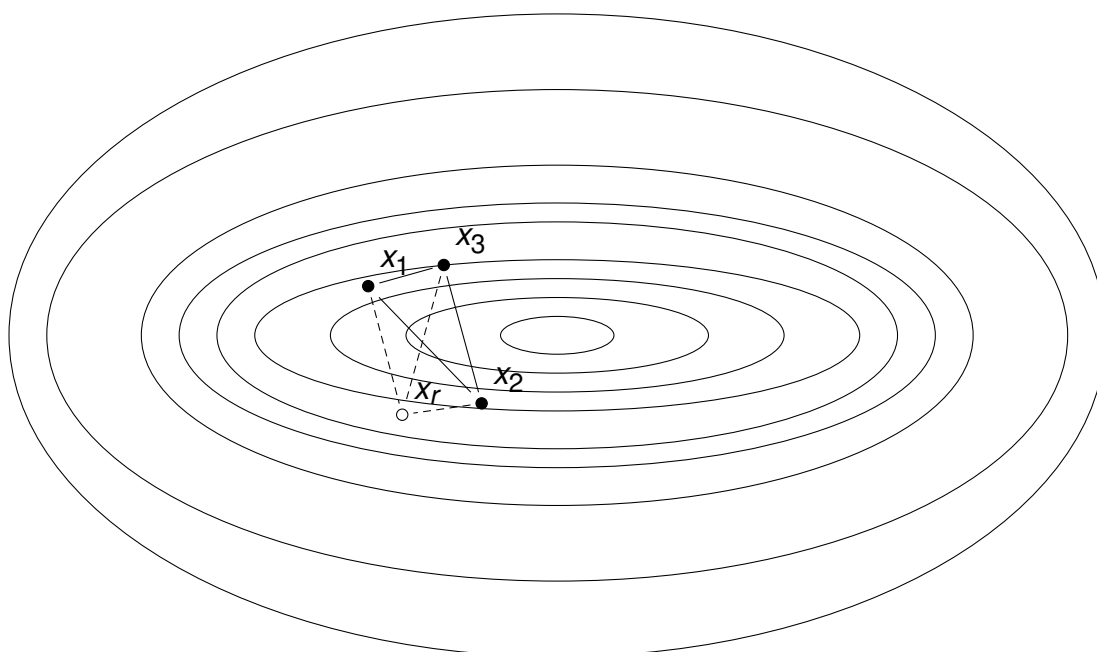
Fonte: Elaborado pelo autor.

$x_2$  e  $x_{ci}$ , refletindo o progresso alcançado até o momento. O simplex reconfigurado representa uma etapa promissora no processo iterativo, direcionando o modelo em direção a uma solução cada vez mais próxima do ótimo global.

Ao longo das iterações subsequentes, o modelo continuará a ajustar e a refinar os pontos do simplex, explorando diferentes estratégias e operações para se aproximar cada vez mais da solução desejada.

Figura 2.5 – Funcionamento do Método de Nelder Mead (4)

(a) Nova organização do simplex

(b) Novo espelhamento de  $x^3$ 

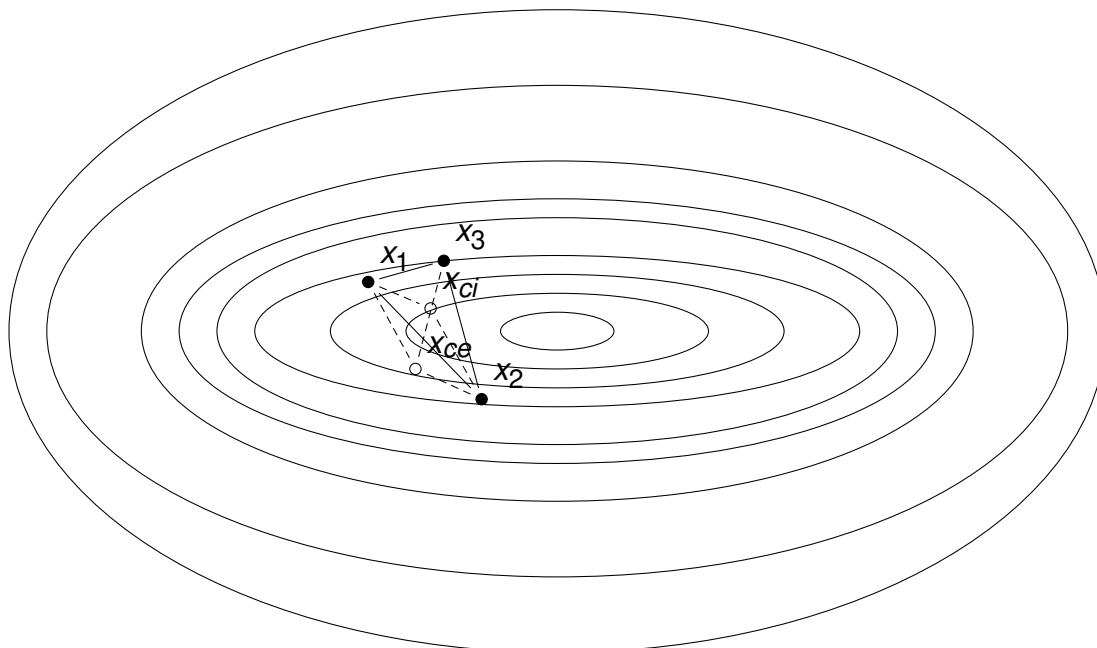
Fonte: Elaborado pelo autor.

No processo de ordenação dos vértices do simplex no início da iteração, é possível que ocorram empates entre os valores da função de dois ou mais pontos. Em relação a esse cenário, Diniz-Ehrhardt, Lopes e Pedroso (2012) destacam a existência de uma regra de desempate sugerida por Lagarias *et al.* (1998):

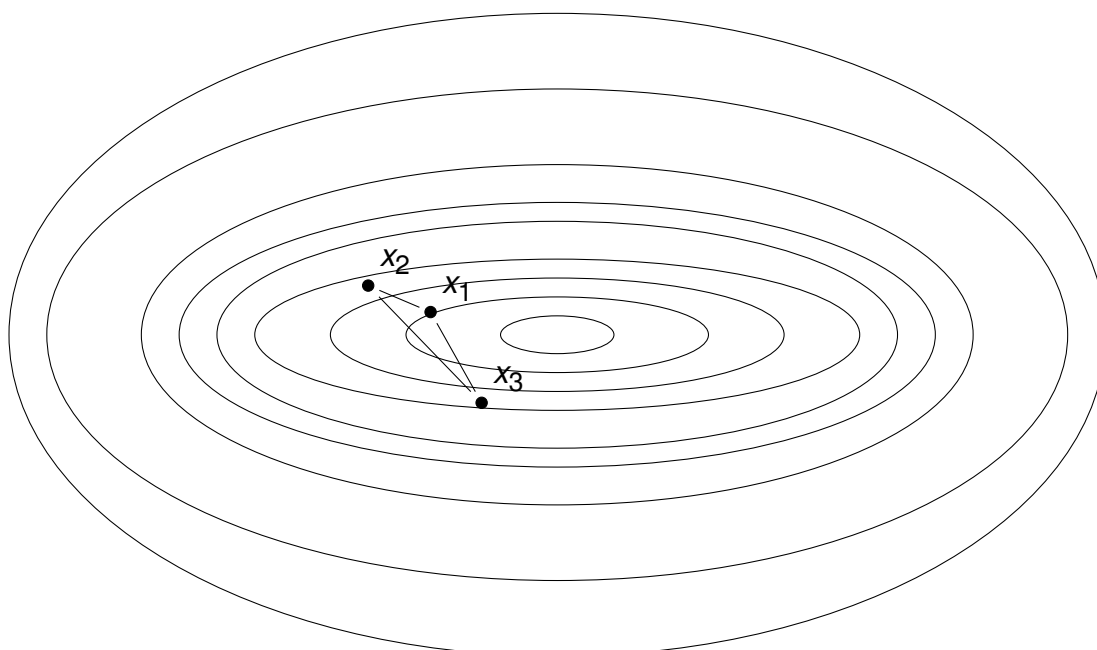
- **Iteração sem redução do simplex:** Quando não há redução do simplex, apenas o pior vértice  $x_{n+1}^{(k)}$  é descartado e substituído por um ponto que cha-

Figura 2.6 – Funcionamento do Método de Nelder Mead (5)

(a) Contração Interna e Externa



(b) Finalização da iteração



Fonte: Elaborado pelo autor.

maremos de  $v^{(k)}$ , criado na iteração  $k$ . Ele tomará a posição  $j+1$  do simplex da iteração seguinte  $(\Delta_{k+1})$ , onde  $j = \max_{0 \leq m \leq n} \{m : f(v^{(k)}) < f(x_{m+1}^{(k)})\}$ . Os outros vértices continuam na ordem que foi estabelecida antes de se iniciar a iteração.

- **Iteração com redução do simplex:** Nesse caso, apenas o vértice  $x^1$  será aproveitado para a próxima iteração. Apenas uma regra de desempate é

necessária: caso o melhor vértice  $x_1^{(k)}$  empate com um ou mais vetores como sendo o ponto de melhor valor da função do novo simplex. Caso isso ocorra, faremos  $x_1^{k+1} = x_1^k$ . Em qualquer outra situação, poderemos fazer uso de qualquer outra regra de desempate depois de uma redução.

Essa regra de desempate proposta por Lagarias *et al.* (1998) auxilia na determinação da ordem dos pontos em situações de empate, garantindo uma abordagem consistente e objetiva para a ordenação dos vértices.

Para compreender a hipótese do Teorema 2.1, que garante um caso de convergência, precisamos de uma definição importante:

**Definição 2.1** (Função estritamente convexa). *Dado um intervalo real  $I$ , uma função  $f : I \rightarrow \mathbb{R}$  é dita estritamente convexa se, para todos  $a, b \in I$  distintos e todo  $t \in (0,1)$ , tivermos:*

$$f((1-t)a + tb) < (1-t)f(a) + tf(b).$$

De uma perspectiva geométrica, a definição de uma função convexa implica que, ao traçar uma linha entre quaisquer dois pontos pertencentes ao gráfico da função, essa linha estará sempre posicionada acima do próprio gráfico. A propriedade de convexidade implica que a função não terá curvas ou variações bruscas em seu gráfico. Em termos práticos, isso significa que a função se comporta de maneira suave e previsível.

A convergência do método do Nelder-Mead para funções de  $\mathbb{R}$  em  $\mathbb{R}$  foi demonstrada por Lagarias *et al.* (1998). O resultado é enunciado no Teorema 2.1:

**Teorema 2.1** (Convergência para  $\mathbb{R}$ ). *Seja  $f : \mathbb{R} \rightarrow \mathbb{R}$  estritamente convexa, com curvas de nível limitadas. Se, o simplex inicial para o Algoritmo de Nelder-Mead com  $\rho\chi \geq 1$  é não degenerado, então os dois vértices convergem ao minimizador.*

Além disso, Diniz-Ehrhardt, Lopes e Pedroso (2012) destacam que, do ponto de vista teórico, não há garantia de convergência do método para dimensões superiores a 2.

No caso de  $f : \mathbb{R}^2 \rightarrow \mathbb{R}$ , Lagarias *et al.* (1998) provam que a convergência é possível se a função for estritamente convexa, com curvas de nível limitadas e o simplex inicial for não degenerado. Eles apresentam duas situações específicas:

1. Usando os parâmetros  $\rho = 1$  e  $\gamma = 0,5$ , os três vértices do simplex convergem para pontos com o mesmo valor da função.
2. Usando os parâmetros  $\rho = 1$ ,  $\chi = 2$  e  $\gamma = 0,5$ , a sequência dos diâmetros do simplex tende a zero.

No entanto, é importante ressaltar que essa convergência pode ocorrer para pontos que não sejam estacionários.

Apesar da falta de garantia de convergência, o método do Nelder-Mead é amplamente utilizado na otimização sem derivadas devido à sua eficácia e simplicidade.

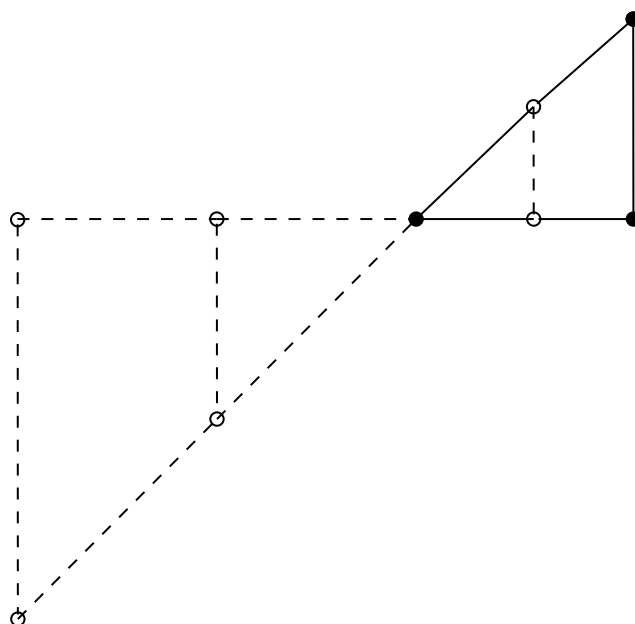
## 2.2 BUSCA MULTIDIRECIONAL

O Método da Busca Multidirecional é uma variação aprimorada do método de Nelder-Mead, no qual não apenas o pior vértice do simplex é modificado, mas sim todos os pontos, exceto o melhor. Isso permite uma rotação do simplex no espaço de busca.

Nele, após realizar a operação de reflexão com base no melhor ponto, se o melhor, dentre os novos vértices obtidos, for melhor que o melhor vértice antigo, é experimentada uma expansão. Caso contrário, é feita uma redução.

A Figura 2.7 ilustra as possibilidades de redução, rotação e expansão do simplex em torno do melhor vértice.

Figura 2.7 – Possibilidades da Busca Multidirecional



Fonte: Adaptado de Diniz-Ehrhardt, Lopes e Pedroso (2012)

Esse processo de modificação e avaliação dos vértices é repetido até que o diâmetro do simplex, representado por  $\Delta$ , atinja uma tolerância desejada.

Os parâmetros  $\chi$  e  $\sigma$  são utilizados para controlar a expansão e a redução do simplex, sendo necessário que  $0 < \sigma < 1 < \chi$ . Esses parâmetros influenciam na magnitude das operações de expansão e redução (DINIZ-EHRHARDT; LOPES; PEDROSO, 2012).

O Algoritmo 2.2, adaptado de Diniz-Ehrhardt, Lopes e Pedroso (2012) e Conn, Scheinberg e Vicente (2009), mostra o funcionamento do método.

**Algoritmo 2.2:** Algoritmo da Busca Multidirecional

**Dados:** Escolher  $n$  pontos  $x_k$ , para  $k = \{1, 2, \dots, n + 1\}$ . Definir os parâmetros  $\chi$  e  $\sigma$ . ( $\chi = 2$  e  $\sigma = 0,5$ )

**Resultado:** Minimizar a função  $f$

1 **enquanto**  $\Delta > \Delta_{tol}$  **faça**

2     Ordenar os vértices do simplex de modo que  $f(x_1) \leq f(x_2) \dots \leq f(x_{n+1})$ ;

3     Rotacionar o simplex ao redor do melhor vértice, fazendo:

$$x_i^f = x_1 - (x_i - x_1), i = 2, 3, \dots, n + 1.$$

   ;

4     Definir  $f_r = \min_{i=2, \dots, n+1} \{f(x_i^f)\}$ ;

5     **se**  $f_r < f(x_1)$  **então**

6         Expandir o simplex rotacionado, fazendo

$$x_i^e = x_1 - \chi(x_i - x_1), i = 2, 3, \dots, n + 1$$

   ;

7         Defina  $f_e = \min_{i=2, \dots, n+1} \{f(x_i^e)\}$ . Se  $f_e < f_r$ , aceite a expansão. Caso contrário, aceite a rotação. Termine a iteração;

8     **senão**

9         Calcule os pontos  $v_i = x_1 + \sigma(x_i - x_1)$ ,  $i = 2, \dots, n + 1$ . Os vértices (ainda fora de ordem), para a próxima iteração são  $x_1, v_2, \dots, v_{n+1}$

10    **fim**

11 **fim**

A demonstração de convergência do Método da Busca Multidirecional é apresentada por Conn, Scheinberg e Vicente (2009). Os autores abordam o método como um dos Métodos de Busca Direta Direcional, que são explorados em detalhes no capítulo seguinte do texto. Eles utilizam a ideia de reticulado inteiro para analisar a convergência do método e chegam a resultados conclusivos sobre sua convergência.

### 3 MÉTODOS DE BUSCA DIRETA DIRECIONAIS

Dentre os modelos de busca direta, estudaremos a Busca Coordenada, Busca Direta, Busca pela Seção Áurea e Interpolação Parabólica Sucessiva. Cada um desses métodos será abordado em termos teóricos, explorando seu funcionamento, descrevendo algoritmos e, quando necessário, demonstrando sua convergência. Suas implementações e exemplos de aplicação serão apresentados em um capítulo posterior.

O estudo dos métodos de busca direcional desempenha um papel fundamental na teoria da otimização sem derivadas. De acordo com Diniz-Ehrhardt, Lopes e Pedroso (2012, p. 39), esses métodos “[...] sintetizam muito da filosofia da otimização sem derivadas, por seus algoritmos simples, de forte apelo geométrico, com robusta teoria de convergência”.

Estes métodos consistem em “testar a função objetivo em um número finito de pontos e, a cada iteração, decidir quais ações serão tomadas na próxima, apenas com base nesses valores da função, sem qualquer aproximação por derivadas ou construção de modelos.” (CONN; SCHEINBERG; VICENTE, 2009)

Para assegurar a convergência, é imperativo que trabalhem com funções contínuas dentro de um conjunto compacto de curvas de nível. Além disso, para que essa convergência seja para um ponto estacionário, é necessário que as funções sejam continuamente diferenciáveis. Estas condições essenciais serão apresentadas e discutidas ao longo do texto. (DENNIS JR; TORCZON, 1991)

Para iniciar a teoria dos métodos de busca direcionais, precisamos definir o que é um conjunto gerador positivo. (CONN; SCHEINBERG; VICENTE, 2009)

**Definição 3.1.** *Um gerador positivo de  $\mathbb{R}^n$  é o conjunto de vetores que gera  $\mathbb{R}^n$  por meio de combinações lineares com coeficientes positivos.*

Isto é, dado um conjunto de vetores  $V = \{v_1, \dots, v_n\}$  tais que

$$\{v \in \mathbb{R}^n : v = \alpha_1 v_1 + \dots + \alpha_n v_n, \alpha_i \geq 0, i = 1, \dots, n\},$$

dizemos que  $V$  é um gerador positivo de  $\mathbb{R}^n$ .

Giuliani (2013) elenca alguns teoremas importantes para a compreensão dos métodos de busca direcionais:

**Teorema 3.1.** *O conjunto  $D = \{d_1, d_2, \dots, d_r\}$  gera positivamente  $\mathbb{R}^n$  por meio de combinações lineares de coeficientes não negativos se, e somente se, para todo vetor não nulo  $v \in \mathbb{R}^n$ , existe  $d \in D$  tal que*

$$v^T d > 0.$$

*Demonstração.* Por um lado, um  $v \in \mathbb{R}^n$  qualquer pode ser expresso como  $v = \lambda_1 d_1 + \lambda_2 d_2 + \dots + \lambda_r d_r$ , com  $\lambda_i \geq 0$ , para  $i = \{1, 2, \dots, r\}$ . Agora,  $0 < v \cdot v = \lambda_1 d_1 \cdot v + \lambda_2 d_2 \cdot v + \dots + \lambda_r d_r \cdot v$  e ao menos um dos termos deve ser positivo.



Reciprocamente, assumamos que o subespaço  $D$ , gerado por  $\{d_1, d_2, \dots, d_r\}$ , não seja  $\mathbb{R}^n$ . Então, existe um hiperplano tal que  $D$  está completamente contido em um de seus lados, isto é, existe  $v \neq 0$  tal que  $d \in D$  implica  $v^T \cdot d \leq 0$ . ■

As combinações não negativas também podem ser chamadas de combinações cônicas e, uma vez que este conjunto gera  $\mathbb{R}^n$  dessa forma, eles podem ser denominados geradores positivos de  $\mathbb{R}^n$ .

**Teorema 3.2.** *Suponha que  $D = \{d_1, d_2, \dots, d_r\}$  gera  $\mathbb{R}^n$  por meio de combinações não negativas. Então o número de vetores é  $r \geq n + 1$ , e é possível  $r = n + 1$*

*Demonstração.* Se  $r \leq n$ , então ou  $\{d_1, d_2, \dots, d_r\}$  falha em gerar  $\mathbb{R}^n$  ou é exatamente uma base de  $\mathbb{R}^n$ . Em ambos os casos, falha em gerar  $\mathbb{R}^n$  positivamente. Por outro lado, para qualquer  $n$ , pode-se escolher  $\{d_1, d_2, \dots, d_n\}$  como uma base e  $d_{n+1} = -\lambda_1 d_1 - \dots - \lambda_n d_n$ ,  $\lambda_i > 0$ , fazendo  $\{d_1, d_2, \dots, d_{n+1}\}$  uma base positiva. ■

O caso  $r = n + 1$  é chamado de *base minimal*.

Ambos os teoremas são resultados demonstrados por Davis (1954). Eles desempenham um papel fundamental nos métodos de busca direta direcionais, sendo utilizados tanto para compreender esses métodos quanto para demonstrar sua convergência.

### 3.1 BUSCA COORDENADA

O primeiro método de busca direta direcional a ser estudado é a Busca Coordenada. Na literatura, este pode ser encontrado, ainda, sob os nomes de Busca Padrão ou Método das Variações Locais. (DINIZ-EHRHARDT; LOPES; PEDROSO, 2012)

Neste método, utilizaremos uma base positiva máxima  $d \in D$ , formada pelas direções canônicas positivas e negativas. Ou seja,  $D$  é formado pelas colunas das matrizes identidades de ordem  $n$ :  $I$  e  $-I$ .

Seja  $x_k \in \mathbb{R}^n$ ,  $k \in \mathbb{N}$ , um ponto da iteração atual,  $\alpha_k \in \mathbb{R}_+$  um valor definido para o tamanho do passo, e  $d \in D$ . O algoritmo da Busca Coordenada avalia a função  $f$  nos pontos do conjunto

$$P_k = \{x_k + \alpha_k d : d \in D\},$$

com o objetivo de encontrar um ponto  $P_k$  que faça decrescer o valor da função objetivo, ou seja, encontrar  $x_{k+1}$  de modo que

$$x_{k+1} = x_k + \alpha_k d \quad \text{tal que} \quad f(x_{k+1}) < f(x_k)$$

Se todos os pontos  $x_{k+1}$  resultarem em um valor  $f(x_{k+1}) > f(x_k)$ , o que indica uma falta de melhora na função objetivo, continuamos no mesmo ponto, ou seja, atribuímos  $x_{k+1} = x_k$ . No entanto, diminuímos pela metade o tamanho do passo  $\alpha_k$ . Isso

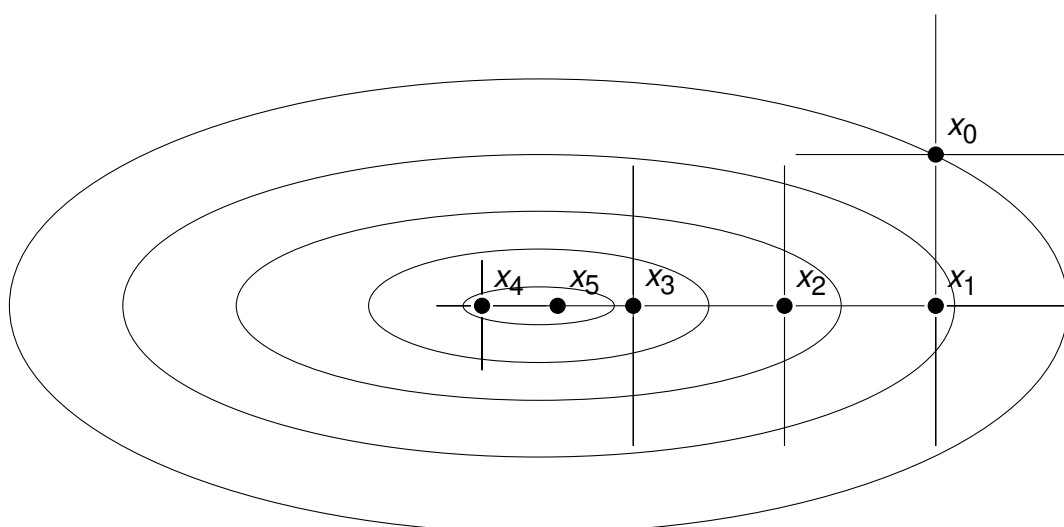
significa que reduzimos o valor de  $\alpha_k$  para  $\alpha_k/2$ . Essa redução no tamanho do passo é realizada na tentativa de explorar outras direções de busca e encontrar uma região mais promissora para a otimização.

Seguindo com esse processo, define-se um valor mínimo para  $\alpha_k$  ao qual aceita-se como resolvido o problema (CONN; SCHEINBERG; VICENTE, 2009).

No Exemplo 3.1, ilustrado pela Figura 3.1, temos uma descrição visual do algoritmo de Busca Direta.

**Exemplo 3.1.** *Iniciando pelo ponto  $x_0$  tomaram-se as direções canônicas positivas e negativas, resultando em quatro pontos. Deles, o melhor ponto foi denotado como  $x_1$ . De mesma forma, a partir de  $x_1$ , achou-se o melhor ponto  $x_2$ , seguido por  $x_3$  e  $x_4$ . Ao se fazer a iteração em  $x_4$ , nenhum dos pontos encontrados é melhor que ele. Assim, deve-se diminuir o parâmetro  $\alpha$  da razão de um meio. E assim, nesse novo parâmetro, encontrou-se o ponto  $x_5$ . O processo continuaria até atingir a tolerância escolhida.*

Figura 3.1 – Método da Busca Direta



Fonte: Elaborado pelo autor

O Algoritmo 3.1 descreve o método da busca coordenada.

A geometria do gráfico da função e a escolha das bases canônicas de  $\mathbb{R}^n$  garantem que todas as direções de busca sejam ortogonais entre si. Isso significa que o conjunto de direções abrange todo o espaço, permitindo explorar diferentes regiões em busca de uma direção de decrescimento da função objetivo.

Devido à cobertura completa do espaço pelas direções ortogonais, é sempre possível encontrar uma direção de decrescimento, desde que seja escolhido um tamanho de passo apropriado  $\alpha_k$ . Essa propriedade intuitiva sugere que, ao utilizar o método de Busca Direta com bases canônicas, temos a capacidade de explorar integralmente o espaço de busca em busca de melhores soluções.

**Algoritmo 3.1:** Algoritmo da Busca Coordenada

**Dados:** Escolha  $x_0$  e  $\alpha_0 > 0$ . Determine um  $\alpha_{tol} > 0$   
**Resultado:** Minimizar a função  $f$

- 1 **enquanto**  $\alpha_k > \alpha_{tol}$  **faça**
- 2     Determine os elementos de  $P_k = \{x_k + \alpha_k d : d \in D\}$  e os organize em alguma ordem determinada;
- 3     Avalie  $f$  em cada um dos pontos;
- 4     **se**  $x_k + \alpha_k$  *for tal que*  $f(x_k + \alpha_k) < f(x_k)$  **então**
- 5         pare o processo;
- 6         Defina  $x_{k+1} = x_k + \alpha_k$ ;
- 7         Declare a iteração um sucesso;
- 8     **senão**
- 9         Declare a iteração malsucedida;
- 10         Defina  $x_{k+1} = x_k$ ;
- 11     **fim**
- 12 **fim**

## 3.2 BUSCA DIRETA DIRECIONAL

Os algoritmos da Busca Coordenada fazem parte de uma classe mais ampla de métodos conhecidos como Busca Direta Direcionais.

Inicialmente, consideramos um ponto  $x_k$  e um parâmetro  $\alpha_k$  que determina o tamanho do passo. O objetivo da iteração  $k$  é encontrar um novo ponto  $x_{k+1}$  de tal forma que  $f(x_{k+1}) < f(x_k)$ .

A cada iteração  $x_{k+1}$ , o processo de cálculo pode ser dividido em duas etapas: a etapa de busca e a etapa de pesquisa.

A etapa de busca, que é opcional e não necessária para a convergência do método, envolve a avaliação da função em um número finito de pontos. A escolha desses pontos é arbitrária e depende da preferência do usuário. Essa etapa pode ajudar a identificar um ponto  $x_k$  com um valor de função  $f(x_k)$  menor do que todos os outros pontos avaliados. No entanto, se a etapa de busca não for bem-sucedida, passamos para a próxima etapa.

A etapa de pesquisa consiste em uma busca local na vizinhança do ponto atual de iteração. Essa busca envolve a exploração de um conjunto de pontos definidos pelo parâmetro de passo  $\alpha_k$  e pela base  $d \in D_k$ , onde  $D_k$  é uma base positiva em  $\mathbb{R}^n$ . Esses pontos são dados pela expressão

$$P_k = \{x_k + \alpha_k d : d \in D_k\}.$$

Nessa etapa de pesquisa, procuramos encontrar um novo ponto  $x_{k+1}$  que resulte em uma redução na função objetivo  $f$ .

É importante destacar que a etapa de busca é opcional e serve como uma ferramenta adicional para encontrar um ponto com um valor de função objetivo menor.

No entanto, mesmo se a etapa de busca não for bem-sucedida, o método de Busca Coordenada prossegue para a etapa de pesquisa, onde ocorre uma busca ativa por pontos que levem a uma redução da função objetivo.

Na etapa de pesquisa da Busca Direta Direcional, o objetivo é reduzir a função objetivo, alcançando um valor suficientemente pequeno para o parâmetro  $\alpha_k$ . Cada iteração é considerada bem-sucedida se um novo ponto  $x_{k+1} \in P_k$  for encontrado, tal que  $f(x_{k+1}) < f(x_k)$ . Se nenhum ponto  $x_k \in P_k$  satisfizer essa condição, a iteração é considerada mal-sucedida e o valor do parâmetro  $\alpha_k$  é reduzido por uma determinada razão. Esse parâmetro pode permanecer inalterado ou ser aumentado caso a iteração seja bem-sucedida (CONN; SCHEINBERG; VICENTE, 2009).

O Algoritmo 3.2 descreve o funcionamento geral do Método de Busca Direta Direcional.

### Algoritmo 3.2: Algoritmo da Busca Direta Direcional

**Dados:** Escolha  $x_0$  e  $\alpha_0 > 0$ ,  $0 < \beta_1 \leq \beta_2 < 1$  e  $\gamma \geq 1$ .

**Resultado:** Minimizar a função  $f$

```

1  enquanto  $\alpha_k > \alpha_{tol}$  faça
2      Procure um ponto  $x$  tal que  $f(x) < f(x_k)$  em um número finito de pontos ;
3      Se for encontrado, defina  $x_{k+1} = x$ , declare a iteração e a etapa de
        busca bem sucedidas e passe a etapa de pesquisa;
4      Escolha uma base positiva  $D_k$  do conjunto  $\mathcal{D}$  de bases positivas;
5      Ordene o conjunto de pontos  $P_k = \{x_k + \alpha_k d : d \in D_k\}$ ;
6      Avalie  $f$  nos pontos de  $P_k$  seguindo a ordem estabelecida;
7      se  $x_k + \alpha_k d_k$  for encontrado de modo que  $f(x_k + \alpha_k d_k) < f(x_k)$  então
8          pare o processo;
9          Defina  $x_{k+1} = x_k + \alpha_k d_k$ ;
10         Declare a iteração um sucesso;
11         Mantenha ou aumente o parâmetro:  $\alpha_{k+1} \in [a_k, \gamma a_k]$ 
12     senão
13         Declare a iteração malsucedida;
14         Defina  $x_{k+1} = x_k$ ;
15         Diminua o tamanho do parâmetro:  $a_{k+1} \in [\beta_1 \alpha_k, \beta_2 \alpha_k]$ ;
16     fim
17 fim

```

O critério natural para finalização da busca direta direcional acontece quando  $\alpha_k < \alpha_{tol}$ , para um  $\alpha_{tol}$  escolhido. Em geral, pode-se fazer  $\alpha_{tol} = 10^{-5}$ . (CONN; SCHEINBERG; VICENTE, 2009)

Note que a Busca Direta Direcional é mais flexível em comparação com a Busca Coordenada. Enquanto a Busca Coordenada utiliza um tamanho de passo fixo, que é reduzido pela metade em caso de insucesso, a Busca Direta Direcional permite a escolha de um parâmetro  $\alpha$  dentro de um intervalo a cada iteração, e esse intervalo é atualizado com base nos resultados obtidos. Isso proporciona uma maior adaptabili-

dade do método às características da função objetivo.

Portanto, a Busca Coordenada pode ser vista como um caso particular da Busca Direta Direcional, onde o tamanho de passo é determinado de maneira fixa e não há atualização do intervalo  $\alpha$  a cada iteração. A flexibilidade adicional da Busca Direta Direcional permite explorar mais efetivamente o espaço de busca, possibilitando uma convergência mais rápida e eficiente.

### 3.3 CONVERGÊNCIA DOS MÉTODOS DE BUSCA DIRETA DIRECIONAIS

Para o estudo da Convergência dos Métodos de Busca Direta Direcionais, Conn, Scheinberg e Vicente (2009) listam algumas hipóteses a serem observadas para a demonstração da convergência. Uma característica geral imperativa é que a função  $f$  seja estritamente convexa.

Dentre as hipóteses particulares, a primeira delas é que as iterações do método fazem parte de um conjunto compacto. Dada que a sequência de iterações  $x_k$  é de modo que  $f(x_k)$  é monótona decrescente, uma forma conveniente de impor essa condição é assumir que o conjunto de níveis  $N(x_0) = \{x \in \mathbb{R}^n : f(x) \leq f(x_0)\}$  é compacto.

**Hipótese 3.1.** *O Conjunto de Níveis  $N(x_0) = \{x \in \mathbb{R}^n : f(x) \leq f(x_0)\}$  é compacto.*

Para analisar a convergência global desses métodos, vamos analisar sua convergência para pontos estacionários a partir de pontos arbitrários.

#### 3.3.1 Comportamento do parâmetro de passos

A primeira análise da convergência se dará sobre a prova de que existe uma subsequência, formada pelos parâmetros dos passos, que converge para zero. Para tal, necessitamos da segunda hipótese:

**Hipótese 3.2.** *Se existir um  $\alpha > 0$  tal que  $\alpha_k > \alpha$ , para todo  $k$ , então o algoritmo contempla um número finito de pontos.*

Com essa hipótese razoável, podemos provar que o tamanho de  $\alpha_k$  tende para zero.

**Teorema 3.3.** *Sejam as Hipóteses 3.1 e 3.2 satisfeitas. Então, a sequência dos parâmetros dos passos satisfaz:*

$$\liminf_{k \rightarrow +\infty} \alpha_k = 0$$

*Demonstração.* Vamos supor, por absurdo, que exista  $\alpha > 0$  tal que  $\alpha_k > \alpha$ , para todo  $k$ . Então, pela hipótese 3.2, o algoritmo contempla um número finito de pontos.

Porém, por outro lado, o algoritmo obtém um valor diferente de  $x_k$  somente quando ocorre um decrescimento da função. Assim, como a função contempla um número finito de pontos, deve existir uma iteração  $\bar{k}$  tal que  $x_k = x_{\bar{k}}$ , para todo  $k \geq \bar{k}$ .

Toda iteração  $m$ ,  $m > \bar{k}$  será malsucedida, e como  $\alpha_{k+1} = \alpha_k/2$  a cada uma delas, segue que  $\lim_{k \rightarrow +\infty} \alpha_k = 0$ . Contradição. ■

Segue o seguinte corolário.

**Corolário 3.4.** *Sejam as Hipóteses 3.1 e 3.2 satisfeitas. Então, existe um ponto  $x_*$  e uma subsequência  $k_i$  de iterações malsucedidas para quais:*

$$\lim_{i \rightarrow +\infty} \alpha_{k_i} = 0 \text{ e } \lim_{i \rightarrow +\infty} x_{k_i} = x_*$$

*Demonstração.* Do Teorema 3.3, concluímos que existe uma subsequência infinita de iterações que levam o parâmetro de tamanho dos passos  $\alpha_k$  para zero.

Como consequência, deve existir uma subsequência infinita de iterações correspondendo às etapas malsucedidas, uma vez que o parâmetro de tamanho de passo é reduzido apenas nessas iterações.

Defina  $K_U^1$  denotando a subsequência de índices de todas as etapas de procedimento sem sucesso.

Segue-se também a partir do procedimento que atualiza o parâmetro de tamanho de passo  $\alpha_k$  e das observações acima que deve existir uma subsequência  $K_U^2 \subset K_U^1$  tal que  $\alpha_{k+1} \rightarrow 0$  para  $k \in K_U^2$ .

Uma vez que  $a_k \leq (1/\beta_1)\alpha_{k+1}$  para  $k \in K_U^2$ , obtemos  $a_k \rightarrow 0$  para  $k \in K_U^2$ .

Como  $\{x_k\}_{k \in K_U^2}$  é monótona e limitada, ela contém uma subsequência convergente  $\{x_k\}_{k \in K_U^3}$ . Definimos  $x_* = \lim_{k \rightarrow \infty} x_k$ ,  $k \in K_U^3$ .

Desde que  $K_U^3 \subset K_U^2$ , segue que  $\lim_{k \rightarrow \infty} \alpha_k = 0$ ,  $k \in K_U^3$ . Por fim, definimos  $\{k_i\} = K_U^3$  e o resultado segue. ■

Esta sequência de estudos é a seguida por Conn, Scheinberg e Vicente (2009).

### 3.3.2 Conjunto Arbitrário de Bases Positivas

Na abordagem dos métodos de busca direta direcional, é possível utilizar um número infinito de bases positivas, desde que essas bases não se tornem degeneradas. A degeneração ocorre quando as medidas de cosseno das bases positivas se aproximam de zero, o que pode levar a problemas numéricos e dificultar a convergência adequada do método.

Para evitar a degeneração, é necessário impor uma restrição nas medidas de cosseno das bases positivas, garantindo que elas estejam uniformemente limitadas longe de zero. Isso significa que as medidas de cosseno não devem se aproximar muito de zero, mantendo uma distância mínima.

Além disso, é importante impor limites no tamanho de todos os vetores presentes nas bases positivas utilizadas. Esses limites asseguram que os vetores não cresçam indefinidamente, o que poderia levar a problemas de convergência e dificultar a busca por um mínimo global.

Giuliani (2013) define a medida do cosseno de um conjunto  $\mathcal{D}$ :

**Definição 3.2.** Definimos a medida do cosseno de um conjunto  $\mathcal{D} \in \mathbb{R}^n$  como

$$k(\mathcal{D}) = \min_{v \in \mathbb{R}^n} \max_{d \in \mathcal{D}} \frac{v^T d}{\|v\| \|d\|},$$

para  $v \neq 0$ .

Assim, “ $k(\mathcal{D})$  é cosseno do maior ângulo interno formado entre um vetor não nulo  $v \in \mathbb{R}^n$  e o vetor  $d \in \mathcal{D}$  mais próximo de  $v$ ”. (GIULIANI, 2013, p. 27)

A Hipótese 3.3 adotada por Conn, Scheinberg e Vicente (2009) irá manter uma boa geometria das direções de busca, garantindo um valor mínimo para  $k(\mathcal{D})$  e irá adotar um tamanho máximo para os vetores  $d \in \mathcal{D}$ .

**Hipótese 3.3.** Sejam  $\xi_1, \xi_2 > 0$  constantes reais positivas e fixas. As bases positivas  $D_k$  usadas no algoritmo são escolhidas a partir do conjunto

$$\mathcal{D} = \{\bar{D} \text{ bases positivas} : k(\bar{D}) > \xi_1, \|\bar{d}\| \leq \xi_2, \bar{d} \in \bar{D}\}.$$

A seguir, para trabalharmos com um número infinito de bases, exigimos que o gradiente de  $f$  seja *Lipschitz Contínuo* no conjunto de níveis  $N(x_0)$ :

**Definição 3.3.** Seja  $I$  um intervalo real. Uma função  $f : I \rightarrow \mathbb{R}$  é dita *Lipschitz Contínua* se existir uma constante  $v > 0$  tal que:

$$|f(x) - f(y)| \leq v|x - y|.$$

Essa definição nos diz que a função  $f$  é Lipschitz Contínua se, para quaisquer dois pontos  $x$  e  $y$  no conjunto  $D$ , a diferença entre os valores de  $f(x)$  e  $f(y)$  é limitada por uma constante multiplicada pela distância entre  $x$  e  $y$ . Essa constante  $v$  é chamada de constante de Lipschitz.

No contexto dos métodos de busca direta direcional, a exigência de que o gradiente de  $f$  seja Lipschitz Contínuo em um conjunto de níveis  $N(x_0)$  implica que a função objetivo possui uma variação suave em seu valor à medida que nos movemos no espaço de busca. Essa condição é importante para garantir a convergência dos métodos e evitar comportamentos indesejados, como oscilações ou divergências.

Além disso, se uma função é lipschitziana, então ela é uniformemente contínua. (LIMA, 1993)

Por fim, a assumiremos uma última hipótese:

**Hipótese 3.4.** O gradiente  $\nabla f$  é Lipschitz Contínuo em um conjunto aberto contendo  $N(x_0)$ , com constante de Lipschitz  $v > 0$ .

Por fim, para demonstrar o Teorema 3.6, o qual estabelece a convergência global para a classe de métodos de busca direta considerados, é necessário utilizar o seguinte resultado, previamente demonstrado por Conn, Scheinberg e Vicente (2009):

**Lema 3.5.** Seja  $D$  um conjunto gerador positivo e  $\alpha > 0$  dado. Assuma que  $\nabla f$  é Lipschitz Contínuo (com constante de Lipschitz  $v > 0$ ) em um conjunto aberto contendo a bola  $B(x, \Delta)$ . Se  $f(x) \leq f(x + \alpha d)$ , para todo  $d \in D$ , então

$$\|\nabla f(x)\| \leq \frac{v}{2} k(D)^{-1} \max_{d \in D} \|d\| \alpha,$$

sendo  $\Delta = \alpha \max_{d \in D} \|d\|$ .

*Demonstração.* Seja  $d \in D$  para o qual

$$k(D) \|\nabla f(x)\| \|d\| \leq -\nabla f(x)^T d. \quad (3.2)$$

Agora, a partir do Teorema do Valor Médio para integrais, e usando o fato que  $f(x) \leq f(x + \alpha d)$ , obtemos, para todo  $d \in D$ , que

$$0 \leq f(x + \alpha d) - f(x) = \int_0^1 \nabla f(x + t\alpha d)^T (\alpha d) dt, \quad (3.3)$$

em que, multiplicando a equação (3.2) por  $\alpha$  e a somando com a equação (3.3), temos:

$$k(D) \|\nabla f(x)\| \|d\| \alpha \leq \int_0^1 (\nabla f(x + t\alpha d) - \nabla f(x))^T (\alpha d) dt \leq \frac{v}{2} \|d\|^2 \alpha^2$$

e a prova está completa. ■

Segue o teorema que demonstra a convergência neste caso:

**Teorema 3.6.** Sejam as hipóteses 3.1, 3.2, 3.3 e 3.4 cumpridas. Então:

$$\lim_{k \rightarrow +\infty} \|\nabla f(x_k)\| = 0$$

E a sequência de iterações  $\{x_k\}$  tem limite no ponto  $x_*$  (dado pelo Corolário 3.4), para qual

$$\nabla f(x_*) = 0$$

*Demonstração.* O Corolário 3.4 mostrou que existe uma subsequência  $\{k_j\}$  de iterações sem sucesso para os quais a hipótese 3.4 é verdadeira.

Do Lema 3.5, temos que:

$$\|\nabla f(x_{k_j})\| \leq \frac{v}{2} k(D_{k_j})^{-1} \max_{d \in D_{k_j}} \|d\| \alpha_{k_j}.$$



Utilizando Hipótese 3.3, e fazendo as devidas substituições, temos

$$\|\nabla f(x_{k_i})\| \leq \frac{v\xi_2}{2\xi_1} \alpha_{k_i}$$

Como  $\alpha_{k_i} \rightarrow 0$ , concluímos que

$$\lim_{i \rightarrow +\infty} \|\nabla f(x_{k_i})\| = 0.$$

Uma vez que  $\nabla f$  é contínua,  $x_* = \lim_{i \rightarrow +\infty} x_{k_i}$  é tal que  $\nabla f(x_*) = 0$ . ■

Além disso, como a função é, por hipótese, estritamente convexa,  $x_*$  é um minimizador da função.

Por fim, estudaremos o caso de um conjunto com finitas bases positivas.

### 3.3.3 Conjunto Finito de Bases Positivas

A prova do Teorema 3.6 é válida quando o conjunto de todas as bases positivas é infinito, desde que  $k(D_k)^{-1}$  seja uniformemente limitado, ou seja, longe de zero. No entanto, o argumento utilizado no Teorema 3.7 a seguir depende fortemente de um número finito de  $D_k$  diferentes. Sua prova permite uma generalização para o caso de funções não suaves, mas que não será objeto desse texto (CONN; SCHEINBERG; VICENTE, 2009).

Para tal, são necessárias duas hipóteses adicionais:

**Hipótese 3.5.** *O conjunto  $\mathcal{D}$  das bases positivas, usado no algoritmo, é finito,*

e

**Hipótese 3.6.** *A função  $f$  é continuamente diferenciável em um conjunto aberto contendo  $N(x_0)$ .*

O Teorema 3.7 prova a convergência neste caso.

**Teorema 3.7.** *Sejam as hipóteses 3.1, 3.2, 3.5 e 3.6 satisfeitas. Então a sequência dos iterantes  $\{x_k\}$  tem ponto limitante  $x_*$  para qual*

$$\nabla f(x_*) = 0$$

*Demonstração.* Lembrando das definições adotadas para  $P_k$  e de iteração malsucedida, o que segue é verdadeiro para qualquer iteração malsucedida  $k$ :

$$\begin{aligned}
f(x_k) &\leq \min_{x \in P_k} f(x) = \min_{x \in \{x_k + \alpha_k d : d \in D_k\}} f(x) \\
&= \min_{d \in D_k} f(x_k + \alpha_k d) \\
&= \min_{d \in D_k} f(x_k) + \nabla f(x_k + t_{k,d} \alpha_k d)^T (\alpha_k d) \\
&= f(x_k) + \alpha_k \min_{d \in D_k} \nabla f(x_k + t_{k,d} \alpha_k d)^T d,
\end{aligned}$$

onde  $t_{d,k} \in (0,1)$  depende de  $k$  e  $d$ , e, conseqüentemente:

$$0 \leq \min_{d \in D_k} \nabla f(x_k + t_{k,d} \alpha_k d)^T d.$$

O Corolário 3.4 mostrou a existência de uma subsequência de iterações malsucedidas  $\{k_j\}$  as quais  $\lim_{i \rightarrow +\infty} \alpha_{k_i} = 0$  e  $\lim_{i \rightarrow +\infty} x_{k_i} = x_*$ .

A inequação acima é válida para essa subsequência  $\{k_j\}$ . Uma vez que o número de bases é finita, então existe ao menos um  $D_* \subset \mathcal{D}$  que é usado um número infinito de vezes em  $\{k_j\}$ . Assim,

$$0 \leq \min_{d \in D_*} \nabla f(x_*)^T d.$$

Mas, pelo Teorema 3.1, para todo vetor não nulo  $v \in \mathbb{R}^n$ , existe  $d \in D$  tal que

$$v^T d > 0,$$

nos restando apenas a possibilidade que  $\nabla f(x_*) = 0$ . ■

### 3.4 OTIMIZAÇÃO UNIDIMENSIONAL

Nesta seção, serão apresentados dois métodos de otimização unidimensional que não dependem do uso de derivadas: a Busca pela Seção Áurea e a Interpolação Parabólica Sucessiva. Esses métodos são úteis quando se deseja encontrar o mínimo de uma função em apenas uma variável.

#### 3.4.1 Busca pela Seção Áurea

O método da Busca pela Seção Áurea, também conhecido como Método de Fibonacci, é utilizado para encontrar máximos e mínimos de funções unimodais, ou seja, que possuem apenas um máximo ou um mínimo em seu domínio.

O método inicia com a seleção de um intervalo conhecido que contém o mínimo desejado, denotado como  $[a, b]$ . Dentro desse intervalo, dois pontos são selecionados,  $x_1$  e  $x_2$ , tais que  $a < x_1 < x_2 < b$ , e a função  $f$  é avaliada nesses pontos.

Caso  $f(x_1) \leq f(x_2)$ , mudamos o intervalo de busca para  $[a, x_2]$ . Caso contrário, isto é,  $f(x_1) > f(x_2)$ , mudamos o intervalo de busca para  $[x_1, b]$ .

A redução do intervalo segue uma proporção determinada pela Seção Áurea, como calcularemos a seguir.

No caso do primeiro, essa razão será dada por:

$$c = \frac{x_2 - a}{b - a},$$

que, resolvendo para  $x_2$ , resulta em:

$$x_2 = (1 - c)a + cb.$$

No segundo caso, a razão será dada por:

$$c = \frac{b - x_1}{b - a},$$

que, resolvendo para  $x_1$ , obtemos:

$$x_1 = ca + (1 - c)b.$$

A determinação de  $x_1$  e  $x_2$  é crucial para a iteração do método. Uma vez que os valores de  $a$  e  $b$  são conhecidos, seguiremos com busca pelo valor de  $c$ .

Note que, se  $f(x_1) < f(x_2)$ , então o novo intervalo será  $[a, x_2]$ . Desse modo, reutilizaremos o ponto  $x_1$ , fazendo dele o novo ponto  $x_2$ :

$$x_2^{\text{nov}} = x_1^{\text{vel}} = ca + (1 - c)b. \quad (3.4)$$

Além disso,

$$x_2^{\text{nov}} = (1 - c)a + cx_2^{\text{vel}} = (1 - c)a + c[(1 - c)a + cb]. \quad (3.5)$$

Igualando as equações (3.4) e (3.5), e realizando as devidas simplificações, temos:

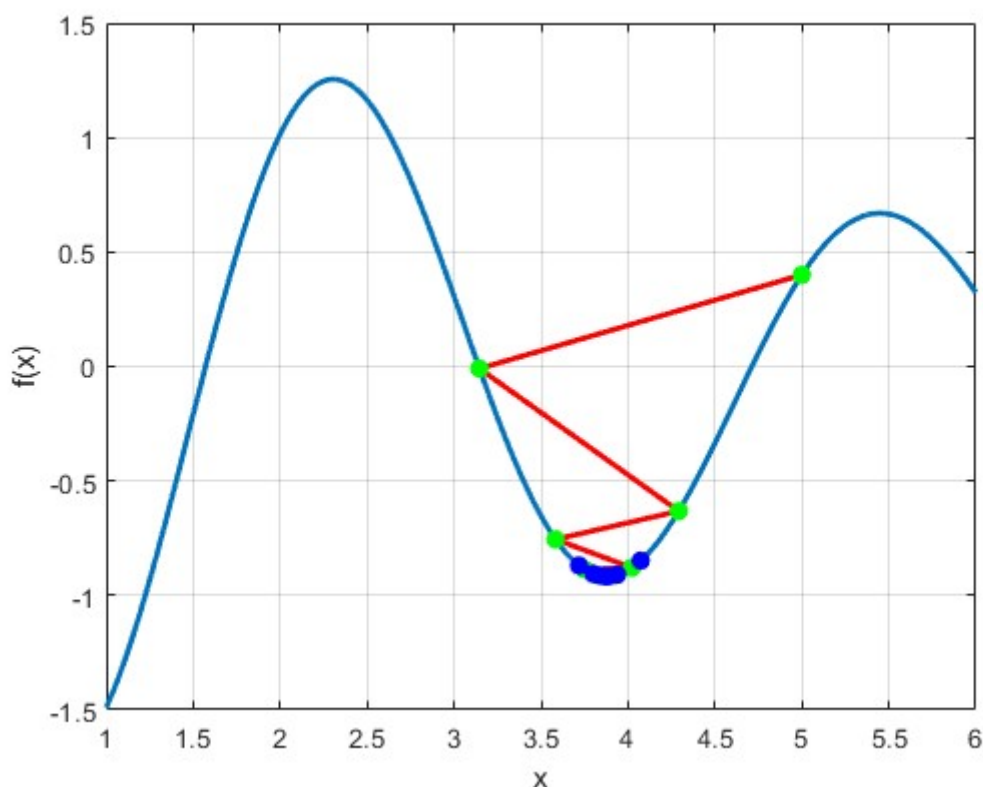
$$c^2 + c - 1 = 0$$

da qual, tomaremos apenas a raiz positiva  $c = \frac{-1 + \sqrt{5}}{2} \approx 0,6180 \Rightarrow \varphi = 1 + c$ . Desse modo, obtemos o valor de  $c$ , que nos permite calcular os novos valores de  $x_1$  e  $x_2$  a cada nova iteração. Ao atingir um determinado valor de tolerância  $t$  do intervalo  $[x_1, x_2]$ , damos o método como concluído. Esse valor pode ser determinado a critério do pesquisador, baseado no quão exata é a necessidade da resposta. um valor de tolerância menor implicará num número maior de iterações do método, fazendo com que mais tempo seja necessário em sua utilização. Em geral, podemos utilizar  $t = 10^{-5}$ . (GERALD; WHEATLEY, 2004)

Para ilustrar a convergência por meio do método, utilizaremos, como exemplo, a função  $f(x) = -2e^{-0.2x} \cdot \sin(2x)$ . Os detalhes sobre o desempenho do método, incluindo o número de iterações, o tempo de execução e o erro, serão abordados no Capítulo 4. Neste momento, vamos nos concentrar nos passos da convergência do método.

A Figura 3.2 ilustra a progressão dos intervalos, conforme determinado pelo método, em direção ao mínimo da função, partindo do o intervalo original.

Figura 3.2 – Aproximação do Mínimo pelas Iterações no Método da Busca pela Seção Áurea



Fonte: Elaborado pelo autor.

Note que, em cada iteração, o intervalo de busca, destacado em vermelho e delimitado pelos pontos verdes, é reduzido com base na razão previamente calculada, aproximando-se do mínimo da função dentro do intervalo considerado.

Por fim, o Algoritmo 3.3 descreve o funcionamento do método.

Heath (2018) argumenta que, na prática, é difícil esperar que uma função seja unimodal, e que, nesses casos, deve-se adotar um intervalo razoavelmente próximo do mínimo. Embora o método da Busca pela Seção Áurea sempre convirja, é possível que não se encontre um mínimo global ou mesmo local, a menos que a função seja unimodal no intervalo de partida.

**Algoritmo 3.3:** Algoritmo da Busca pela Seção Áurea (GERALD; WHEATLEY, 2004)

**Dados:** Dados  $f(x)$  unimodal, com o mínimo em  $[a,b]$   
**Resultado:** Minimizar a função  $f$

```

1 enquanto  $x_2 - x_1 < t$  faça
2   Computar  $x_1 = a + (1 - c)(b - a)$  e  $x_2 = a + c(b - a)$ , com  $c = 0,6180$ ;
3   Calcule  $f_1 = f(x_1)$  e  $f_2 = f(x_2)$ ;
4   se  $f_2 > f_1$  então
5      $b = x_2$ ;
6      $x_2 = x_1$ ;
7      $f_2 = f_1$ ;
8      $x_1 = a + (1 - c)(b - a)$ ;
9      $f_1 = f(x_1)$ ;
10  senão
11     $a = x_1$ ;
12     $x_1 = x_2$ ;
13     $f_1 = f_2$ ;
14     $x_2 = a + c(b - a)$ ;
15     $f_2 = f(x_2)$ ;
16  fim
17 fim
```

**3.4.2 Interpolação Parabólica Sucessiva**

Se compararmos o método da Busca pela Seção Áurea com a Interpolação Parabólica Sucessiva, notaremos que enquanto o primeiro se assemelha aos métodos de bissecção, o segundo emprega polinômios de grau dois para se aproximar das curvas da função e encontrar seu mínimo.

Desse modo, dada uma função  $f$  unimodal no intervalo inicial, buscaremos sua aproximação por meio de uma parábola. Assim, sejam  $x_1$ ,  $x_2$  e  $x_3$  três pontos do domínio de  $f$ , de modo que  $x_1 < x_2 < x_3$ , e sejam  $f(x_1)$ ,  $f(x_2)$  e  $f(x_3)$  suas imagens. A seguir, usando a Fórmula de Coeficientes de Lagrange, podemos expressar a parábola temos:

$$p(x) = f(x_1) \frac{(x - x_2)(x - x_3)}{(x_1 - x_2)(x_1 - x_3)} + f(x_2) \frac{(x - x_1)(x - x_3)}{(x_2 - x_1)(x_2 - x_3)} + f(x_3) \frac{(x - x_1)(x - x_2)}{(x_3 - x_1)(x_3 - x_2)}$$

A partir dessa expressão, buscamos determinar o ponto mínimo de  $p(x)$ . Como  $p(x)$  é uma parábola, sua derivada é simples de calcular, o que nos permite encontrar o ponto mínimo sem necessariamente derivar a função  $f$ , mas sim apenas o polinômio quadrático  $p$ .

Assim, calcularemos  $p'(x_0) = 0$  que, após as devidas simplificações, resulta em:

$$x_0 = \frac{x_1 + x_2}{2} - \frac{(f(x_2) - f(x_1))(x_3 - x_1)(x_3 - x_2)}{2[(x_2 - x_1)(f(x_3) - f(x_2)) - (f(x_2) - f(x_1))(x_3 - x_2)]}$$

Como todos os valores de  $x_1$ ,  $x_2$  e  $x_3$ , bem como  $f(x_1)$ ,  $f(x_2)$  e  $f(x_3)$  são todos conhecidos,  $x_0$  é um número real.

A seguir, fazemos a seguinte avaliação:

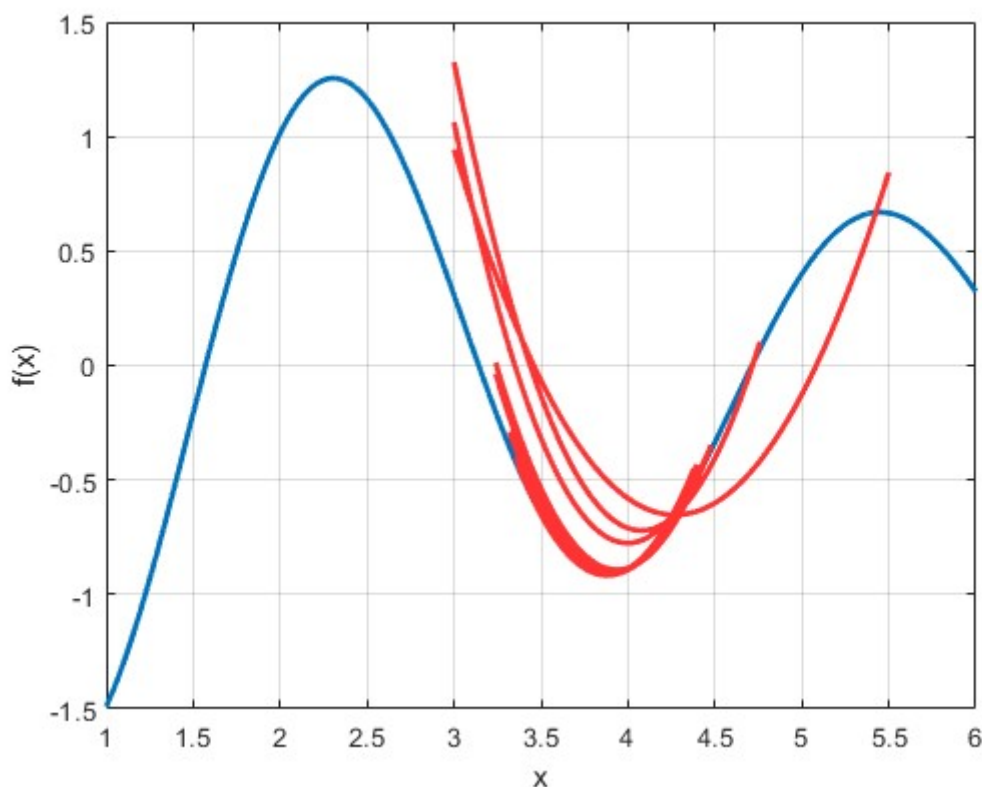
- Se  $f(x_0) < f(x_2)$ , então  $x_1^{\text{nov}} = x_1^{\text{velho}}$ ,  $x_2^{\text{nov}} = x_0$  e  $x_3^{\text{nov}} = x_2^{\text{velho}}$  e repetimos o procedimento para os três pontos.
- Se  $f(x_0) > f(x_2)$ , então  $x_1^{\text{nov}} = x_0$ ,  $x_2^{\text{nov}} = x_2^{\text{velho}}$  e  $x_3^{\text{nov}} = x_2^{\text{velho}}$  e repetimos o procedimento para os três pontos até que  $x_3 - x_1$  seja menor que alguma tolerância  $t$  escolhida. Nesse ponto, adotamos  $x_2$  como mínimo de  $f$ .

O mínimo encontrado, pelas aproximações por meio das parábolas correspondentes, com erro mínimo, ao mínimo da função. (HEATH, 2018)

Assim como no método da Busca pela Seção Áurea, as iterações prosseguem até que uma tolerância  $t$ , onde  $t$  representa a diferença entre  $x_1$  e  $x_3$ . Novamente, seu valor fica a critério do utilizador, sendo razoável adotar  $t = 10^{-5}$ .

Utilizando a mesma função,  $f(x) = -2e^{-0.2x} \cdot \sin(2x)$ , vamos ilustrar como as parábolas geradas nas iterações se aproximam do mínimo da função. Na Figura 3.3, a função original é representada em azul, enquanto as parábolas geradas pelas iterações são destacadas em vermelho.

Figura 3.3 – Aproximação do Mínimo pelas Iterações no Método da Interpolação Parabólica Sucessiva



Fonte: Elaborado pelo autor.

Observe que, conforme as iterações avançam, as parábolas se aproximam cada vez mais do formato da função. Por consequência, o mínimo das parábolas também se aproxima cada vez mais do mínimo da função, até que seus valores coincidam, ou que exista um erro tolerável, conforme desejado pelo utilizador.

O algoritmo 3.4 descreve o funcionamento do método:

**Algoritmo 3.4:** Algoritmo da Interpolação Parabólica Sucessiva

**Dados:** Dados  $f(x)$  unimodal e  $t$  uma tolerância

**Resultado:** Minimizar a função  $f$

```
1 enquanto  $x_3 - x_1 < t$  faça
2   | Computar  $f(x_1)$ ,  $f(x_2)$  e  $f(x_3)$ ;
3   | Calcular  $x_0$  e  $f(x_0)$ ;
4   | se  $f(x_0) > f(x_2)$  então
5   |   |  $x_1 = x_1$ ;
6   |   |  $x_2 = x_0$ ;
7   |   |  $x_3 = x_2$ ;
8   | senão
9   |   |  $x_1 = x_0$ ;
10  |   |  $x_2 = x_2$ ;
11  |   |  $x_3 = x_2$ ;
12  | fim
13 fim
```

Reiteramos a importância de considerar a hipótese da função ser unimodal, pelo menos em escala local, para assegurar a convergência das iterações em direção ao mínimo.

## 4 ALGUMAS IMPLEMENTAÇÕES E COMPARAÇÕES ENTRE OS MÉTODOS

Os testes de funções para otimização desempenham um papel crucial no desenvolvimento e aprimoramento de algoritmos de otimização. Eles permitem uma avaliação objetiva dos algoritmos em cenários controlados e auxiliam na seleção e ajuste dos métodos de otimização mais adequados para problemas do mundo real. Essas funções de teste são escolhidas de forma a representar diferentes desafios e características encontradas em problemas reais de otimização, permitindo uma análise completa e comparativa dos métodos disponíveis.

Neste capítulo, iremos aplicar alguns dos algoritmos que foram apresentados ao longo deste texto em algumas funções e, em seguida, compararemos os resultados obtidos.

Todos os dados podem variar se os testes forem replicados em máquinas que possuem *hardwares* diferentes. Os testes a seguir foram realizados em um computador equipado com um processador Ryzen 5600X em *stock*, refrigerado com um *water cooler* e suprido em uma placa mãe de *chipset* B550M.

Cada teste foi repetido 100 vezes para mitigar possíveis distorções em diferentes tentativas. Os valores apresentados nas tabelas refletem a média desses resultados.

### 4.1 FUNÇÕES DE $\mathbb{R}^2$ EM $\mathbb{R}$

Dadas suas diferenças, desafios e particularidades, selecionamos quatro funções para executar os testes dos algoritmos de Nelder-Mead, como representante dos modelos baseados em simplex e o Algoritmo da Busca Coordenada, como representante dos métodos de busca direta direcionais. Esses são os métodos mais utilizados quando se pensa em otimização sem derivadas, e convém avaliar sua funcionalidade.

As funções selecionadas são a Função de Himmelblau, Função de Beale, Função de Booth e a Função Three-Hump Camel. Elas são apenas alguns exemplos do amplo conjunto de funções utilizadas nos testes de otimização e representam diferentes desafios, como formas não lineares complexas, múltiplos mínimos locais e superfícies irregulares. Ao aplicar algoritmos de otimização a essas funções, podemos comparar o desempenho e a eficácia desses algoritmos na busca do mínimo global.

Todas essas funções não satisfazem às hipóteses dos métodos completamente em seu domínio. Porém, localmente, perto dos pontos de mínimo, as condições necessárias são cumpridas, e torna-se viável sua utilização.

Em cada uma delas, analisaremos, partindo de um mesmo ponto, donde há convergência, qual foi o número de iterações necessárias, o erro estimado, e o tempo de máquina. A partir desses dados, traçaremos conclusões sobre os métodos.

Os algoritmos foram escritos em MATLAB e estão disponíveis nos apêndices desse texto.



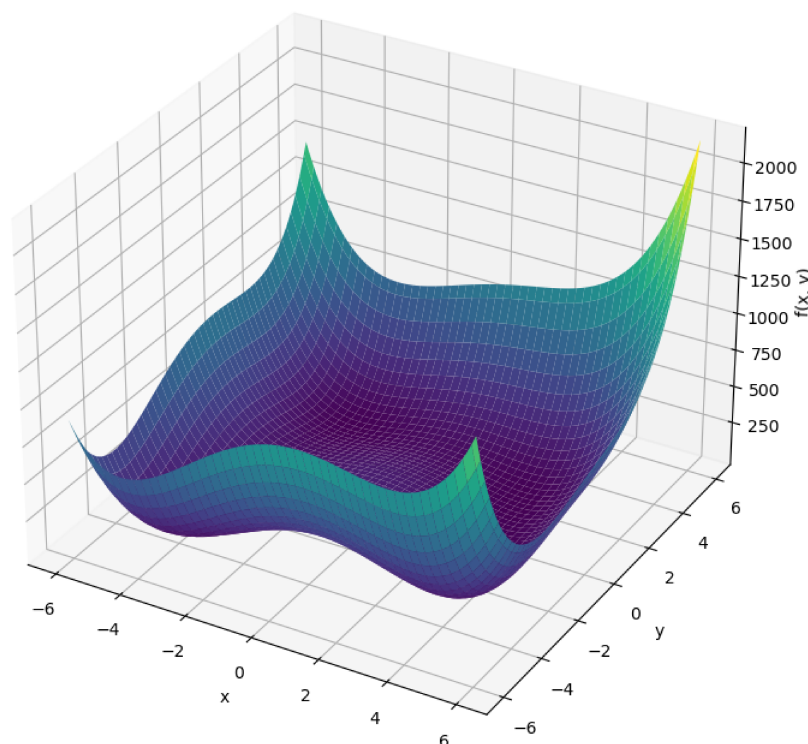
### 4.1.1 Função de Himmelblau

Uma das funções amplamente utilizadas para otimização é a função de Himmelblau, dada por

$$f(x,y) = (x^2 + y - 11)^2 + (x + y^2 - 7)^2,$$

e está presente na Figura 4.1 com curvas de nível presentes na Figura 4.2.

Figura 4.1 – Função de Himmelblau



Fonte: Elaborado pelo autor.

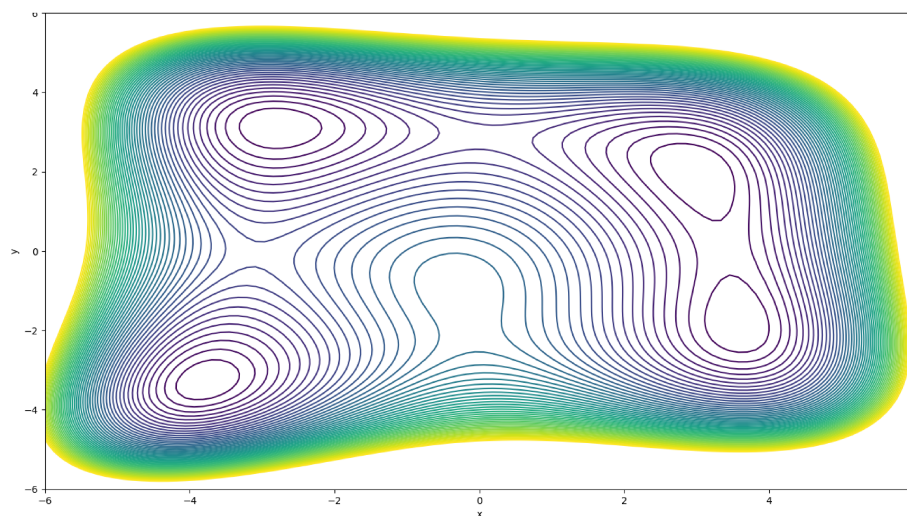
Proposta por Himmelblau *et al.* (1972), essa função é um exemplo desafiador de otimização com múltiplos mínimos locais. A função de Himmelblau é uma função de duas variáveis e possui uma forma não linear complexa. É frequentemente utilizada como um *benchmark* para testar a capacidade dos algoritmos de otimização de encontrar o mínimo global da função.

Seu mínimo global conhecido é 0, atingido em (3,2), (-2.805118,3.131312), (-3.779310,-3.283186) e (3.584428,-1.848126).

Para busca esse mínimo, aplicamos os algoritmos de Nelder-Mead, presente no Apêndice A.1 e da Busca Coordenada, presente no Apêndice A.2, partindo do mesmo ponto inicial, (1,1) e obtivemos os resultados presentes no Quadro 4.1.

Observamos que o Método da Busca Coordenada precisou de menos iterações, atingiu exatamente o ponto de mínimo e levou menos tempo para concluir o resultado, ao passo que o Método de Nelder-Mead foi mais lento, levou mais iterações e chegou

Figura 4.2 – Curvas de Nível da Função de Himmelblau



Fonte: Elaborado pelo autor.

Quadro 4.1 – Métodos de Nelder-Mead e Busca Coordenada aplicados à Função de Himmelblau

Algoritmo	N. de Iterações	Erro	Tempo de Máquina (segundos)
Nelder-Mead	45	$7.2618 \times 10^{-11}$	0.016087
Busca Coordenada	20	0	0.012549

Fonte: Elaborado pelo autor.

ao erro na ordem de  $10^{-11}$ , o que é praticamente irrelevante. Ambos atingiram o ponto de mínimo em (3,2), que era o mais próximo do ponto inicial (1,1).

#### 4.1.2 Função de Beale

Outra função comumente empregada é a Função de Beale, dada por:

$$f(x,y) = (1.5 - x + xy)^2 + (2.25 - x + xy^2)^2 + (2.625 - x + xy^3)^2,$$

com gráfico presente na Figura 4.3 e curvas de nível presentes em 4.4.

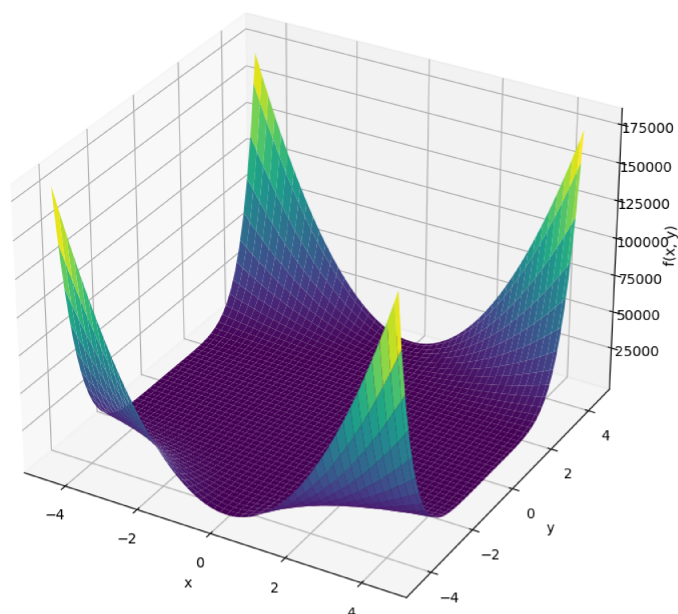
Assim como a função de Himmelblau, a função de Beale possui vários mínimos locais e é usada como exemplo ilustrativo em problemas de otimização.

Seu mínimo global é 0 em (3,0.5). (SURJANOVIC; BINGHAM, 2013a)

Para busca esse mínimo, aplicamos os algoritmos partindo do mesmo ponto inicial, (5,5) e obtivemos os resultados presentes no Quadro 4.2.

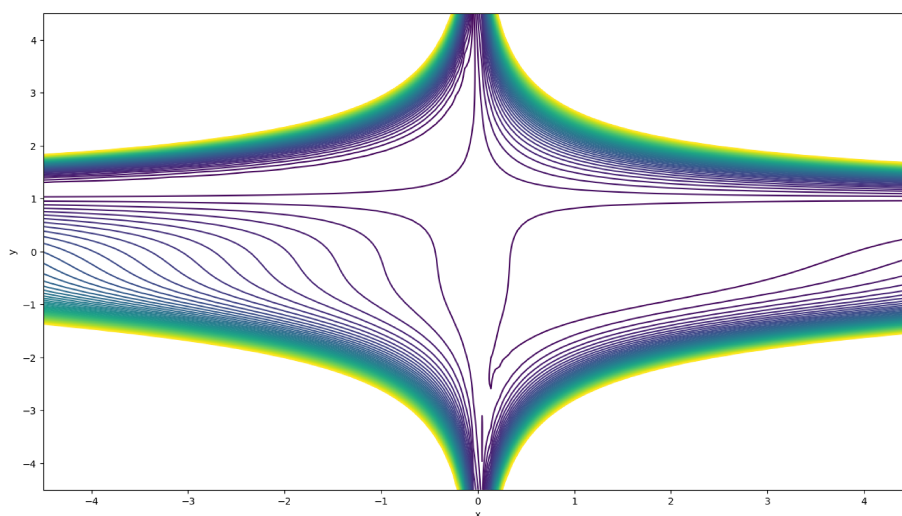
Novamente a Busca Coordenada necessitou de um número menor de iterações e chegou ao ponto mínimo exatamente, enquanto Nelder-Mead precisou de mais iterações, chegando a um erro de ordem de  $10^{-10}$ , novamente praticamente irrelevante. A diferença nesse caso foi no tempo de máquina, em que, apesar de ter quase o triplo de iterações, Nelder-Mead teve tempo de máquina menor.

Figura 4.3 – Função de Beale



Fonte: Elaborado pelo autor.

Figura 4.4 – Curvas de Nível da Função de Beale



Fonte: Elaborado pelo autor.

### 4.1.3 Função de Booth

A função de Booth também é uma função popular nos testes de otimização. Ela é dada por:

$$f(x,y) = (x + 2y - 7)^2 + (2x + y - 5)^2,$$

e podem ser vistos seu gráfico na Figura 4.5 e curvas de nível na Figura 4.6

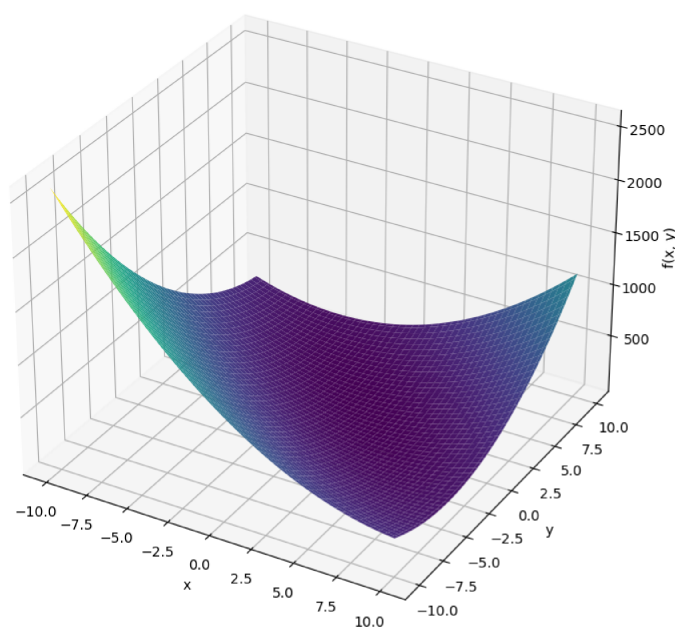
Embora não haja uma publicação específica associada a ela, a função de Booth é conhecida por sua forma característica. Essa função de duas variáveis apresenta múltiplos mínimos locais e desafia os algoritmos de otimização na busca do mínimo

Quadro 4.2 – Métodos de Nelder-Mead e Busca Coordenada aplicados à Função de Beale

Algoritmo	N. de Iterações	Erro	Tempo de Máquina (segundos)
Nelder-Mead	62	$4.5126 \times 10^{-10}$	0.011488
Busca Coordenada	26	0	0.012557

Fonte: Elaborado pelo autor.

Figura 4.5 – Função de Booth



Fonte: Elaborado pelo autor.

global.

Seu mínimo global é 0, em (1,3). (SURJANOVIC; BINGHAM, 2013b)

Para busca esse mínimo, aplicamos os algoritmos partindo de (1,1) e obtivemos os resultados presentes no Quadro 4.3.

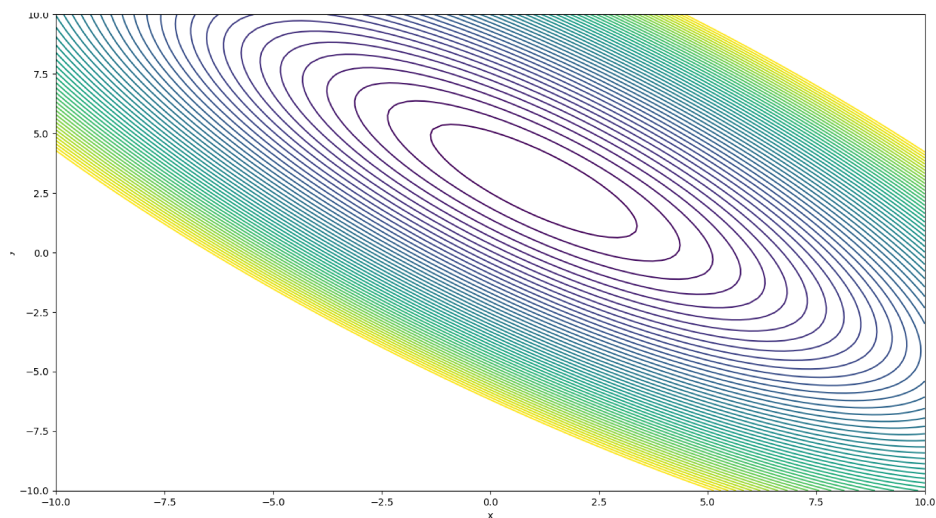
Quadro 4.3 – Métodos de Nelder-Mead e Busca Coordenada aplicados à Função de Booth

Algoritmo	N. de Iterações	Erro	Tempo de Máquina (segundos)
Nelder-Mead	53	0	0.011523
Busca Coordenada	19	0	0.013212

Fonte: Elaborado pelo autor.

Mais uma vez a Busca Coordenada necessitou de um número menor de iterações, quase um terço do número necessário em Nelder-Mead. Ambas alcançaram o erro zero, e, novamente, Nelder-Mead teve um tempo de máquina menor, apesar do

Figura 4.6 – Curvas de Nível da Função de Booth



Fonte: Elaborado pelo autor.

maior número de iterações.

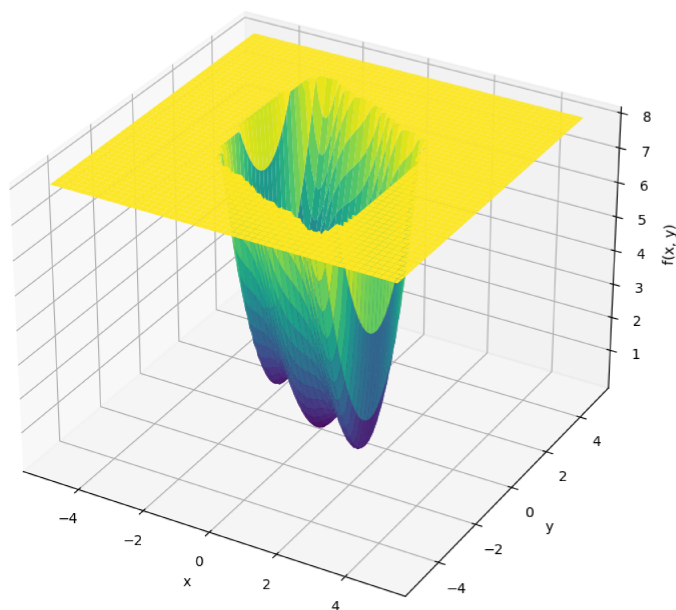
#### 4.1.4 Função Three-Hump Camel

Por fim, a função Three-Hump Camel é outra função comumente utilizada em testes de otimização, e é dada por:

$$f(x,y) = 2x^2 - 1.05x^4 + (x^6/6) + xy + y^2,$$

com gráfico presente na Figura 4.7 e curvas de nível presentes em 4.8.

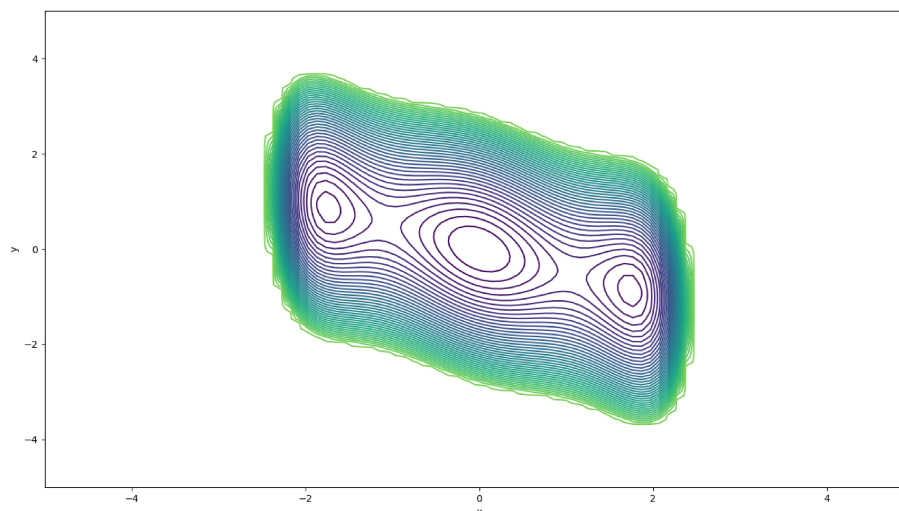
Figura 4.7 – Função Three-Hump Camel



Fonte: Elaborado pelo autor.



Figura 4.8 – Curvas de Nível da Função Three-Hump Camel



Fonte: Elaborado pelo autor.

Essa função também possui múltiplos mínimos locais e é empregada para avaliar a capacidade dos algoritmos de otimização de encontrar o mínimo global, tendo seu mínimo em 0, ocorrendo em (0,0). (SURJANOVIC; BINGHAM, 2013c)

Para atingir esse mínimo, aplicamos os algoritmos partindo de (2,1) e obtivemos os resultados presentes no Quadro 4.4.

Quadro 4.4 – Métodos de Nelder-Mead e Busca Coordenada aplicados à Função Three-Hump Camel

Algoritmo	N. de Iterações	Erro	Tempo de Máquina (segundos)
Nelder-Mead	27	0.00051059	0.018269
Busca Coordenada	20	0	0.011383

Fonte: Elaborado pelo autor.

Dessa vez, tendo número de iterações e tempo de máquina menor, o algoritmo da Busca Coordenada alcançou o erro zero, enquanto Nelder-Mead teve um erro na ordem de  $10^{-5}$ , já não tão irrelevante, teve mais iterações e um tempo de máquina maior (quase o dobro).

#### 4.1.5 Análise Geral

Uma análise geral da convergência das funções nos métodos implementados deixa claro que ambos os métodos são eficazes, desde que as condições sob as quais se propõem a trabalhar sejam respeitadas. Para que haja certeza quanto à sua convergência, é necessário cumprir todas as hipóteses estabelecidas em seus respectivos capítulos. Caso uma dessas hipóteses seja violada, não é possível garantir a convergência, embora ela possa ocorrer em muitos casos. É fundamental reconhecer

a importância dessas hipóteses para garantir a confiabilidade dos resultados obtidos por meio desses métodos. Portanto, ao aplicá-los, é necessário avaliar cuidadosamente as condições e restrições específicas de cada problema para evitar quaisquer desvios que possam comprometer a convergência esperada.

O Método de Nelder-Mead mostrou alcançar o mínimo com um número maior de iterações, porém, seu tempo de execução não é significativamente superior e, em alguns casos, pode até mesmo ser menor do que o tempo necessário para executar o método da Busca Coordenada. É interessante observar que, apesar do maior número de iterações, o Método de Nelder-Mead consegue chegar ao resultado mínimo desejado em todos os testes realizados. Isso indica sua eficácia e robustez para solucionar problemas de otimização.

Por outro lado, o método da Busca Coordenada, em contraste com o Método de Nelder-Mead, apresentou um menor número de iterações em todos os testes realizados e conseguiu atingir o resultado exato desejado em todos esses testes.

Esses resultados mostram que o método da Busca Coordenada é altamente eficiente em termos de convergência, exigindo menos iterações para alcançar o mínimo desejado. Além disso, o fato de ter alcançado o resultado exato em todos os testes é um indicativo da precisão desse método de otimização. Adicionalmente, o tempo de máquina equivalente ou menor necessário para executar o método da Busca Coordenada em comparação com o Método de Nelder-Mead é uma vantagem, pois permite uma execução mais rápida e eficiente.

Considerando o desempenho do Método de Nelder-Mead em termos de número de iterações e tempo de máquina, podemos concluir que ele é uma abordagem promissora e eficiente para problemas de otimização. Sua capacidade de alcançar o mínimo desejado, mesmo com um número maior de iterações, juntamente com um tempo de execução competitivo, torna-o uma opção viável e atrativa para diversos cenários de aplicação. Por outro lado, o método da Busca Coordenada se mostra como uma outra opção bastante viável e eficaz, apresentando um menor número de iterações em todos os testes, alcançando o resultado exato e tendo um tempo de máquina equivalente ou menor. Ambos os métodos têm suas vantagens e a escolha entre eles dependerá das características específicas do problema em questão e das preferências do usuário.

Ressaltamos que ao modificar o ponto inicial, há a possibilidade de não alcançar o mínimo desejado, visto que as funções podem não atender completamente às premissas dos métodos. É também possível alcançar um mínimo diferente ou não convergir para nenhum ponto.

## 4.2 FUNÇÕES DE $\mathbb{R}$ EM $\mathbb{R}$

Para executar os testes nos algoritmos de funções de  $\mathbb{R}$  em  $\mathbb{R}$  tratados nesse texto, a Busca pela Seção Áurea e a Interpolação Parabólica Sucessiva, selecionamos

duas funções unimodais com um único ponto de mínimo nos intervalos de teste.

Vamos comparar o desempenho dos algoritmos em cada função, considerando o número de iterações necessárias para atingir o mínimo com uma determinada tolerância, o tempo de execução e o erro encontrado.

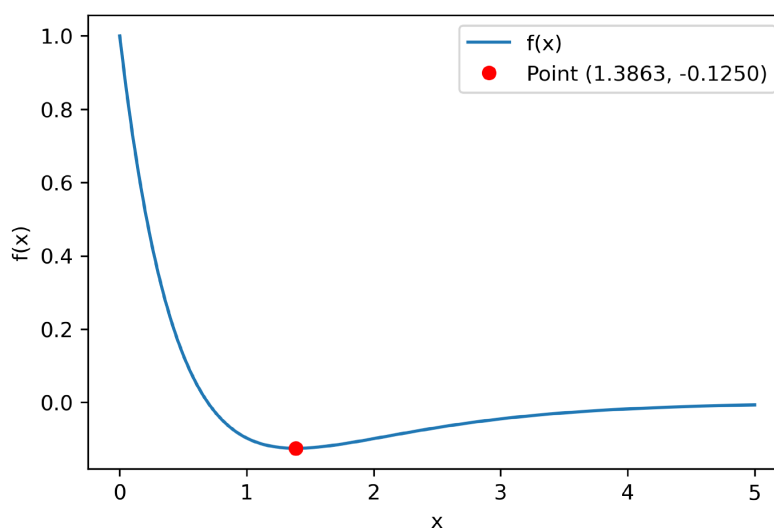
#### 4.2.1 Primeira Função

A primeira função escolhida para teste foi

$$f(x) = 2e^{-2x} - e^{-x},$$

cujo gráfico está presente na Figura 4.9.

Figura 4.9 – Primeira Função



Fonte: Elaborado pelo Autor.

No intervalo analisado,  $[1,2]$ , há um mínimo de  $-0.1250$ , atingido em  $x = 1.3863$ .

Nela, foram aplicados os algoritmos da Busca pela Seção Áurea, presente no Apêndice A.3, e da Interpolação Parabólica Sucessiva, presente no Apêndice A.4, e obtivemos os resultados presentes no Quadro 4.5.

Quadro 4.5 – Métodos Busca pela Seção Áurea e da Interpolação Parabólica Sucessiva aplicados à primeira função

Algoritmo	N. de Iterações	Erro	Tempo de Máquina (segundos)
Seção Áurea	24	$2.5813 \times 10^{-14}$	0.0029
Interpolação Parabólica	17	$10^{-5}$	0.0096

Fonte: Elaborado pelo autor.

Conforme evidenciado no Quadro 4.5, a Interpolação Parabólica Sucessiva precisou de um menor número de iterações para alcançar o resultado dentro do erro



aceitável. No entanto, é importante ressaltar que esse método exigiu mais de três vezes o tempo de processamento em comparação com a abordagem da Seção Áurea. Portanto, podemos concluir que, embora a Interpolação Parabólica Sucessiva tenha menos iterações, ela apresenta um custo computacional superior em relação à Seção Áurea. Esta última, embora demande um maior número de iterações, consegue atingir o resultado desejado em apenas um terço do tempo de processamento do método anteriormente mencionado.

Uma justificativa para isso reside no fato de que o Método da Seção Áurea requer apenas duas avaliações da função em cada iteração, em comparação com as quatro necessárias no Método da Interpolação Parabólica Sucessiva. Essa diferença contribui significativamente para a eficiência do processo de otimização, impactando diretamente os resultados obtidos.

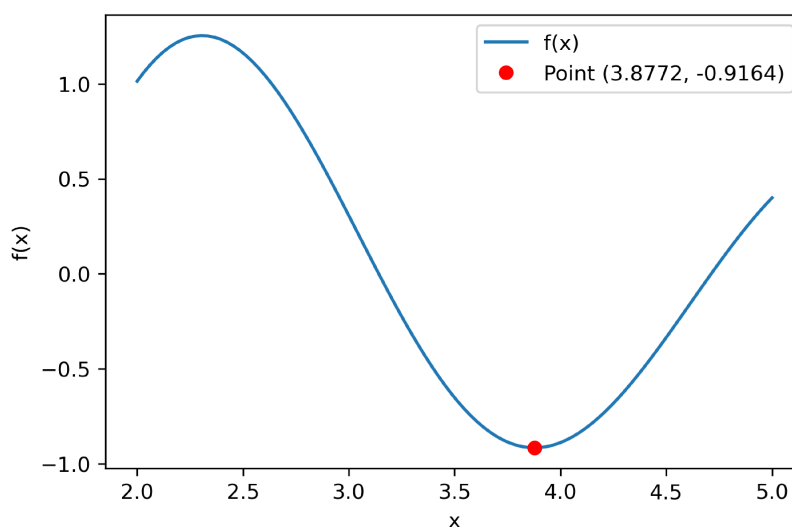
#### 4.2.2 Segunda Função

A segunda função escolhida para teste foi

$$f(x) = -2e^{-0.2x} \cdot \sin(2x),$$

cujo gráfico está presente na Figura 4.10.

Figura 4.10 – Segunda Função



Fonte: Elaborado pelo Autor.

No intervalo analisado,  $[2,5]$ , há um mínimo de  $-0.9164$ , atingido em  $x = 3.8772$ . Nela, foram aplicados os algoritmos e obtivemos os resultados explícitos no Quadro 4.6.

Nessa segunda função, observamos o mesmo cenário explicitado na primeira: Interpolação Parabólica Sucessiva com um menor número de iterações, porém com um tempo de máquina maior, e a Busca pela Seção Áurea, com mais iterações, porém

Quadro 4.6 – Métodos Busca pela Seção Áurea e da Interpolação Parabólica Sucessiva aplicados à segunda função

Algoritmo	N. de Iterações	Erro	Tempo de Máquina (segundos)
Seção Áurea	27	$4.4764 \times 10^{-7}$	0.0027
Interpolação Parabólica	19	$4.4764 \times 10^{-7}$	0.0068

Fonte: Elaborado pelo autor.

com tempo de máquina menor. Tudo isso, novamente, justificável pelo mesmo motivo: o número de avaliações da função a cada iteração.

Ao analisar os resultados de ambos os métodos, podemos concluir que eles são eficazes quando aplicados a funções ou intervalos que satisfazem suas premissas, tornando-se ferramentas robustas para calcular mínimos. No entanto, caso as funções não atendam às condições estabelecidas, existe a opção de restringir ainda mais o intervalo. Alternativamente, é possível aplicar os métodos mesmo em casos desafiadores, embora com o risco de não obter um resultado satisfatório.

## 5 ENSINO, A BNCC E OTIMIZAÇÃO

No ensino de matemática, a Modelagem Matemática e Resolução de Problemas são duas metodologias que andam de mãos dadas, existindo uma estreita relação entre as duas. Através dessas abordagens pedagógicas, os estudantes são provocados a desenvolver habilidades, bem como a aplicá-las de forma prática e contextualizada.

A Modelagem Matemática é uma metodologia que desempenha um papel significativo no ensino e aprendizado da matemática. Segundo Barbosa (2001), pode-se dizer que, em geral, a Modelagem Matemática se refere à aplicação da matemática em outras áreas do conhecimento ou no dia a dia, configurando um ambiente de aprendizagem, onde os alunos são convidados a questionarem e investigarem as situações do mundo real, utilizando a matemática como ferramenta. Ao envolver os alunos na criação de modelos matemáticos, permite-se que os estudantes explorem problemas autênticos e contextualizados, desenvolvendo uma compreensão mais profunda dos conceitos matemáticos e sua aplicação prática.

Bassanezi (1994) destaca que “a modelagem matemática é um processo dinâmico de busca de modelos adequados, que sirvam de protótipos de alguma entidade”. Ele ainda destaca que a modelagem matemática é comumente usada em várias situações, por engenheiros, biólogos, físicos, e todos os outros cientistas, com o intuito de transformar situações reais em termos matemáticos, a fim de resolver problemas, interpretar resultados, e assim, ter clareza e consciência na tomada de decisões.

Além disso, a Modelagem Matemática proporciona uma oportunidade para o desenvolvimento de habilidades de resolução de problemas. Conforme ressaltado por Thiel (2015), ela auxilia no desenvolvimento da teoria matemática, das habilidades de resolução de problemas aplicados e na análise da natureza e do papel dos modelos matemáticos na sociedade. Ao enfrentar desafios reais, os estudantes são incentivados a aplicar seu conhecimento matemático de forma criativa e crítica, buscando soluções efetivas.

Por sua vez, a Resolução de Problemas é considerada uma parte essencial do processo de ensino e aprendizado da matemática. Através da Resolução de Problemas, os alunos são encorajados a investigar e compreender conteúdos matemáticos, desenvolver estratégias e aplicá-las nas resoluções, além de relacionar a matemática com situações cotidianas e ver a face atraente e desafiadora da matemática. (CARVALHO; PIRES; GOMES, 2005)

No ínterim dessa metodologia, Polya (1995), a maior referência em termos de Resolução de Problemas, acrescenta que deve-se seguir quatro etapas em seu desenvolvimento: a compreensão do problema, quando fazemos sua modelagem; construção da estratégia de solução, quando estudamos o problema e determinamos o que deve ser feito; execução da estratégia, aplicando o decidido na etapa anterior; revisão da

solução, quando examinamos o resultado obtido, verificamos sua coerência e factibilidade.

As etapas propostas por Polya (1995) vão ao encontro das etapas propostas por Lesh e Doerr (2003) para a Modelagem Matemática, que compreendem: descrição do problema, modelando o problema em funções e outras ferramentas; manipulação do modelo, para gerar previsões ou ações relacionadas à situação original de resolução de problemas; tradução, onde os resultados da etapa serão traduzidos à luz do mundo real; verificação, onde os resultados encontrados serão verificados sobre sua acurácia.

Essa equivalência não é mero acaso: a relação entre Modelagem Matemática e Resolução de Problemas é intrínseca. A Modelagem Matemática oferece um contexto autêntico para a Resolução de Problemas, permitindo que os alunos apliquem conceitos matemáticos em situações reais e compreendam sua relevância. Conforme Bassanezi (2002, p.56) explica, “a modelagem matemática é um excelente contexto para a aprendizagem da resolução de problemas, já que a construção do modelo e a busca pela solução requerem um conjunto de habilidades e estratégias comuns”. A modelagem matemática desafia os estudantes a enfrentarem problemas complexos e, ao mesmo tempo, estimula o desenvolvimento de suas habilidades de resolução de problemas.

Em síntese, a Modelagem Matemática e a Resolução de Problemas estão interligadas e se complementam no contexto do ensino e aprendizado da matemática. A primeira proporciona uma abordagem prática e contextualizada, incentivando os alunos a aplicarem seus conhecimentos matemáticos em situações do mundo real, enquanto a segunda, por sua vez, estimula o pensamento crítico, a criatividade e a tomada de decisões, capacitando os alunos a enfrentarem desafios matemáticos de forma eficaz. Ao unir essas metodologias, os estudantes têm a oportunidade de desenvolver habilidades matemáticas sólidas e aplicáveis, preparando-se para enfrentar problemas complexos em diversas áreas da vida.

Usufruindo dessa complementariedade, a otimização combina a modelagem matemática, que permite descrever e formular um problema em termos matemáticos, com a resolução de problemas, que envolve a aplicação de técnicas e algoritmos para encontrar a solução ótima. Essa combinação de ferramentas é essencial para encontrar soluções eficientes e melhores em uma ampla variedade de campos e setores.

## 5.1 OTIMIZAÇÃO E A BNCC

A Base Nacional Comum Curricular (BNCC) foi homologada no Brasil em dezembro de 2017 e está sendo gradualmente implementada nas escolas do país. Seu processo de elaboração começou em 2015, envolvendo amplas discussões e consultas públicas.

A BNCC tem como objetivo estabelecer os conhecimentos, competências e ha-

bilidades essenciais que todos os estudantes brasileiros devem desenvolver ao longo da educação básica, abrangendo desde a educação infantil até o ensino médio. Seu propósito é definir uma base comum de aprendizado em todo o país, garantindo acesso igualitário a uma educação de qualidade para todos os alunos, independentemente de sua localização ou condição socioeconômica.

A BNCC está estruturada em áreas de conhecimento, como Linguagens, Matemática, Ciências da Natureza e Ciências Humanas, e define competências específicas que os estudantes devem adquirir em cada etapa da educação básica.

Cada área de conhecimento é composta por competências que, por sua vez, se desdobram em habilidades. Essas habilidades são numeradas de acordo com um padrão estabelecido, utilizando caracteres alfanuméricos (BRASIL, 2018).

No contexto do ensino médio, a área de conhecimento de Matemática e Suas Tecnologias apresenta cinco competências específicas.

Logo na primeira competência específica dela, encontramos oportunidades para a utilização das alternativas apresentadas neste trabalho no ensino. Especialmente na primeira habilidade dessa competência, que consiste em utilizar estratégias, conceitos e procedimentos matemáticos para interpretar situações em diversos contextos, sejam eles do cotidiano, das Ciências da Natureza, Ciências Humanas, questões socioeconômicas ou tecnológicas, divulgados por diferentes meios, contribuindo para uma formação geral (BRASIL, 2018, p. 532).

Nessa habilidade, já podemos identificar uma clara referência à análise de gráficos e taxas de variação. Por exemplo, a habilidade (EM13MAT101) propõe interpretar criticamente situações econômicas, sociais e fatos relacionados às Ciências da Natureza, que envolvam a variação de grandezas, por meio da análise de gráficos das funções representadas e das taxas de variação, com ou sem o auxílio de tecnologias digitais (BRASIL, 2018, p. 533).

Na segunda competência específica, que envolve a proposição ou participação em ações para investigar desafios do mundo contemporâneo e tomar decisões éticas e socialmente responsáveis, com base na análise de problemas sociais, como saúde, sustentabilidade e implicações da tecnologia no mundo do trabalho, também encontramos oportunidades. Em particular, a terceira habilidade (EM13MAT203) destaca a aplicação de conceitos matemáticos no planejamento, execução e análise de ações que envolvam o uso de aplicativos e a criação de planilhas, como controle de orçamento familiar, simuladores de cálculos de juros simples e compostos, entre outros, para tomar decisões (BRASIL, 2018, p. 534).

Na terceira competência específica, enunciada por

Utilizar estratégias, conceitos, definições e procedimentos matemáticos para interpretar, construir modelos e resolver problemas em diversos contextos, analisando a plausibilidade dos resultados e a adequação das soluções propostas, de modo a construir argumentação consistente (BRASIL, 2018, p. 535).

encontramos diversas possibilidades de implementação, especialmente nas seguintes habilidades:

(EM13MAT301) Resolver e elaborar problemas do cotidiano, da Matemática e de outras áreas do conhecimento, que envolvem equações lineares simultâneas, usando técnicas algébricas e gráficas, com ou sem apoio de tecnologias digitais.

(EM13MAT302) Construir modelos empregando as funções polinomiais de 1º ou 2º grau, para resolver problemas em contextos diversos, com ou sem apoio de tecnologias digitais.

[...]

(EM13MAT304) Resolver e elaborar problemas com funções exponenciais nos quais seja necessário compreender e interpretar a variação das grandezas envolvidas, em contextos como o da Matemática Financeira, entre outros.

(EM13MAT305) Resolver e elaborar problemas com funções logarítmicas nos quais seja necessário compreender e interpretar a variação das grandezas envolvidas, em contextos como os de abalos sísmicos, pH, radioatividade, Matemática Financeira, entre outros.

(EM13MAT306) Resolver e elaborar problemas em contextos que envolvem fenômenos periódicos reais (ondas sonoras, fases da lua, movimentos cíclicos, entre outros) e comparar suas representações com as funções seno e cosseno, no plano cartesiano, com ou sem apoio de aplicativos de álgebra e geometria. (BRASIL, 2018, p. 536)

Nessa competência, existem diversas possibilidades de utilizar algoritmos para a determinação de mínimos em problemas com diferentes tipos de funções, como polinomiais, exponenciais, logarítmicas e periódicas, incentivando explicitamente o uso de tecnologias digitais.

Na quarta competência específica, há uma instrução clara sobre o uso de algoritmos e linguagens de programação, buscando compreender e utilizar diferentes registros de representação matemática (algébrico, geométrico, estatístico, computacional etc.) na busca de soluções e comunicação de resultados de problemas (BRASIL, 2018, p. 538). Destaca-se a habilidade (EM13MAT405) que propõe utilizar conceitos iniciais de uma linguagem de programação na implementação de algoritmos escritos em linguagem corrente e/ou matemática (BRASIL, 2018, p. 539).

Por fim, na quinta competência específica, cujo objetivo é investigar e estabelecer conjecturas sobre diferentes conceitos e propriedades matemáticas, utilizando estratégias e recursos como observação de padrões, experimentações e diferentes tecnologias, destaca-se a habilidade (EM13MAT503) que propõe investigar pontos de máximo ou mínimo de funções quadráticas em contextos envolvendo superfícies, matemática financeira ou cinemática, com o apoio de tecnologias digitais (BRASIL, 2018, p. 541).

Ao longo de todo o documento, a BNCC enfatiza fortemente o uso da Modelagem Matemática e da Resolução de Problemas como abordagens fundamentais. Além disso, ela sugere a integração dessas duas metodologias, destacando a importância

de utilizar a Modelagem Matemática como uma ferramenta para resolver problemas de forma contextualizada e significativa. A BNCC reconhece que os processos de resolução de problemas, investigação, desenvolvimento de projetos e modelagem são formas privilegiadas de atividade matemática, sendo simultaneamente objetos e estratégias para a aprendizagem (BRASIL, 2018, p. 267).

No que diz respeito à resolução de problemas, os estudantes são encorajados a identificar os conceitos e procedimentos matemáticos necessários ou que possam ser aplicados na formulação matemática do problema (BRASIL, 2018, p. 535). Em alguns casos, é necessário identificar ou construir um modelo para obter respostas adequadas (BRASIL, 2018, p. 535).

Considerando o incentivo ao uso da modelagem matemática e da resolução de problemas na BNCC, é importante ressaltar que existem habilidades que podem ser exploradas no contexto da otimização, incluindo o uso de ferramentas computacionais prontas ou programáveis. Nesse sentido, a incorporação de métodos e algoritmos de otimização não diferenciáveis pode desempenhar um papel relevante no ensino de matemática no ensino médio, enriquecendo ainda mais a experiência educacional dos alunos.

## 6 PROPOSTA DE SEQUÊNCIA DIDÁTICA

A sequência didática proposta tem como objetivo integrar o estudo da teoria de funções com os conceitos básicos de otimização. Ela busca introduzir os alunos aos métodos mais simples de otimização, sem a necessidade do uso de derivadas, e promover o entendimento dos mesmos de forma geométrica e numérica, por meio de exemplos e aplicações práticas.

Essa proposta foi elaborada considerando sua aplicação no primeiro ano do ensino médio, quando os estudantes começam a explorar o estudo das funções. A sequência lógica de estudos nessa etapa começa com a Teoria de Conjuntos, avança para Relações e Funções e, nesse momento, pode-se introduzir a presente proposta.

Levando em conta o tempo necessário para o estudo de cada etapa e a realização da avaliação, estimamos que serão necessárias 14 aulas para a aplicação completa dessa sequência didática. Esse período permitirá aos alunos assimilar os conceitos de forma adequada e desenvolver habilidades práticas na resolução de problemas de otimização.

A sequência proposta está sujeita a variações de acordo com o público-alvo para o qual será aplicada. Pode ser necessário adicionar aulas extras para revisar pré-requisitos, fornecer mais exemplos e exercícios, ou, alternativamente, acelerar algumas etapas, dependendo do perfil da turma.

É essencial adaptar a abordagem de ensino às necessidades e ao ritmo de aprendizado dos estudantes. Observar o progresso e o engajamento da turma ao longo da sequência ajudará a identificar se é necessário aprofundar certos conceitos ou avançar mais rapidamente.

Portanto, é importante ter flexibilidade e estar aberto a ajustes durante a implementação da sequência, a fim de atender melhor às demandas e capacidades dos alunos. Dessa forma, poderemos proporcionar uma experiência de aprendizado mais personalizada e eficaz.

É importante ressaltar que, ao longo de toda a sequência, apesar de não estar presente nos documentos que norteiam o ensino médio, não devemos limitar nosso estudo apenas a funções de uma variável real ( $\mathbb{R}$  em  $\mathbb{R}$ ), mas também podemos, a critério do docente, abordar o caso de funções de duas variáveis reais ( $\mathbb{R}^2$  em  $\mathbb{R}$ ). Isso amplia as oportunidades de explorarmos casos mais interessantes de otimização e permitir a utilização de outros métodos de otimização. Dessa forma, estabelecemos a seguir as aulas para a aplicação da sequência didática proposta:

### 6.1 AULA 1 E 2: INTRODUÇÃO AO CONCEITO DE FUNÇÃO

Nessas duas aulas iniciais, é importante apresentar aos alunos o conceito de função como uma relação entre dois conjuntos. Podemos fazer uma analogia a uma



“máquina” que transforma um número de um conjunto em outro número de um conjunto diferente. Além disso, é necessário revisar o conceito de variável dependente e independente, para que os alunos compreendam a relação entre elas.

Será fundamental definir os termos domínio, contradomínio e imagem, a fim de estabelecer uma base sólida para o entendimento das funções. Para isso, podemos explorar exemplos de funções simples e acessíveis aos estudantes, como funções de custo, receita e lucro, velocidade, altura, peso e índice de massa corporal, entre outros. Dessa forma, os alunos poderão visualizar a aplicação dessas funções no contexto do seu dia a dia.

Por fim, é importante introduzir a notação funcional,  $f(x)$ , para representar uma função. Essa notação facilita a compreensão da relação entre a variável independente  $x$  e a variável dependente  $f(x)$ . Ao usar essa notação, os alunos poderão expressar e analisar as funções de maneira mais precisa e organizada.

## 6.2 AULA 3 E 4: GRÁFICOS DE FUNÇÕES

Na terceira e quarta aulas, será apresentado aos alunos o tema das representações gráficas das funções. Inicialmente, é importante revisar o conceito de plano cartesiano e pares ordenados, assim como a habilidade de localizar pontos no plano. Em seguida, procederemos com a construção de esboços de gráficos de funções.

Para isso, podemos utilizar tabelas como auxílio para o esboço dos gráficos, assim como aproveitar recursos tecnológicos, como softwares, para criar visualizações mais precisas. É recomendado utilizar as mesmas funções que foram trabalhadas nas aulas anteriores, a fim de manter uma conexão entre os temas abordados.

Posteriormente, será demonstrado aos alunos como interpretar informações importantes a partir dos gráficos, como identificar intervalos de crescimento, decréscimo, pontos de interceptação com os eixos e extremos. Essa análise permitirá que os alunos compreendam melhor o comportamento das funções e sua relação com os gráficos correspondentes.

Por fim, será abordado o conceito de curvas de nível em um gráfico, que são linhas que conectam pontos com a mesma altura ou valor da função. Isso possibilitará aos alunos explorar a representação visual das funções em três dimensões e compreender como as curvas de nível podem revelar informações adicionais sobre a função.

Ao incorporar esses elementos nas aulas, estaremos fornecendo aos alunos ferramentas importantes para analisar, interpretar e visualizar funções por meio de suas representações gráficas.

### 6.3 AULA 5 E 6: FUNÇÕES INVERSA, INJETIVIDADE, SOBREJETIVIDADE E BIJETIVIDADE

Nas quinta e sexta aulas, abordaremos os conceitos de função inversa, injetividade, sobrejetividade e bijetividade. Esses conceitos são fundamentais para compreender as propriedades das funções e suas relações com os conjuntos de partida e chegada.

Começaremos apresentando a ideia de função inversa, que representa a relação inversa entre os elementos do domínio e do contradomínio de uma função. Serão discutidos os critérios para determinar se uma função possui uma inversa e como encontrar sua expressão analítica.

Em seguida, exploraremos os conceitos de injetividade, sobrejetividade e bijetividade.

Durante a aula, utilizaremos exemplos práticos e gráficos para ilustrar essas definições. Os alunos serão incentivados a identificar se uma função é injetiva, sobrejetiva ou bijetiva, além de compreender as relações entre essas propriedades.

Ao compreender esses conceitos, os alunos poderão analisar e classificar funções com base em suas propriedades de injetividade, sobrejetividade e bijetividade. Essas noções são cruciais para a compreensão de diversos aspectos do estudo das funções e serão fundamentais em etapas posteriores da sequência didática.

### 6.4 AULA 7: OTIMIZAÇÃO COM FUNÇÕES

Na sétima aula, abordaremos o tema da otimização com funções, sem a utilização de derivadas. O objetivo é introduzir os alunos aos conceitos básicos de otimização e apresentar métodos simples para encontrar valores ótimos.

Começaremos revisando o conceito de otimização e sua importância na resolução de problemas matemáticos. Em seguida, exploraremos exemplos práticos de problemas de otimização simples, nos quais será possível encontrar os valores ótimos por meio de testes de pontos ou pela análise visual do gráfico da função.

Utilizando casos de aplicação cotidiana, como maximizar o lucro em um negócio ou minimizar o tempo de uma tarefa, os alunos serão incentivados a identificar os valores ótimos por meio de tentativas e aproximações.

Durante a aula, serão apresentados métodos simples para encontrar os valores ótimos, como o método de tentativa e erro, a análise do comportamento da função em intervalos relevantes e a comparação entre diferentes pontos. Serão realizados exemplos práticos e exercícios para que os alunos possam praticar e desenvolver suas habilidades de otimização.

Ao final da aula, os alunos deverão compreender que a otimização pode ser abordada mesmo sem o uso de derivadas, por meio de métodos simples de análise e

aproximação. Essa base será fundamental para o estudo posterior de métodos mais avançados de otimização.

## 6.5 AULA 8: TÉCNICAS DE OTIMIZAÇÃO DE $\mathbb{R}$ EM $\mathbb{R}$

Na oitava aula, abordaremos os métodos de otimização não-diferenciável de  $\mathbb{R}$  em  $\mathbb{R}$ , mais especificamente a busca pela seção áurea e a interpolação parabólica. Esses métodos são úteis quando a função não é diferenciável ou quando o uso de derivadas não é viável.

Começaremos explicando os fundamentos desses métodos, destacando suas características e princípios de funcionamento. Em seguida, realizaremos exemplos práticos e exercícios para demonstrar como esses métodos podem ser aplicados para otimizar funções.

No método da busca pela seção áurea, exploraremos como dividir iterativamente um intervalo em partes menores com base na proporção áurea, a fim de encontrar o ponto ótimo. Mostraremos como aplicar esse método passo a passo, ilustrando sua eficácia na otimização.

Já no método da interpolação parabólica, mostraremos como aproximar a função com uma parábola e encontrar o máximo ou mínimo local. Será enfatizado o processo de construção da parábola e como utilizá-la para determinar o ponto ótimo.

Durante a aula, os alunos serão incentivados a resolver exercícios práticos, aplicando os métodos da busca pela seção áurea e da interpolação parabólica. Isso proporcionará a oportunidade de praticar suas habilidades de otimização em situações reais.

Ao final da aula, os alunos devem estar familiarizados com esses métodos não-diferenciáveis de otimização, compreender seus princípios e ser capazes de aplicá-los para encontrar pontos ótimos em diferentes funções. Esses métodos ampliam o leque de ferramentas disponíveis para otimização e preparam os alunos para enfrentar problemas desafiadores em etapas posteriores da sequência didática.

## 6.6 AULA 9 E 10: TÉCNICAS DE OTIMIZAÇÃO DE $\mathbb{R}^2$ EM $\mathbb{R}$

Na nona e décima aulas, abordaremos os métodos de otimização não-diferenciável de  $\mathbb{R}^2$  em  $\mathbb{R}$ . Especificamente, introduziremos os métodos de Nelder-Mead e da Busca Direta, que são amplamente utilizados para otimização em problemas com múltiplas variáveis.

Começaremos explicando os fundamentos desses métodos, destacando sua geometria e princípios de funcionamento. Abordaremos como esses métodos podem ser aplicados para otimizar funções em duas variáveis.

No método de Nelder-Mead, exploraremos como um conjunto de pontos é atualizado iterativamente com base em sua avaliação na função objetivo. Mostraremos como essas atualizações podem levar ao ponto ótimo, levando em consideração as informações das curvas de nível da função.

No método da Busca Direta, discutiremos como a busca sistemática em uma região do espaço das variáveis pode levar à identificação do ponto ótimo. Exploraremos os algoritmos numéricos utilizados nesse método e como eles podem ser implementados para otimizar funções.

Durante a aula, os alunos serão incentivados a resolver exemplos práticos e exercícios, aplicando os métodos de Nelder-Mead e da Busca Direta. Serão fornecidos problemas que envolvem otimização em duas variáveis, permitindo aos alunos praticar suas habilidades de otimização.

Ao final da aula, os alunos deverão estar familiarizados com esses métodos de otimização não-diferenciável de  $\mathbb{R}^2$  em  $\mathbb{R}$ . Eles compreenderão os princípios desses métodos, incluindo a geometria envolvida, e serão capazes de aplicá-los para encontrar pontos ótimos em diferentes funções de duas variáveis. Esses métodos ampliam as opções de ferramentas disponíveis para otimização e preparam os alunos para enfrentar problemas mais complexos nas etapas seguintes da sequência didática.

## 6.7 AULA 11: EXERCÍCIOS PRÁTICOS DE OTIMIZAÇÃO

Na décima primeira aula, proporemos uma série de exercícios práticos de otimização, abordando tanto problemas que envolvam métodos diferenciáveis quanto não diferenciáveis. O objetivo é que os alunos apliquem os conhecimentos adquiridos ao longo da sequência didática e pratiquem suas habilidades de otimização.

Os exercícios serão projetados para desafiar os alunos a encontrar soluções ótimas utilizando tanto métodos analíticos quanto geométricos. Será incentivada a utilização da geometria e da intuição para compreender o comportamento das funções e identificar os pontos ótimos.

Os problemas propostos podem envolver situações práticas de otimização em diferentes contextos, como maximizar a área de um terreno, minimizar o custo de produção de um objeto, otimizar o tempo de viagem entre dois pontos, entre outros.

É altamente recomendado o uso de estudos de caso envolvendo situações reais, como empresas, indústrias, empreendimentos ou outros cenários relevantes. Esses estudos de caso proporcionam aos alunos a oportunidade de aplicar os conceitos de otimização em contextos concretos, tornando o aprendizado mais significativo e aplicável.

Por exemplo, é possível apresentar um estudo de caso de uma empresa que precisa maximizar seus lucros, levando em consideração variáveis como custos de produção, preços de venda e demanda de mercado. Os alunos podem ser desafiados

a identificar as variáveis relevantes, formular a função objetivo e utilizar os métodos de otimização estudados para encontrar a melhor solução para a empresa.

Outro exemplo seria um estudo de caso envolvendo uma indústria que busca minimizar o tempo de produção de um determinado produto, considerando fatores como capacidade de produção, disponibilidade de recursos e restrições de tempo. Os alunos podem analisar os dados fornecidos, modelar a situação como um problema de otimização e utilizar os métodos adequados para encontrar a solução ótima.

A inclusão de estudos de caso enriquece a aula, tornando-a mais prática e alinhada com situações do mundo real. Além de aplicar os conceitos aprendidos, os alunos também desenvolvem habilidades de análise crítica, resolução de problemas e tomada de decisões.

Ao final da aula, os alunos deverão ter praticado a aplicação dos métodos estudados, desenvolvido suas habilidades de otimização e aprimorado sua capacidade de encontrar soluções ótimas em diferentes tipos de problemas. Isso os preparará para enfrentar desafios mais complexos e aplicar os conhecimentos adquiridos em situações do mundo real.

## 6.8 AULA 12: PROGRAMAÇÃO DOS MÉTODOS ESTUDADOS

Na décima segunda aula, utilizaremos o tempo para mostrar aos alunos os métodos de otimização com técnicas de programação. O objetivo é expandir o conhecimento dos alunos, permitindo-lhes entender como esses métodos podem ser implementados de forma prática e eficiente. Começaremos revisando brevemente os métodos de otimização estudados até agora, reforçando seus conceitos fundamentais. Em seguida, introduziremos a ideia de programação dos métodos, destacando sua importância na resolução de problemas mais complexos.

Apresentaremos aos alunos diferentes técnicas de programação que podem ser utilizadas na implementação dos métodos de otimização. Isso pode incluir o uso de linguagens de programação, bibliotecas específicas, algoritmos e estruturas de dados relevantes.

Durante a aula, demonstraremos exemplos práticos de como programar os métodos de otimização estudados, mostrando a aplicação dos conceitos teóricos em códigos reais. Serão fornecidos exemplos de problemas de otimização e os alunos serão incentivados a programar soluções utilizando as técnicas apresentadas.

Ao final da aula, os alunos terão uma compreensão mais profunda dos métodos de otimização e sua implementação por meio de técnicas de programação. Eles estarão preparados para utilizar essas ferramentas em problemas reais, desenvolvendo suas habilidades de programação e resolução de problemas. Essa aula amplia o leque de possibilidades dos alunos, permitindo-lhes explorar e solucionar desafios de otimização de maneira mais eficiente e prática.

Essa aula pode ser adaptada de acordo com as condições estruturais da unidade educacional. É importante levar em consideração os recursos disponíveis, como laboratório de informática, equipamentos audiovisuais e conhecimento prévio dos alunos em programação.

Se a escola possui um laboratório de informática e os alunos têm alguma base em programação, é recomendado permitir que eles usem os algoritmos em seus computadores. Isso proporcionará uma experiência prática e interativa, permitindo que os alunos apliquem os conceitos aprendidos e desenvolvam suas habilidades de programação.

Caso a escola não possua um laboratório de informática ou os alunos não tenham conhecimento prévio em programação, pode-se adaptar a aula utilizando recursos como data-show ou televisão para apresentar os algoritmos aos alunos. Nesse caso, a ênfase pode ser dada à compreensão dos conceitos e à discussão sobre como os algoritmos funcionam, mesmo sem a implementação prática.

Se a escola não tiver acesso a nenhum desses equipamentos ou a aula não puder ser realizada nas condições mencionadas, a aula 12 pode ser suprimida sem causar um prejuízo grave ao entendimento do conteúdo. Nesse caso, é importante reforçar os conceitos teóricos dos métodos de otimização e sua aplicação, enfatizando a compreensão dos conceitos e a resolução de problemas práticos de otimização sem o uso de programação. O objetivo principal é adaptar a aula de acordo com as possibilidades e recursos disponíveis, garantindo que os alunos alcancem os objetivos de aprendizagem estabelecidos para a sequência didática.

## 6.9 AULA 13 E 14: REVISÃO E AVALIAÇÃO

Na décima terceira e décima quarta aulas, dedicaremos um tempo para revisar os conceitos abordados ao longo da sequência didática de forma a reforçar o aprendizado dos alunos. Faremos uma recapitulação dos principais tópicos estudados, enfatizando os conceitos-chave e suas aplicações.

Durante a revisão, resolveremos exercícios de fixação em sala de aula. Os exercícios selecionados devem abranger os diferentes temas estudados, permitindo que os alunos pratiquem a aplicação dos métodos de otimização e fortaleçam sua compreensão dos conceitos. Após a resolução dos exercícios, aplicaremos uma avaliação para verificar a compreensão dos alunos sobre o conteúdo apresentado. A avaliação pode incluir questões teóricas e problemas práticos de otimização, nos quais os alunos devem aplicar os métodos aprendidos para encontrar soluções.

A avaliação servirá como uma oportunidade de os alunos demonstrarem o conhecimento adquirido ao longo da sequência didática. Ela permitirá avaliar o nível de compreensão individual e identificar possíveis lacunas que ainda precisam ser abordadas.

Ao final da aula, realizaremos uma breve discussão sobre os resultados da avaliação, oferecendo feedback aos alunos e esclarecendo dúvidas. Também aproveitaremos esse momento para destacar os pontos fortes e as áreas que precisam de mais atenção, preparando os alunos para revisões adicionais, se necessário.

A aula de revisão e avaliação é fundamental para consolidar o aprendizado dos alunos e avaliar o progresso alcançado ao longo da sequência didática. Ela oferece uma oportunidade valiosa para reforçar conceitos, resolver dúvidas e preparar os alunos para futuros desafios. É um ponto importante destacar que a avaliação não deve ser limitada apenas ao final da sequência didática, mas sim ocorrer ao longo de todo o processo de ensino e aprendizagem. A aplicação de critérios de avaliação em cada aula possibilita um acompanhamento contínuo do progresso dos alunos, permitindo uma intervenção imediata para sanar dúvidas e auxiliar no desenvolvimento das habilidades necessárias.

Ao longo das aulas, podem ser estabelecidos critérios de avaliação que considerem diferentes aspectos, como participação ativa dos alunos nas discussões e atividades, a qualidade da resolução dos exercícios propostos, demonstração de interesse e engajamento no aprendizado, entre outros.

A avaliação contínua proporciona um feedback constante aos alunos, permitindo que eles percebam seu próprio progresso e identifiquem áreas que precisam de maior atenção. Além disso, ao estabelecer critérios claros de avaliação em cada aula, os alunos se sentirão mais motivados e envolvidos no processo de aprendizagem, pois saberão que seu desempenho é valorizado e levado em consideração.

É importante ressaltar que a avaliação não deve ser utilizada apenas como um meio de atribuir notas, mas sim como uma ferramenta pedagógica para auxiliar o desenvolvimento dos alunos. Os resultados da avaliação devem ser utilizados para identificar dificuldades individuais e coletivas, adaptar estratégias de ensino e oferecer suporte adicional, quando necessário.

Dessa forma, a avaliação contínua e criteriosa ao longo da sequência didática proporciona uma visão mais abrangente do progresso dos alunos, permitindo uma intervenção pedagógica mais efetiva e contribuindo para um processo de ensino e aprendizagem mais significativo e enriquecedor.

## 6.10 CONCLUSÃO DA SEQUÊNCIA DIDÁTICA

Com essa sequência didática flexível, porém robusta, os estudantes terão a oportunidade de compreender os conceitos básicos da teoria de funções e otimização, além de explorar métodos não diferenciáveis de otimização. Ao abordar esses métodos, como a busca pela seção áurea, interpolação parabólica, Nelder-Mead e Busca Direta, os alunos poderão desenvolver habilidades de análise e resolução de problemas sem a necessidade de utilizar derivadas.

Ao longo da sequência didática, também é dada ênfase à interpretação gráfica das funções, explorando a representação visual dos dados e a análise dos gráficos para compreender características como crescimento, decrescimento e pontos de intercepção. Isso permite que os alunos desenvolvam uma compreensão intuitiva dos conceitos e fortaleçam sua habilidade de interpretar informações a partir de representações visuais.

Com essa abordagem focada em métodos não diferenciáveis, os estudantes poderão compreender a utilidade desses conceitos no cotidiano. Ao explorar exemplos práticos de otimização, como encontrar a maior área de um retângulo com perímetro fixo, os alunos poderão aplicar os métodos aprendidos para resolver problemas reais de forma eficiente.

Essa sequência didática proporciona uma base sólida para os alunos, preparando-os para explorar conceitos mais avançados e desenvolver uma visão mais ampla da matemática e sua aplicação na vida cotidiana. Ao desenvolver habilidades de resolução de problemas e pensamento crítico por meio de métodos não diferenciáveis, os estudantes estarão preparados para enfrentar desafios matemáticos e aplicar seus conhecimentos em diferentes situações do dia a dia.



## 7 CONCLUSÃO

Durante o percurso dessa dissertação, iniciamos o estudo dos principais métodos de otimização sem derivadas, baseados em modelos simplex (Nelder-Mead e a Busca Multidirecional) e de busca direta direcional. Embora de aplicação simples, os métodos de otimização sem derivadas mostraram-se com uma base teórica sólida e já estabelecida nos estudos da matemática, servindo como alternativa eficiente para os casos onde não é possível realizar a diferenciação de funções. Sua aplicabilidade real já é consolidada e utilizada em situações reais, tais como os exemplos listados no Capítulo 1.

Após a realização da implementação do Método de Nelder-Mead e da Busca Direta Direcional no MATLAB, realizaram-se testes nas funções de Himmelblau, de Baele, de Booth e Three-Hump Camel, que são funções desafiadoras aos métodos, dadas suas formas complexas, múltiplos mínimos locais, e superfícies irregulares. Registramos o número de iterações, o erro e o tempo de máquina de cada uma. Os dois métodos funcionaram de maneira satisfatória, com o segundo deles necessitando de menos iterações para atingir o mínimo desejado, com menor tempo de máquina.

Testes idênticos, partindo de sua implementação, foram realizados com os métodos da Busca Pela Seção Áurea e da Interpolação Parabólica Sucessiva em duas funções. Neles, também obtivemos acerto dos métodos, encontrando o mínimo desejado, com a Interpolação Parabólica Sucessiva necessitando de menos iterações, mas com tempo de máquina maior, em comparação ao outro método.

Observamos também os paralelos entre a Modelagem Matemática, a Resolução de Problemas, a Otimização Matemática e sua presença nas competências e habilidades da Base Nacional Comum Curricular. A Modelagem Matemática e a Resolução de Problemas são metodologias de ensino e pesquisa interdependentes, isto é, estão intrinsecamente ligadas, e complementares, sendo uma ferramenta muito importante na otimização. Ademais, sua utilização no ensino da matemática é de extrema importância e é recomendada pelo documento da BNCC, dadas suas competências e habilidades registradas.

Por fim, propomos uma sequência didática compreendendo alguns dos métodos abordados no escopo desse texto, explorando o ensino de funções, gráficos, otimização, e as potencialidades dos algoritmos de otimização sem derivadas. Essa sequência deverá ser ajustada de acordo com o nível da turma a ser trabalhada, podendo ser facilitada ou dificultada, garantindo maior aprendizagem e aproveitamento dos estudantes.

Acreditamos que os métodos de otimização sem derivadas, apesar de recentes no contexto da matemática, podem desempenhar um papel importante na otimização, em especial, nos problemas em que não conseguimos aplicar os métodos clássicos,

que exigem a diferenciação. Seu desenvolvimento e aprimoramento podem garantir uma ferramenta determinante para a solução de problemas, desenvolvimento de tecnologias, otimização de processos, entre outras aplicações.

## REFERÊNCIAS

- BARBOSA, Jonei Cerqueira. Modelagem na Educação Matemática: contribuições para o debate teórico. **Reunião anual da ANPED**, v. 24, n. 7, p. 1–15, 2001.
- BASSANEZI, Rodney C. **Ensino-aprendizagem com modelagem matemática: uma nova estratégia**. [S.l.]: Editora Contexto, 2002.
- BASSANEZI, Rodney C. Modelagem Matemática. **Dynamis**, v. 1, n. 7, p. 55–83, 1994.
- BOYD, Stephen P; VANDENBERGHE, Lieven. **Convex optimization**. [S.l.]: Cambridge university press, 2004.
- BRASIL. **Base Nacional Comum Curricular**. Brasília, 2018.
- CARVALHO, Ana Márcia Fernandes Tucci de; PIRES, Magna Natália Marin; GOMES, Marilda Trecenti. Fundamentos teóricos do pensamento matemático. IESDE Brasil, 2005.
- CONN, Andrew R; SCHEINBERG, Katya; VICENTE, Luis N. **Introduction to derivative-free optimization**. [S.l.]: SIAM, 2009. P. 277.
- DAVIS, Chandler. Theory of positive linear dependence. **American Journal of Mathematics**, JSTOR, v. 76, n. 4, p. 733–746, 1954.
- DENNIS JR, John E; TORCZON, Virginia. Direct search methods on parallel machines. **SIAM Journal on Optimization**, SIAM, v. 1, n. 4, p. 448–474, 1991.
- DINIZ-EHRHARDT, Maria Aparecida; LOPES, Véra Lucia da Rocha; PEDROSO, Lucas Garcia. Métodos sem derivadas para minimização irrestrita. **Notas em Matemática Aplicada**, SBMAC, v. 49, n. 3, p. 87, 2012.
- GERALD, CF; WHEATLEY, PO. **Applied Numerical Analysis, International ed**. [S.l.]: Pearson/Addison-Wesley, Boston, MA, 2004.
- GIULIANI, Caio Merlini. **Estratégias de otimização não diferenciável aplicadas à maximização da produção de campos de petróleo**. 2013. Diss. (Mestrado) – Universidade Federal de Santa Catarina.

HEATH, MT. Scientific computing: an introductory survey. **New York, NY: McGraw**, 2018.

HIMMELBLAU, David M *et al.* **Applied nonlinear programming**. [S.l.]: McGraw-Hill, 1972.

LAGARIAS, Jeffrey C; REEDS, James A; WRIGHT, Margaret H; WRIGHT, Paul E. Convergence properties of the Nelder–Mead simplex method in low dimensions. **SIAM Journal on optimization**, SIAM, v. 9, n. 1, p. 112–147, 1998.

LESH, Richard; DOERR, Helen M. Foundations of a models and modeling perspective on mathematics teaching, learning, and problem solving. **Beyond constructivism: Models and modeling perspectives on mathematics problem solving, learning, and teaching**, v. 3, p. 33, 2003.

LIMA, Elon Lages. **Espaços Métricos**. 3. ed. Rio de Janeiro: IMPA, 1993. P. 299.

NELDER, John A; MEAD, Roger. A simplex method for function minimization. **The computer journal**, Oxford University Press, v. 7, n. 4, p. 308–313, 1965.

POLYA, George. **A arte de resolver problemas**. 2. ed. Rio de Janeiro: Interciência, 1995. P. 196.

SURJANOVIC, Sonja; BINGHAM, Derek. **Beale Function**. 2013. Disponível em: <https://www.sfu.ca/~ssurjano/beale.html>. Acesso em: 21 mai. 2023.

SURJANOVIC, Sonja; BINGHAM, Derek. **Booth Function**. 2013. Disponível em: <https://www.sfu.ca/~ssurjano/booth.html>. Acesso em: 21 mai. 2023.

SURJANOVIC, Sonja; BINGHAM, Derek. **Three-Hump Camel Function**. 2013. Disponível em: <https://www.sfu.ca/~ssurjano/camel3.html>. Acesso em: 21 mai. 2023.

THIEL, Afrânio A. **Práticas Reflexivas Compartilhadas: abordagens metodológicas no ensino de matemática**. [S.l.]: Quality Copiadora & Gráfica Rápida, 2015.

## APÊNDICE A – CÓDIGOS DOS ALGORITMOS

### A.1 ALGORITMO DE NELDER MEAD APLICADO À FUNÇÃO DE HIMMELBLAU

```
1 % Carregar a Funcao
2 fun = @himmelblau;
3 esperado=0;
4
5 % Definindo o Simplex Inicial
6 x0 = [1, 1]; % Palpite Inicial
7 simplex = [x0; x0 + [1, 0]; x0 + [0, 1]];
8
9 % Definindo a tolerancia
10 tol = 1e-10;
11
12 % Rodando o Algoritmo
13 tic;
14 iter = 0;
15 while true
16     % Classificando os pontos pelo valor da funcao
17     values = zeros(size(simplex, 1), 1);
18     for i = 1:size(simplex, 1)
19         values(i) = fun(simplex(i, :));
20     end
21     [values, order] = sort(values);
22     simplex = simplex(order, :);
23
24     % Calculando o Centroide
25     centroid = mean(simplex(1:end-1, :));
26
27     % Refletindo o pior ponto
28     reflected = 2 * centroid - simplex(end, :);
29
30     % Avaliando o pior ponto
31     reflectedValue = fun(reflected);
32
33     if reflectedValue < values(end-1)
34         % Expandir
35         expanded = 3 * centroid - 2 * simplex(end, :);
36         expandedValue = fun(expanded);
```

```
37
38     if expandedValue < reflectedValue
39         simplex(end, :) = expanded;
40         values(end) = expandedValue;
41     else
42         simplex(end, :) = reflected;
43         values(end) = reflectedValue;
44     end
45 elseif reflectedValue >= values(end-1)
46     % Contrair
47     contracted = 0.5 * (centroid + simplex(end, :));
48     contractedValue = fun(contracted);
49     if contractedValue < values(end)
50         simplex(end, :) = contracted;
51         values(end) = contractedValue;
52     else
53         simplex = 0.5 * (simplex(1, :) + simplex(2:end,
54             :)); % Encolher
55         values = zeros(size(simplex, 1), 1);
56         for i = 1:size(simplex, 1)
57             values(i) = fun(simplex(i, :));
58         end
59     end
60 end
61 % Checando o criterio de tolerancia
62 if max(abs(values - values(1))) <= tol
63     break;
64 end
65
66 iter = iter + 1;
67 end
68 tempo=toc;
69
70 % Mostrar os Resultados
71 minimizer = simplex(1, :);
72 minimum = fun(minimizer);
73 erro = norm(esperado - minimum);
74 disp('Nelder-Mead Algorithm:');
```

```
75 disp(['Valor Minimo: ' num2str(minimum)]);
76 disp(['Minimizador: ' num2str(minimizer)]);
77 disp(['Iteracoes: ' num2str(iter)]);
78 disp(['Erro: ' num2str(erro)]);
79 disp(['Tempo de maquina:' num2str(tempo)]);
```

## A.2 ALGORITMO DA BUSCA DIRETA APLICADO À FUNÇÃO DE HIMMELBLAU

```
83 % Funcao de Himmelblau
84 f = @(x, y) (x^2 + y - 11)^2 + (x + y^2 - 7)^2;
85
86 % Parametros iniciais
87 esperado=0;
88 x0 = 1;
89 y0 = 1;
90 step = 1;
91 minStepSize = 1e-5; % Tamanho minimo do passo
92 maxRuntime = 10; % Tempo de execucao maximo em segundos
93
94 % Inicializacao das variaveis
95 minimizer = [x0, y0];
96 minimum = f(x0, y0);
97 startTime = tic;
98 iteration = 0; % Contador de iteracoes
99
100 % Funcao auxiliar para verificar se um ponto esta dentro da
    base canonica
101 isInsideCanonicalBase = @(point) all(point >= 0);
102
103 % Funcao auxiliar para cortar o tamanho do passo pela metade
104 halfStep = @(stepSize) stepSize / 2;
105
106 % Loop de busca
107 while step >= minStepSize
108     % Verifica o tempo de execucao
109     if toc(startTime) > maxRuntime
110         disp('Tempo de execucao maximo atingido. ');
111         break;
112     end
```

```
113
114     % Avalia os pontos vizinhos na base canonica
115     neighbors = [minimizer(1) + step, minimizer(2);...
116                 minimizer(1) - step, minimizer(2);...
117                 minimizer(1), minimizer(2) + step;...
118                 minimizer(1), minimizer(2) - step];
119
120     % Verifica os pontos vizinhos
121     foundBetterSolution = false;
122     for i = 1:size(neighbors, 1)
123         point = neighbors(i, :);
124         if isInsideCanonicalBase(point)
125             value = f(point(1), point(2));
126
127             % Atualiza o minimo se necessario
128             if value < minimum
129                 minimum = value;
130                 minimizer = point;
131                 foundBetterSolution = true;
132             end
133         end
134     end
135
136     % Verifica se nao encontrou uma solucao melhor
137     if ~foundBetterSolution
138         step = halfStep(step);
139     end
140
141     iteration = iteration + 1; % Incrementa o contador de
142                               iteracoes
143 end
144 erro = norm(esperado - minimum);
145 % Exibe os resultados
146 disp('-----');
147 disp('Resultados:');
148 disp(['Numero de iteracoes: ' num2str(iteration)]);
149 disp(['Minimizador: (' num2str(minimizer(1)) ', ' num2str(
    minimizer(2)) ')']);
150 disp(['Valor minimo: ' num2str(minimum)]);
```



```
150 disp(['Erro: ' num2str(erro)]);
151 disp(['Tempo de execucao: ' num2str(toc(startTime)) '
      segundos']);
```

### A.3 ALGORITMO DA BUSCA PELA SEÇÃO ÁUREA APLICADO À PRIMEIRA FUNÇÃO

```
152 % Definindo a Funcao
153 f = @(x) 2 * exp(-2 * x) - exp(-x);
154
155 % Definindo o intervalo [a, b]
156 a = 1; % limite inferior
157 b = 2; % superior
158
159 % Definindo a Tolerancia
160 error = 1e-5;
161 esperado=-0.1250;
162
163 % Calculo da razao
164 golden_ratio = (sqrt(5) - 1) / 2;
165
166 tic;
167
168 iterations = 0;
169 while abs(b - a) > error
170     % Calculando os pontos
171     x1 = b - golden_ratio * (b - a);
172     x2 = a + golden_ratio * (b - a);
173
174     f1 = f(x1);
175     f2 = f(x2);
176
177     % Atualizando os valores da funcao
178     if f1 < f2
179         b = x2;
180     else
181         a = x1;
182     end
183
```

```
184     iterations = iterations + 1; % Incrementando o contador
185 end
186
187 elapsed_time = toc;
188 diferenca = (esperado - f((a + b) / 2));
189 % Mostrando os resultados
190 disp('Minimizador:');
191 disp((a + b) / 2);
192 disp('Minimo:');
193 disp(f((a + b) / 2));
194 disp('Tempo (s):');
195 disp(elapsed_time);
196 disp('Iteracoes:');
197 disp(iterations);
198 disp('Erro:')
199 disp(diferenca);
```

#### A.4 ALGORITMO DA INTERPOLAÇÃO PARABÓLICA SUCESSIVA APLICADO À PRIMEIRA FUNÇÃO

```
200 % Definindo a Funcao
201 f = @(x) 2 * exp(-2 * x) - exp(-x);
202
203 % Definindo o intervalo [a, b]
204 a = 1; % limite inferior
205 b = 2; % superior
206
207 % Definindo a Tolerancia
208 error = 1e-5;
209 esperado = -0.124990
210
211 tic;
212
213 iterations = 0;
214 while abs(b - a) > error
215     % Avaliando nos tres pontos
216     x1 = a;
217     x2 = (a + b) / 2;
218     x3 = b;
```

```
219
220     f1 = f(x1);
221     f2 = f(x2);
222     f3 = f(x3);
223
224     % Computando os coeficientes
225     c0 = f1;
226     c1 = (f2 - f1) / (x2 - x1);
227     c2 = ((f3 - f1) / (x3 - x1) - c1) / (x3 - x2);
228
229     % Calculando o minimo da parabola
230     x_min = 0.5 * (x1 + x2 - c1 / c2);
231     f_min = f(x_min);
232
233     % Ataulizando os valores de x
234     if f_min < f2
235         if x_min < x2
236             b = x2;
237         else
238             a = x2;
239         end
240     else
241         if x_min < x2
242             a = x_min;
243         else
244             b = x_min;
245         end
246     end
247
248     iterations = iterations + 1; % Incrementando o contador
249 end
250
251 elapsed_time = toc;
252 diferenca = (esperado - f((a + b) / 2));
253 % Mostrando os resultados
254 disp('Minimizador:');
255 disp((a + b) / 2);
256 disp('Minimo:');
257 disp(f((a + b) / 2));
```

```
258 disp('Tempo (s):');
259 disp(elapsed_time);
260 disp('Iteracoes:');
261 disp(iterations);
262 disp('Erro:')
263 disp(diferenca);
```