



UNIVERSIDADE  
ESTADUAL DE LONDRINA

---

GIZELLI RENATA MENDES

MÉTODOS DE OTIMIZAÇÃO APLICADOS EM UMA  
EQUAÇÃO DIFERENCIAL COM MÚLTIPLOS PONTOS DE  
FRONTEIRA

---

Londrina  
2015

GIZELLI RENATA MENDES

**MÉTODOS DE OTIMIZAÇÃO APLICADOS EM UMA  
EQUAÇÃO DIFERENCIAL COM MÚLTIPLOS PONTOS DE  
FRONTEIRA**

Dissertação de mestrado apresentada ao Departamento de Matemática da Universidade Estadual de Londrina, como requisito parcial para a obtenção do Título de MESTRE em Matemática Aplicada e Computacional.

Orientador: Prof. Dr. André Luís Machado  
Martinez

Londrina  
2015

Ficha de identificação da obra elaborada pelo autor, através do Programa de Geração Automática do Sistema de Bibliotecas da UEL

Mendes, Gizelli Renata .

Métodos de otimização aplicados em uma equação diferencial com múltiplos pontos de fronteira / Gizelli Renata Mendes. - Londrina, 2015.  
79 f. : il.

Orientador: André Luís Machado Martinez.

Dissertação (Mestrado em Matemática Aplicada e Computacional) - Universidade Estadual de Londrina, Centro de Ciências Exatas, Programa de Pós-Graduação em Matemática Aplicada e Computacional, 2015.

Inclui bibliografia.

1. Computação - Tese. 2. Matemática Aplicada - Tese. 3. Criptografia - Tese. 4. Curvas elípticas - Tese. I. Martinez, André Luís Machado . II. Universidade Estadual de Londrina. Centro de Ciências Exatas. Programa de Pós-Graduação em Matemática Aplicada e Computacional. III. Título.

GIZELLI RENATA MENDES

**MÉTODOS DE OTIMIZAÇÃO APLICADOS EM UMA EQUAÇÃO  
DIFERENCIAL COM MÚLTIPLOS PONTOS DE FRONTEIRA**

Dissertação de mestrado apresentada ao Departamento de Matemática da Universidade Estadual de Londrina, como requisito parcial para a obtenção do Título de MESTRE em Matemática Aplicada e Computacional.

**BANCA EXAMINADORA**

---

Orientador: Prof. Dr. André Luís Machado  
Martinez  
Universidade Tecnológica Federal do Paraná –  
UTFPR

---

Prof. Dr. Emerson Vitor Castelani  
Universidade Tecnológica Federal do Paraná –  
UTFPR

---

Prof. Dr. Robinson Hoto  
Universidade Estadual de Londrina - UEL

---

Profa. Dra. Glaucia Bressan  
Universidade Tecnológica Federal do Paraná –  
UTFPR

Londrina, 30 de Agosto de 2015.

*Dedico este trabalho ao meu filho João Batista e  
aos meus pais Marilza e Inácio*

## **AGRADECIMENTOS**

Agradeço primeiro lugar a Deus por estar comigo desde o início me sustentando cada dia desta etapa. Também agradeço aos meus pais pelo apoio integral em especial ao meu filho João Batista o qual é a motivação da minha vida. Também agradeço a todos colegas que direta ou indiretamente me ajudaram nesta caminhada em especial ao amigo Pedro Lima. Também agradeço ao meu orientador pela paciência, dedicação que teve no decorrer do curso. A todos professores do PGMAC pela paciência e confiança. Por fim agradeço a CAPES pelo auxílio financeiro. Muito obrigada, que Deus abençoe a todos .

MENDES, Gizelli Renata. **Uso dos métodos de otimização na determinação de solução para uma equação de m-pontos de fronteira**. 2015. 79f. Dissertação (Mestrado em Matemática Aplicada e Computacional) - Universidade Estadual de Londrina, Londrina, 2015.

## RESUMO

Neste trabalho exploramos o estudo de métodos de otimização não linear na determinação de solução numérica para uma equação diferencial de segunda ordem com múltiplos pontos de fronteira, em geral este problema é solucionado utilizando métodos baseados no teorema de ponto fixo de Banach ver [2]. O uso de métodos de otimização não linear mostrou-se vantajoso por permitir uma análise qualitativa dos problemas, além de não depender de que o operador integral seja uma contração na vizinhança da solução. Deste modo apresentamos duas abordagens baseadas em métodos de otimização não linear para o problema na primeira analisamos uma estratégia baseada no método de Gauss-Newton com a equação discretizada, na segunda além da equação discretizada utilizamos como nos métodos baseados no teorema de Banach a equação integral associada a equação diferencial e aplicamos um método de otimização não linear com restrições.

**Palavras-chave:** Equações diferenciais. Otimização não linear.

MENDES, Gizelli Renata. **Using optimization methods for the determination of solution a border m-points of the equation.** 79p. 2015. Dissertation (Masters in Mathematics Applied and Computational) - State University of Londrina, 2015.

### ABSTRACT

In this paper we explore the study of nonlinear optimization methods for the determination of numerical solution to a differential equation of second order with multiple border points, in general this problem is solved using methods based on fixed-point theorem of Banach see [2]. The use of nonlinear optimization methods proved to be advantageous to allow a qualitative analysis of the problems, and does not depend on the integral operator is a contraction in the vicinity of the solution. Therefore we present two approaches based nonlinear optimization methods to the problem in the first analyzed a strategy based on Gauss-Newton method with the discretized equation, the second addition of discretized equation used like in the methods based on the Banach theorem integral equation associated with differential equation and apply a non-linear optimization method with constraints.

**Keywords:** Differential equations. Nonlinear optimization.

## LISTA DE FIGURAS

2.1	O algoritmo 2.16 pode não encontrar um ponto estacionário . . . . .	24
2.2	Convergência do método gradiente . . . . .	28
2.3	Iteração método de Newton . . . . .	30
4.1	Solução exata exemplo 1 . . . . .	56
4.2	Solução exata exemplo 2 . . . . .	58
4.3	Solução exata exemplo 3 . . . . .	59
4.4	Solução exata exemplo 4 . . . . .	60
4.5	Solução exata exemplo 5 . . . . .	61
4.6	Solução exata exemplo 6 . . . . .	61
4.7	Solução exata exemplo 7 . . . . .	62
4.8	Solução exata exemplo 8 . . . . .	63
4.9	Solução exata exemplo 9 . . . . .	64
4.10	Solução exata exemplo 10 . . . . .	64
4.11	Solução exata exemplo 1 . . . . .	65
4.12	Solução exata exemplo 2 . . . . .	66
4.13	Solução exata exemplo 3 . . . . .	67
4.14	Solução exata exemplo 4 . . . . .	67
4.15	Solução exata exemplo 5 . . . . .	68
4.16	Solução exata exemplo 6 . . . . .	68
4.17	Solução exata exemplo 7 . . . . .	69
4.18	Solução exata exemplo 8 . . . . .	70
4.19	Solução exata exemplo 9 . . . . .	70
4.20	Solução exata exemplo 10 . . . . .	71
4.21	Soluções numéricas obtidas. . . . .	73

## LISTA DE TABELAS

2.1	Exemplo 1 . . . . .	34
2.2	Exemplo 2 . . . . .	34
2.3	Exemplo 3 . . . . .	35
2.4	Método Gauss Newton . . . . .	44
2.5	Método Newton para sistema não linear . . . . .	44
4.1	Testes comparativos . . . . .	74

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>12</b>
<b>2</b>	<b>MÉTODOS NUMÉRICOS PARA PROBLEMAS NÃO LINEARES</b>	<b>15</b>
2.1	PROBLEMAS DE MINIMIZAÇÃO IRRESTRITA . . . . .	16
2.1.1	Resultados Preliminares . . . . .	16
2.1.2	Condições de Otimalidade para Problemas Irrestritos . . . . .	18
2.1.3	Existência de solução . . . . .	21
2.1.4	Algoritmos de Otimização . . . . .	22
2.1.5	Cálculo do passo $t_k$ . . . . .	23
2.1.6	Convergência . . . . .	25
2.1.7	Direções Para o Método de Busca Linear . . . . .	27
2.1.8	Testes realizados com os Algoritmos Implementados . . . . .	34
2.2	PROBLEMAS DE MÍNIMOS QUADRADOS . . . . .	36
2.2.1	Método de Gauss Newton . . . . .	38
2.3	EQUAÇÕES NÃO LINEARES . . . . .	41
<b>3</b>	<b>PROBLEMAS COM RESTRIÇÕES</b>	<b>45</b>
3.1	INTRODUÇÃO . . . . .	45
3.2	CONDIÇÕES DE OTIMALIDADE PARA PROBLEMAS RESTRITOS . . . . .	46
3.2.1	Condições Necessária de Primeira Ordem Com Restrições De Igualdade e Desigualdade . . . . .	48
3.2.2	Condição de Qualificação . . . . .	49
<b>4</b>	<b>MÉTODOS DE OTIMIZAÇÃO APLICADOS EM UMA EQUAÇÃO DIFERENCIAL COM MÚLTIPLOS PONTOS DE FRONTEIRA</b>	<b>52</b>
4.1	MÉTODOS NUMÉRICOS . . . . .	52
4.1.1	Algoritmo 1 . . . . .	53
4.1.2	Algoritmo 2 . . . . .	53
4.2	TESTES NUMÉRICOS . . . . .	55
4.2.1	Testes realizados com 1º Algoritmo . . . . .	55
4.2.2	Testes realizados com 2º Algoritmo . . . . .	65
4.2.3	Comparativo . . . . .	72
<b>5</b>	<b>CONCLUSÃO</b>	<b>75</b>
	<b>REFERÊNCIAS</b>	<b>77</b>

## 1 INTRODUÇÃO

Neste trabalho vamos apresentar métodos de otimização não linear para encontrar a solução do problema 1.1. Este problema é conhecido na literatura como problema de segunda ordem com múltiplos pontos de fronteira, ou simplesmente, com m-pontos.

$$\begin{cases} u'' + q(t)f(t, u, u') = 0, \\ u(0) = 0 \quad u(1) = g(u(\eta_1), u(\eta_2), \dots, u(\eta_{m-2})) \end{cases} \quad (1.1)$$

onde  $g : \mathbb{R}^{m-2} \rightarrow \mathbb{R}$ ,  $f : [0, 1] \times \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$  e  $q : \mathbb{R}_+ \rightarrow \mathbb{R}$  são contínuas e  $\eta_1, \dots, \eta_{m-2} \in (0, 1)$ . Esta classe de problema foi proposta pela escola soviética [24] e é aplicado em várias áreas da ciência matemática. De acordo com [27], equações com múltiplos pontos surgem em problemas que modelam fluxos visco elásticos, inelásticos e deformação de vigas. A maioria dos estudos para este problema são em relação a existência de soluções. Os estudos para a existência de solução são variadas. Entre eles, mencionamos os primeiros resultados apresentados por Il'in e Moiseev [25] e [24], onde a equação considerada foi:

$$\begin{cases} u'' + f(t, u, u') = 0, \\ u(0) = 0 \quad u(1) = \sum_{k=1}^{m-2} u(\eta_k) \end{cases} \quad (1.2)$$

Duas das classes do problema 1.2 têm aplicações importantes, além disso muitos autores desenvolvem estudos considerando variações e generalizações do problema 1.2. Além dos estudos de existência também costumam apresentar existência de múltiplas soluções. As técnicas para obter resultados de existência são variadas. Por exemplo, Gupta [15] e [7], obteve as condições de existência usando argumentos de teorias do grau. Ma [22], usou alternativa de Leray-Schauder's, o qual demonstrou existência de solução considerando o seguinte problema com m-pontos de fronteira:

$$\begin{cases} u''(t) = f(t, u, u') + e(t), \\ u'(0) = 0 \quad u(1) = \sum_{k=1}^{m-2} \alpha_k u(\eta_k) \end{cases} \quad (1.3)$$

onde  $f : [0, 1] \times \mathbb{R}^2 \rightarrow \mathbb{R}$  e  $e : [0, 1] \rightarrow \mathbb{R}$  são funções contínuas e  $a_i \in \mathbb{R}$  com os  $a_i$  tendo mesmo sinal. Recentemente, Sun obteve solução positiva usando o teorema Krasnoselskii's para o seguinte problema

$$\begin{cases} u''(t) + a(t)f(u) = 0 \\ u'(0) = 0; \quad u(1) = \sum_{m-2}^{i=2} k_i u(\eta_i) = b \end{cases} \quad (1.4)$$

com as seguintes condições:

- $k_i > 0$  ( $i = 1, \dots, m-2$ ),  $0 < \eta_1 < \eta_2 < \dots < \eta_{m-2} < 1$ ,  $\sum_{i=1}^{m-2} k_i < 1$  e  $\sum_{i=1}^{m-2} k_i \eta_i$ ;
- $a$  e  $f$  são funções contínuas com  $a(t) \neq 0$  para algum  $t \in [0, 1]$ ;

Uma das vantagens de usar teorema de Krasnoselskii's é que podemos explorar aspectos qualitativos da solução como concavidade e positividade.

No artigo [14] foi mostrado a existência de solução para o problema 4.1 usando a alternativa de Leray-Schauder. Neste mesmo artigo foi aplicado o teorema de krasnoselskii's a fim de obter soluções positivas, também foi mostrado existência de solução positiva impondo mais restrições sobre a  $f$ , e outras vezes impondo mais restrições sobre a  $g$ .

. Enquanto existem muitos estudos voltados a existência de soluções, ainda são poucos os estudos em relação a solução numérica e quando estes existem são baseados no teorema do ponto fixo de Banach.

Entre alguns estudos numéricos realizados para encontrar solução do problema 1.1 podemos citar o método iterativo [10], onde ele considera que a solução  $u$  do problema 1.1 é dada pela equação:

$$u(x) = \int_0^x t(1-x)q(t)f(t, u(t), u'(t))dt + \int_x^1 x(1-t)q(t)f(t, u(t), u'(t))dt + g(u(\eta_1), \dots, u(\eta_{m-2}))x$$

Assim o método iterativo considera o próximo iterando dado por:

$$u^{k+1}(x) = \int_0^x t(1-x)q(t)f(t, u^k(t), u^{k'}(t))dt + \int_x^1 x(1-t)q(t)f(t, u^k(t), u^{k'}(t))dt + g(u(\eta_1), \dots, u(\eta_{m-2}))x$$

Outro método numérico que podemos citar é o método de tiros, também apresentado em [10]. Este método consiste em transformamos o problema 4.1 com fronteira em um problema de valor inicial onde o vetor velocidade inicial deste novo problema é determinado de maneira que a solução dos dois problemas se coincidam. Este método consiste em resolver o seguinte problema de valor inicial:

$$\begin{cases} u'' = f(x, u, u') = 0, & 0 < x < 1 \\ u(0) = 0 \\ u'(0) = v \end{cases} \quad (1.5)$$

O objetivo principal deste trabalho foi apresentar duas alternativas que resolve o problema 1.1 usando métodos de otimização não linear. A primeira alternativa se resume em discretizarmos o problema 1.1 e aplicarmos o método de Gauss Newton combinado com spline. A segunda alternativa leva em consideração que o problema 1.1 pode ser modelado como uma equação integral e devido a este fato podemos aplicar o teorema do ponto fixo de Banach. Além desta informação consideramos o sistema não linear  $r(u) = 0$  discretizado como restrição de igualdade. Assim o novo problema resume em resolvermos o seguinte problema:

$$\begin{aligned} \min \quad & \|T(u) - u\|_2^2 \\ \text{sujeito a} \quad & \frac{u_{j+1} - 2u_j + u_{j-1}}{h^2} + f\left(t_j, u_j, \frac{u_{j+1} - u_{j-1}}{2h}\right) = 0 \\ & u_1 - g(u(\eta_1), u(\eta_2), \dots, u(\eta_{m-2})) = 0 \\ & Q(u) = 0 \end{aligned}$$

A motivação para definirmos os Algoritmo 1 e 2 consiste no fato de que métodos numéricos baseados em ponto fixo não permitem a inclusão de análise qualitativa da solução, além disso estes métodos apresentam forte tendência de convergir para a solução de menor norma, quando o problema possui múltiplas soluções, mesmo utilizando variedade de aproximações iniciais.

O nosso trabalho esta dividido da seguinte forma: O segundo capítulo iniciamos fazendo uma introdução a problemas de otimização não linear. Neste capítulo apresentamos três problemas não lineares, são eles: minimização de uma função, solução para um sistema não linear e solução para problemas de mínimos quadrados. Para cada problema apresentamos métodos numéricos, implementamos estes métodos e testamos alguns exemplos.

No terceiro capítulo apresentamos conceitos de otimização restrita, este capítulo se dedica em resolver um problema com restrições de igualdade e desigualdade. Damos ênfase às equações de KKT os quais são condições necessárias de otimalidade para problemas restritos. Os exemplos que apresentamos foram resolvidos de maneira analítica.

No quarto capítulo apresentamos o objetivo principal do nosso trabalho e concluímos este capítulo efetuando testes em dez exemplares a fim de validarmos e compararmos as duas alternativas que resolveram o problema 1.1.

## 2 MÉTODOS NUMÉRICOS PARA PROBLEMAS NÃO LINEARES

Neste capítulo apresentamos algoritmos que resolvem três importantes problemas não lineares, são eles:

- i) Resolver um problema de minimização irrestrita ;
- ii) Resolver problemas de mínimos quadrados não lineares e
- iii) Resolver um sistema de equações não lineares.

Na primeira seção apresentamos métodos que resolva problemas do tipo 2.1 ou seja, maximizar, minimizar uma função sobre um conjunto  $\Omega$ . Este tipo de problema surge em diversas áreas como na matemática, engenharia, química entre outras.

Representamos um problema de otimização da seguinte maneira:

$$\begin{aligned} \min \quad & f(x) \\ \text{sujeito a} \quad & x \in \Omega \end{aligned} \quad (2.1)$$

onde  $f$  é a função objetivo,  $x$  é o vetor de variáveis, e  $\Omega$  é o subconjunto do  $\mathbb{R}^n$ . Quando  $\Omega = \mathbb{R}^n$ , dizemos que o problema é irrestrito.

Na segunda seção apresentamos métodos que resolva problemas de mínimos quadrados não lineares. Este tipo de problema surge em diversas aplicações, principalmente em ajustes de curvas, mas pode ocorrer no caso mais geral : em um sistema de equações não lineares com mais equações do que incógnitas. Para resolver problemas de quadrados mínimos não lineares, usaremos o método de Gauss Newton. Este problema pode ser escrito da seguinte forma

$$\min \quad f(x) = \frac{1}{2} \sum_{j=1}^m r_j^2(x)$$

Por último apresentamos métodos que resolva um sistema de equações não lineares com  $n$  incógnitas e  $n$  equações. Primeiramente, temos por objetivo resolver o seguinte problema:

$$\text{Encontrar } x_* \in \mathbb{R}^n \quad \text{tal que} \quad F(x_*) = 0 \quad (2.2)$$

onde  $F : \mathbb{R}^n \rightarrow \mathbb{R}^m$  é contínua e diferenciável. Para resolver este tipo de problema foram apresentados dois métodos: o método de Newton onde construímos um modelo linear para  $F$  nas proximidades de um ponto  $x_0$  e o método de Broyden.

## 2.1 PROBLEMAS DE MINIMIZAÇÃO IRRESTRITA

### 2.1.1 Resultados Preliminares

**Teorema 2.1** (Teorema de Taylor). *Suponha que  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  é continuamente diferenciável e que  $p \in \mathbb{R}^n$ . Então existe  $t \in (0, 1)$  tal que*

$$f(x + p) = f(x) + \nabla f(x + tp)^T p \quad (2.3)$$

Entretanto se  $f$  é duas vezes continuamente diferenciável, temos que

$$\nabla f(x + p) = \nabla f(x) + \int_0^1 \nabla^2 f(x + tp) p dt \quad (2.4)$$

e

$$f(x + p) = f(x) + \nabla f(x)^T p + \frac{1}{2} p^T \nabla^2 f(x + tp) p \quad (2.5)$$

para algum  $t \in (0, 1)$

**Definição 2.2.** *Uma função  $H : \mathbb{R}^n \rightarrow \mathbb{R}$  é dita de forma quadrática se  $H$  aplicado em  $v = (v_1, \dots, v_n) \in \mathbb{R}^n$  indicado por  $Hv^2$  é dada por*

$$Hv^2 = \sum_{i,j=1}^n h_{ij} v_i v_j$$

onde  $h_{ij}$  é uma matriz  $m \times n$  simétrica  $h_{ji} = h_{ij}$   $i, j = 1, \dots, n$ .

- a) Uma forma quadrática é dita positiva se  $Hv^2 > 0 \forall v \in \mathbb{R}^n \ v \neq 0$ ;
- b) Uma forma quadrática é dita negativa se  $Hv^2 < 0 \forall v \in \mathbb{R}^2/0$ ;
- c) Uma forma quadrática é dita indefinida se existem  $v$  e  $w \in \mathbb{R}^n/0$  tais que  $Hv^2 > 0$  e  $Hw^2 < 0$ .

**Definição 2.3.** *Considere uma função vetorial  $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$ . Sua derivada chamada de Jacobiana, é uma matriz  $m \times n$  escrita da seguinte maneira*

$$J(x) = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \dots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \dots & \frac{\partial f_m}{\partial x_n} \end{pmatrix}_{n \times m} \quad (2.6)$$

**Definição 2.4.** *Seja  $f : U \subset \mathbb{R}^n \rightarrow \mathbb{R}$  duas vezes diferenciável em  $a$ . A matriz*

$$\left( \frac{\partial^2 f(a)}{\partial x_i \partial x_j} \right)_{i,j=1,\dots,n}$$

é chamada matriz Hessiana de  $f$  em  $a$  onde  $i$  é a linha e  $j$  é a coluna.

**Definição 2.5.** Seja  $f : U \subset \mathbb{R}^n \rightarrow \mathbb{R}$  duas vezes diferenciável em  $a$ . O ponto crítico  $a$  diz-se não degenerado quando a matriz neste ponto é invertível isto é

$$\det \left( \frac{\partial^2 f(a)}{\partial x_i \partial x_j} \right)_{i,j=1,\dots,n} \neq 0$$

**Definição 2.6.** Seja  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  diferenciável. Um ponto  $x^* \in U$  é dito ponto crítico de  $f$  se  $df(x^*) = 0$  isto é

$$\nabla f(x^*) = df(x^*) = \left( \frac{\partial f(x^*)}{\partial x_1}, \frac{\partial f(x^*)}{\partial x_2}, \dots, \frac{\partial f(x^*)}{\partial x_n} \right) = (0, 0 \dots 0)$$

**Definição 2.7.** Sejam  $U \subset \mathbb{R}^m, V \subset \mathbb{R}^n$  abertos e  $f : U \rightarrow \mathbb{R}^n, g : V \rightarrow \mathbb{R}^p$  diferenciáveis nos pontos  $a \in U, b = f(a) \in V, \text{com } f(U) \subset V$ . Então  $g \circ f : U \rightarrow \mathbb{R}^p$  é diferenciável no ponto  $a$  e

$$(g \circ f)'(a) = g'(b) \cdot f'(a) : \mathbb{R}^m \rightarrow \mathbb{R}^p$$

Outras notações:

$$\frac{\partial g_i}{\partial x_j} = \sum_{k=1}^n \frac{\partial g_i}{\partial y_k} \frac{\partial y_k}{\partial x_j}$$

## Diferenças Finitas

O método por diferenças finitas baseia no princípio da série de Taylor em que um ponto qualquer pode ser expresso em função dos seus pontos próximos ou seja:

$$f(x+h) = f(x) + \frac{df(x)}{dx}h + \frac{d^2f(x)}{dx^2} \frac{h^2}{2!} + \frac{d^3f(x)}{dx^3} \frac{h^3}{3!} + \dots \quad (2.7)$$

Vamos considerar  $h$  a distância entre  $f(x_i)$  e  $f(x_{i+1})$  e usar a série de Taylor para escrever uma aproximação para a função no ponto  $f(x_{i+1})$ . Analogamente escrevemos a aproximação para  $f(x_{i-1})$ .

$$f(x_{i+1}) = f(x_i) + \frac{\partial f(x_i)}{\partial x_i}h + \frac{d^2f(x_i)}{dx^2} \frac{h^2}{2!} + \frac{d^3f(x_i)}{dx^3} \frac{h^3}{3!} + \dots \quad (2.8)$$

$$f(x_{i-1}) = f(x_i) - \frac{\partial f(x_i)}{\partial x_i}h + \frac{d^2f(x_i)}{dx^2} \frac{h^2}{2!} + \frac{d^3f(x_i)}{dx^3} \frac{h^3}{3!} + \dots \quad (2.9)$$

No caso das aproximações acima se utilizarmos até o termo de primeira ordem, cada uma pode resultar em aproximações para a primeira derivada ou seja,

$$\frac{\partial f(x_i)}{\partial x_i} = \frac{f(x_i) - f(x_{i-1})}{h} \quad (2.10)$$

A fórmula acima é conhecida como aproximação para a derivada avançada e a fórmula abaixo é conhecida como aproximação para a derivada recuada.

$$\frac{\partial f(x_i)}{\partial x_i} = \frac{f(x_{i+1}) - f(x_i)}{h} \quad (2.11)$$

Estas duas expressões são aproximações de primeira ordem, a expressão para a primeira derivada considerando os termos de até segunda ordem pode ser obtida fazendo a subtração entre as equações 2.8 e 2.9 assim obtemos.

$$f'(x) = \frac{f(x_{i+1}) - f(x_{i-1}))}{2h} \quad (2.12)$$

Como estamos trabalhando com os termos de até segunda ordem podemos obter também uma expressão para a segunda derivada. Isto é possível fazendo a soma das equações 2.8 e 2.9 obtendo

$$f(x_{i+1}) + f(x_{i-1}) = 2f(x_i) + 2f''(x) \frac{h^2}{2} \quad (2.13)$$

Rearranjando na forma desejada resulta em

$$f''(x) = \frac{f(x_{i+1}) - 2f(x_i) + f(x_{i-1}))}{h^2} \quad (2.14)$$

Esta seção foi baseada em [4].

## 2.1.2 Condições de Otimalidade para Problemas Irrestritos

**Teorema 2.8.** (Condições necessárias de 1º ordem). *Seja  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  diferenciável no ponto  $a \in \mathbb{R}^n$ . Se  $a$  é um mínimo local de  $f$ , então  $a$  é ponto crítico de  $f$ . Ou seja:*

$$\nabla f(a) = 0 \quad (2.15)$$

### Demonstração

Considere  $d \in \mathbb{R}^n \setminus \{0\}$  arbitrário. Como  $f$  tem mínimo local em  $a$  temos que existe um  $\delta > 0$  tal que

$$f(a) \leq f(a + te_i)$$

para todo  $\|t\| < \delta$ . Para cada  $i \in 1, \dots, n$  defina  $\varphi_i$  e  $\lambda(t)$  respectivamente como

$$\begin{aligned} \varphi_i : \mathbb{R} &\rightarrow \mathbb{R} \\ \varphi_i &\mapsto f(a + te_i) \end{aligned}$$

$$\begin{aligned} \lambda : \mathbb{R} &\rightarrow \mathbb{R}^n \\ \lambda &\mapsto a + te_i \end{aligned}$$

Note, que para todo  $\|t\| < \delta$  temos que

$$\varphi_i = (f \circ \lambda)(t)$$

Observe que

$$a + te_i = (a_1, a_2, \dots, a_i, \dots, a_n) + t(0, 0, 0, 1, \dots, 0, 0 \dots 0) = (a_1, a_2, a_3, \dots, a_i + t, \dots, a_n)$$

Note que  $\varphi$  é diferenciável, pois  $f$  é diferenciável por hipótese e  $\lambda$  é diferenciável, e a derivada de  $\varphi$  é dada por

$$\begin{aligned} \varphi'_i(t) &= (f \circ \lambda)' = \sum_{j=1}^n \frac{\partial f}{\partial x_j}(\lambda(t)) * \lambda'_j(t) \\ &= \frac{\partial}{\partial x_1} f(a + te_i) \lambda'_1(t) + \frac{\partial}{\partial x_2} f(a + te_i) \lambda'_2(t) + \dots + \frac{\partial}{\partial x_i} f(a + te_i) \lambda'_i(t) + \dots + \frac{\partial}{\partial x_n} f(a + te_i) \lambda'_n(t) \\ &= \frac{\partial}{\partial x_1} f(a + te_i) \frac{d}{dt} a_1 + \frac{\partial}{\partial x_2} f(a + te_i) \frac{d}{dt} a_2 + \dots + \frac{\partial}{\partial x_i} f(a + te_i) \frac{d}{dt} (a_i + t) \dots + \frac{\partial}{\partial x_n} f(a + te_i) \frac{d}{dt} a_n \end{aligned}$$

Por outro lado temos que

$$\varphi_i(t) = f(a + te_i) \geq f(a) = \varphi_i(0)$$

$$t \in (0, \delta)$$

Logo zero é um ponto de mínimo local para a função  $\varphi$ . Como a função  $\varphi$  esta definida de  $\mathbb{R}$  em  $\mathbb{R}$  temos que

$$\varphi'(0) = \frac{\partial f(a)}{\partial x_i} = 0$$

para todo  $i = \{1, \dots, n\}$ . Concluindo assim que

$$\nabla f(a) = 0$$

**Teorema 2.9.** (Condições necessárias de 2º ordem) Se  $x_*$  é mínimo local de  $f$  e  $\nabla^2 f$  existe e é contínua numa vizinhança de  $x_*$ , então  $\nabla f(x_*) = 0$  e  $\nabla^2 f(x_*)$  é semi definitiva positiva, ou seja  $h^T \nabla^2 f(x_*) h \geq 0$ .

Ver demonstração [17] página 16.

**Teorema 2.10.** (Condições Suficientes de 2º Ordem) Seja  $f : U \subset \mathbb{R}^n \rightarrow \mathbb{R}$  de classe  $C^2$   $x_* \in U$  ponto crítico de  $f$  e  $H = \nabla^2 f(x_*)$  (matriz hessiana). Assim,

- i) Se  $H$  é definida positiva então  $x_*$  é ponto de mínimo local não degenerado;
- ii) Se  $H$  é definida negativa então  $x_*$  é ponto de máximo local não degenerado;
- iii) Se  $H$  é indefinida então  $x_*$  é dito ponto de sela, não é nem de máximo nem de mínimo.

### Demonstração

- i) Seja  $B(x_*, \delta) \subset U$ . Pela fórmula de Taylor com resto infinitesimal tem-se  $\forall v \in \mathbb{R}^n, x_* + v \in U$ , vale

$$f(x_* + v) = f(x_*) + \nabla f(x_*)v + \frac{1}{2}\nabla^2 f(x_*)v^2 + r(v)$$

onde

$$\lim_{v \rightarrow 0} \frac{r(v)}{\|v\|^2} = 0$$

Mas  $x_*$  é ponto crítico  $\nabla f(x_*)v = 0$  e  $H = \nabla^2 f(x_*)$  matriz hessiana logo

$$f(x_* + v) = f(x_*) + \frac{1}{2}Hv^2 + r(v) = f(x_*) + \left[ \frac{1}{2}H \left( \frac{v}{\|v\|} \right)^2 + \frac{r(v)}{\|v\|^2} \right] \|v\|^2$$

A função contínua  $\frac{1}{2}H$  sendo positiva em todos os pontos da esfera unitária de  $\mathbb{R}^n$ , a qual é um conjunto compacto então existe  $\epsilon > 0$  tal que  $\frac{1}{2}Hu^2 > \epsilon$  para todo  $u \in \mathbb{R}^n$  de norma 1. Então  $\frac{1}{2}H \left( \frac{v}{\|v\|} \right)^2 > \epsilon$  para todo  $v \neq 0$  em  $\mathbb{R}^n$ . Ora, pela fórmula de Taylor infinitesimal  $\exists \delta > 0$  tal que  $0 < \|v\| < \delta \Rightarrow \frac{|r(v)|}{\|v\|^2} < \frac{\epsilon}{2} \Rightarrow \frac{-\epsilon}{2} < \frac{r(v)}{\|v\|^2} < \frac{\epsilon}{2}$ . Logo obtemos,

$$f(x_* + v) - f(x_*) = \frac{1}{2}H \left( \frac{v}{\|v\|} \right)^2 + \frac{r(v)}{\|v\|^2} \geq \left( \epsilon - \frac{\epsilon}{2} \right) \|v\|^2 > 0$$

Concluindo assim que  $f(x_* + v) > f(x_*) \forall v \in \mathbb{R}^n, x_* + v \in U, v \neq 0, \|v\| < \delta$ . Ou seja  $x_*$  é ponto de mínimo.

- ii) A demonstração é semelhante a anterior.
- iii) Se  $H$  é indefinida, existem vetores  $v, w \in \mathbb{R}^n$  tais que  $Hv^2 > 0$  e  $Hw^2 < 0$ . Então para todo  $t \neq 0$  real, temos  $H(tv)^2 = t^2Hv^2 > 0$  e  $H(tw)^2 < 0$ . A igualdade acima implica então que  $f(x_* + tv) > f(x_*)$  e  $f(x_* + tw) < f(x_*)$  para todo  $t \neq 0$  suficientemente pequeno, logo  $x_*$  não é ponto de mínimo local nem de máximo local para  $f$ .

Estes resultados são as bases para os algoritmos de otimização irrestrita. Esta seção foi baseada em [9].

### 2.1.3 Existência de solução

Antes de encontrarmos a solução do problema 2.1 é importante verificarmos a existência de solução, assim evitamos procurar por soluções que não existem. Podemos encontrar em [1] as demonstrações dos teoremas sobre existência.

**Teorema 2.11. (Weierstrass)** *Sejam  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  contínua e  $\Omega \subset \mathbb{R}^n$  compacto não vazio. Então  $f$  atinge seu valor de máximo e mínimo em pontos de  $\Omega$ . Ou seja, existem  $x_0, x_1 \in \Omega$  tais que  $f(x_0) \leq f(x) \leq f(x_1)$ .*

**demonstração** Ver teorema 2.2, página 27, em [1]

**Definição 2.12.** *A função  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  é coerciva quando  $\lim_{\|x\| \rightarrow \infty} f(x) = \infty$ , ou seja, quando para todo  $M > 0$ , existe  $r > 0$  tal que  $f(x) > M$  sempre que  $\|x\| > r$ .*

**Teorema 2.13.** *Seja  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  uma função contínua e coerciva. Então,  $f$  tem um minimizador global.*

**demonstração** Ver teorema 2.6, página 28, em [1]

**Teorema 2.14.** *Seja  $f$  uma função convexa definida sobre um conjunto convexo  $\Omega$ . Então o conjunto onde a  $f$  atinge seu mínimo é convexo e qualquer mínimo relativo de  $f$  é um mínimo global.*

**demonstração** Ver teorema 1, página 197, em [8]

Para o problema 2.1 temos dois tipos de soluções: O minimizador global e o minimizador local.

**Definição 2.15.** *Considere uma função  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  e  $x^* \in \Omega \subset \mathbb{R}^n$ . Dizemos que  $x^*$  é um minimizador local de  $f$  em  $\Omega$  quando existe  $\delta > 0$ , tal que  $f(x^*) \leq f(x)$ , para todo  $x \in B(x^*, \delta) \cap \Omega$ . Caso  $f(x^*) \leq f(x)$ , para todo  $x \in \Omega$ ,  $x^*$  é dito minimizador global de  $f$  em  $\Omega$ .*

O melhor ponto que podemos encontrar como solução é o mínimo global, mas isto pode ser difícil pois o conhecimento da  $f$  geralmente é local [17] (páginas 12 e 14). De acordo com a definição 2.15 temos que a maneira de determinar se um ponto  $x^*$  é um minimizador local é examinar todos os pontos na sua vizinhança e verificar se nenhum dele possui valor de função menor. No entanto, quando a função é suave ou seja, duas vezes diferenciável, podemos dizer que  $x^*$  é um mínimo local examinando apenas o gradiente  $\nabla f(x^*)$  e a hessiana  $\nabla^2 f(x^*)$ . As condições necessárias e suficientes que o gradiente e a hessiana devem satisfazer para caracterizar um minimizador são conhecidas como condições de otimalidade.

### 2.1.4 Algoritmos de Otimização

As soluções para problemas de otimização não linear geralmente são encontradas através de processos iterativos. Ou seja, a partir de um ponto inicial  $x_0$ , obtemos um ponto melhor  $x_{k+1}$  dado por  $x_{k+1} = x_0 + t_k d_k$  onde  $x_k \in \mathbb{R}^p$ ,  $t_k \in \mathbb{R}^+$  define o tamanho do passo na direção  $d_k \in \mathbb{R}^p$ . Para decidir como mover do ponto  $x_0$  para o ponto  $x_{k+1}$  é utilizada duas estratégias: busca linear e regiões de confiança. Nós vamos apresentar neste trabalho apenas a busca linear. Na busca linear o algoritmo escolhe uma direção  $d_k$  de descida e busca ao longo desta direção, a partir de  $x_0$ , um ponto  $x_{k+1}$  onde a função decresce. A distância a se mover ( $t_k$ ) ao longo de  $d_k$  será calculada de forma a satisfazer a condição de Armijo ou através do algoritmo *backtraking*. Os algoritmos de busca linear geralmente se diferenciam pela direção  $d_k$ .

Um dos critérios de parada para o algoritmo é parar quando  $\nabla f(x^k)$  for suficientemente pequeno ou seja quando  $\nabla f(x^k)$  tender a zero [17].

Todo bom algoritmo de otimização possui as seguintes características [17]:

- vii) Robustez: Significa que o algoritmo funciona bem em uma grande variedade de problemas;
- viii) Eficiência: não deve exigir muito tempo computacional ou espaço em memória para ser executado e;
- ix) Precisão: deve ser capaz de identificar uma solução com precisão, sem ser muito sensível a erros nos dados ou a erros de arredondamento.

#### **algoritmo 2.16.** Algoritmo básico

Passo 1 Dado  $x_0 \in \mathbb{R}^n$

Passo 2 Repita enquanto  $\nabla f(x_k) \neq 0$ ;

Passo 3 Calcule  $d_k$  tal que  $\nabla f(x_k)^T d^k < 0$ ;

Passo 4 Escolha  $t_k > 0$  tal que  $f(x_k + t_k d_k) < f(x_k)$

Passo 5 faça:  $x_{k+1} = x_k + t_k d_k$ ;

Passo 6  $k = k + 1$

### 2.1.5 Cálculo do passo $t_k$

#### Condições de Armijo e de Wolfe

No cálculo de  $t_k$  procuramos um passo que reduz suficientemente a função

$$\min_{t_k > 0} f(x_k + t_k d_k) \quad (2.16)$$

mas ao mesmo tempo não queremos gastar muito tempo fazendo esta escolha. Encontrar o mínimo da função 2.16 geralmente requer muitas avaliações da função objetivo e possivelmente do gradiente da  $f$ . Na prática a estratégia é fazer uma busca linear inexata para encontrar  $t_k$  que atinge reduções adequadas para  $f$  a um custo mínimo. Tipicamente algoritmos de busca linear testam uma sequência de candidatos para valores de  $t_k$ , parando quando certas condições são satisfeitas.

A simples condição imposta sobre  $t_k$  que requer a redução da  $f$  isto é

$$f(x_k + t_k d_k) < f(x_k)$$

, não garanti a convergência para o mínimo  $x^*$ . De fato, considere  $f : \mathbb{R} \rightarrow \mathbb{R}$  dada por  $f(x) = x^2$  e as sequências  $x^k = 1 + \frac{1}{k+1}$  e  $y^k = (-1)^k + \frac{(-1)^k}{k+1}$ . Ambas as sequências podem ser produzidas pelo algoritmo 2.16 mas nenhuma converge para zero.

A insuficiência redução da  $f$  em cada passo causa uma falha na convergência para o mínimo da função convexa. Para evitarmos este comportamento necessitamos aplicar uma condição suficiente de decrescimento, as condições apresentadas neste trabalho são as condições de Armijo e de Wolfe.

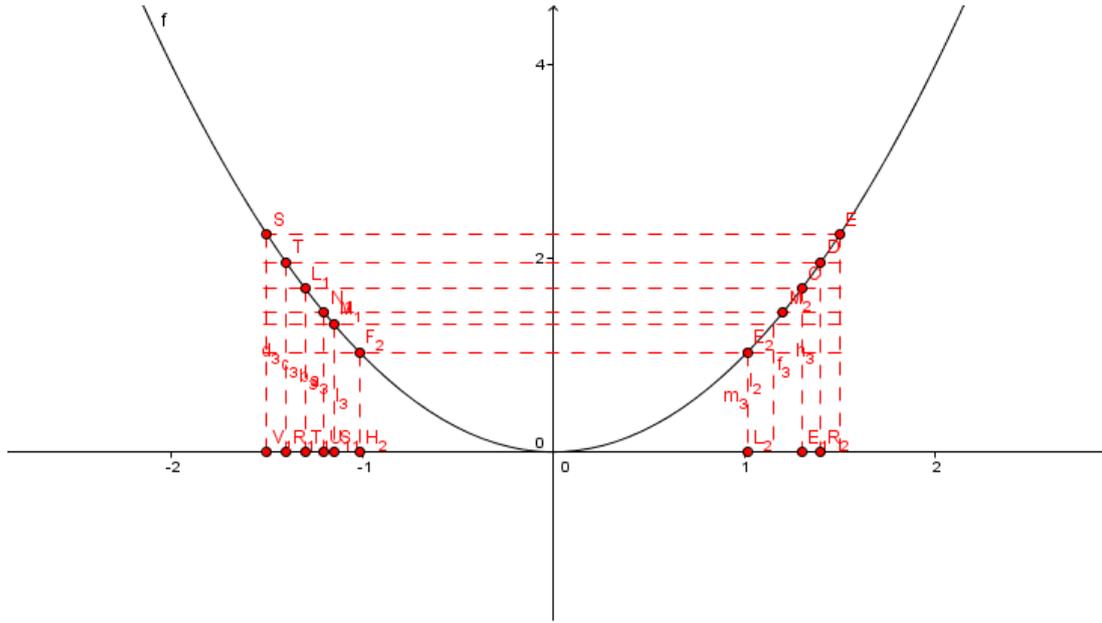


Figura 2.1: O algoritmo 2.16 pode não encontrar um ponto estacionário

Uma popular pesquisa de busca linear inexata estipula que  $t_k$  deve antes de tudo estipular um decrescimento da função objetivo  $f$  como medido pela desigualdade.

$$f(x^* + t_k d_k) \leq f(x^*) + c_1 t_k \nabla f(x^*)^T d_k \tag{2.17}$$

para uma constante  $c_1 \in (0, 1)$ . Em outras palavras, a redução da  $f$  deve ser proporcional ao comprimento de passo  $t_k$  e a derivada direcional  $\nabla f(x_k)^T d_k$ . Esta primeira desigualdade é conhecida como condição de Armijo. Seja  $l(t_k) = f(x_k) + c_1 t_k \nabla f(x_k)^T d_k$ , então a condição de Armijo pode ser expressa como  $f(x_k + t_k d_k) \leq l(t_k)$ . Note que a função  $l(t_k)$  é uma função decrescente e linear em  $t_k$ .

A condição de Armijo não é suficiente para garantir que o algoritmo tenha um progresso, pois esta condição é satisfeita para todos valores de  $t_k$  suficientemente pequeno.

Para eliminar tamanhos de passos inaceitavelmente pequenos, introduzimos uma segunda condição, chamada condição de curvatura, o qual requer que  $t_k$  satisfaça :

$$\nabla f(x_k + t_k d_k)^T \geq c_2 \nabla f(x_k)^T d_k$$

para uma constante  $c_2 \in (c_1, 1)$ , onde  $c_1$  é a constante de 2.17.

A condição de Armijo juntamente com a condição de curvatura são conhecidas como condições de Wolfe 2.18.

$$\begin{aligned} f(x_k + t_k d_k) &\leq f(x_k) + c_1 t_k \nabla f(x_k)^T d_k \\ \nabla f(x_k + t_k d_k)^T &\geq c_2 \nabla f(x_k)^T d_k \end{aligned} \quad (2.18)$$

### Redução Suficiente e Backtracking

Utilizando-se o chamado método de retrocesso, podemos encontrar um passo  $t_k$  que satisfaz as condições de Wolfe 2.18 sem considerar a segunda condição.

**algoritmo 2.17.** Retrocesso

Passo 1 Inicializar com  $i \leftarrow 0$ ,  $t_0, c \in (0, 1)$  e  $\rho > 0$ ;

Passo 2 Enquanto  $f(x_k + t_i d_k) > f(x_k) + c t_i \nabla f(x_k)^T d_k$

$$t_{i+1} = \rho t_i;$$

Passo 3  $i \leftarrow i + 1$

Usualmente  $t = 1, \rho = 0.5$  e  $c = 10^{-4}$

Para o método de Newton e Quase-Newton o passo inicial escolhido será 1. O tamanho de passo  $t_k$  será aceito somente quando a desigualdade do passo 2 for satisfeita. Na prática a cada iteração o fator de contração  $\rho$  pode variar, a única exigência é garantir que  $\rho \in [\rho_1, \rho_2]$  para constantes fixadas  $0 < \rho_1 < \rho_2 < 1$ .

### 2.1.6 Convergência

Dado um processo iterativo que produz uma sequência de pontos  $(x_k)$  a partir de um ponto inicial  $x_0$ , nós estamos interessados em determinar se esta sequência converge para a solução  $x_*$  e ainda quão rapidamente. Se nós assumirmos que esta converge, então as seguintes definições caracterizam as propriedades que serão necessárias.

**Definição 2.18.** *Seja  $x_* \in \mathbb{R}^n$ , e  $x_k \in \mathbb{R}^n, k = 0, 1, 2, \dots$ . Então a sequência  $x_k = x_0, x_1, x_2, \dots$  é dita convergente a  $x_*$  se*

$$\lim_{k \rightarrow \infty} \|x_k - x_*\| = 0$$

**Definição 2.19.** *Dizemos que a sequência  $(x_k) \subset \mathbb{R}^n$  converge linearmente para  $x_* \in \mathbb{R}^n$ , com razão de convergência  $r \in [0, 1)$ , quando*

$$\frac{\|x_{k+1} - x_*\|}{\|x_k - x_*\|} \leq r$$

para todo  $k$  suficientemente grande.

**Definição 2.20.** A sequência  $(x_k) \subset \mathbb{R}^n$  converge superlinearmente para  $x^* \in \mathbb{R}^n$  quando

$$\frac{\|x^{k+1} - x^*\|}{\|x^k - x^*\|} \rightarrow 0$$

**Definição 2.21.** A sequência  $(x_k) \subset \mathbb{R}^n$  converge quadraticamente para  $x^* \in \mathbb{R}^n$  quando  $x^k \rightarrow x^*$  e existe uma constante  $M > 0$  tal que

$$\frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|^2} \leq M$$

para todo  $k$  suficientemente grande.  $M$  é uma constante positiva, não necessariamente menor do que 1.

## 2.1.7 Direções Para o Método de Busca Linear

### Direções de Descida

Vimos anteriormente como encontrar o tamanho do passo  $t_k$  para caminharmos em uma certa direção  $d_k$ . Agora estudaremos sobre esta direção  $d_k$ .

Os algoritmos de pesquisa linear requer que  $d_k$  seja uma direção de descida afim de garantir que a função  $f$  seja reduzida ao longo desta direção. Geralmente o vetor direção de descida  $d_k$  tem a forma

$$d_k = -B_k^{-1}\nabla f(x_k), \quad (2.19)$$

onde  $B_k$  é uma matriz simétrica e não singular (determinante diferente de zero). No método do gradiente,  $B_k$  é simplesmente a matriz identidade, o método de Newton,  $B_k$  é a matriz hessiana  $\nabla^2 f(x_k)$ , no método de quase Newton,  $B_k$  é aproximação da Hessiana que é atualizada a cada iteração por meio de uma fórmula.

**Definição 2.22.** *Considere uma função  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  um ponto  $x^* \in \mathbb{R}^n$  e uma direção  $d \in \mathbb{R}^n \setminus \{0\}$ . Dizemos que  $d$  é uma direção de descida para  $f$ , a partir de  $x^*$ , quando existe  $\delta > 0$  tal que  $f(x^* + td) < f(x^*)$  para todo  $t \in (0, \delta)$ .*

**Teorema 2.23.** *Se  $\nabla f(x^*)^T d < 0$ , então  $d$  é uma direção de descida para  $f$ , a partir de  $x^*$ .*

**Demonstração** Sabemos que

$$\nabla f(x^*)^T d = \frac{\partial f}{\partial d}(x^*) = \lim_{t \rightarrow 0} \frac{f(x^* + td) - f(x^*)}{t}$$

Pela hipótese e pela preservação do sinal, existe  $\delta > 0$  tal que

$$\frac{f(x^* + td) - f(x^*)}{t} < 0$$

para todo  $\|t\| < \delta$ ,  $t \neq 0$ . Portanto,  $f(x^* + td) < f(x^*)$ , para todo  $t \in (0, \delta)$ , o que completa a demonstração.

Quando  $d_k = -B_k^{-1}\nabla f(x_k)$  e  $B_k$  é definida positiva, nós temos

$$d_k^T \nabla f(x_k) = \nabla f(x_k)^T B_k^{-1} \nabla f(x_k) < 0$$

concluindo que  $d_k$  é uma direção de descida. Desta forma é possível conseguirmos sempre uma direção de descida, basta considerarmos  $d^k = -B_k^{-1}\nabla f(x^k)$ . Dependendo do método escolhido alteramos o valor de  $B_k$ .

## Método do Gradiente

O método do gradiente é o método de busca linear que move ao longo de

$$d_k = -\nabla f(x_k) \quad (2.20)$$

a cada passo. Esta escolha se justifica com a observação de que, localmente, essa é a direção na qual a função  $f$  decresce mais rapidamente.

De fato,

Se  $d = -\nabla f(x)$  e  $v \in \mathbb{R}^n$  é tal que  $\|v\| = \|d\|$ , então

$$\frac{\partial f}{\partial d}(x) = \nabla f(x)^T d = -\|\nabla f(x)\|^2 = -\|\nabla f(x)\| \|v\| \leq \nabla f(x)^T v = \frac{\partial f}{\partial v}(x)$$

Uma vantagem deste método é que ele requer o cálculo do gradiente da  $f$  mas não requer o cálculo da segunda derivada. A desvantagem é que este método pode ser lento para problemas difíceis. O algoritmo do método gradiente será o mesmo que o algoritmo 2.16 basta considerarmos  $B_k$  igual a identidade.

A figura 4.20 mostra 4 iterações do algoritmo com busca exata aplicado para minimizar uma função quadrática convexa. Conforme podemos ver temos que duas direções consecutivas são ortogonais e a convergência é linear.

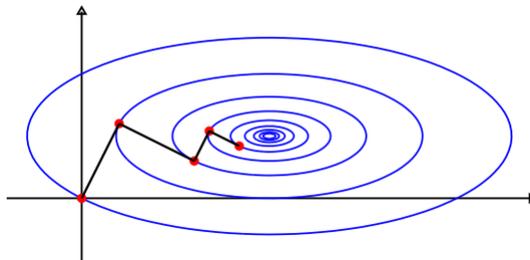


Figura 2.2: Convergência do método gradiente

## Método de Newton

O método que vamos estudar foi proposto por Isaac Newton. Este é um processo iterativo de fundamental importância no desenvolvimento de muitos algoritmos de programação não linear. Quando o problema é de minimização irrestrita a ideia básica para resolvê-lo é aproximarmos a função  $f(x_{k+1})$  pela série de Taylor de segunda ordem em torno do ponto  $x_k$ . Esta aproximação é dada da seguinte forma:

$$f(x_{k+1}) \approx f(x_k) + d^T \nabla f(x_k) + \frac{1}{2} d^T \nabla^2 f(x_k) d \quad (2.21)$$

A fim de que  $x_{k+1}$  seja o mínimo da aproximação temos que este ponto deve satisfazer a condição necessária de primeira ordem. Ou seja

$$\nabla f(x_{k+1}) = 0 \Rightarrow \nabla f(x_k) + \nabla^2 f(x_k)(x_{k+1} - x_k) = 0 \quad (2.22)$$

$$d_k = (\nabla^2 f(x_k))^{-1} \nabla f(x_k) \quad (2.23)$$

O método de Newton se baseia na busca de um ponto para o qual o gradiente de  $f(x_{k+1})$  é nulo. Este método não garante que o processo iterativo convirja para um ponto de mínimo, podendo divergir ser atraído para um ponto de inflexão ou um ponto de máximo. No caso em que a função é quadrática o algoritmo do método de Newton, considerando 2.23, converge para a solução em apenas uma iteração. Mas geralmente as funções não são quadráticas, embora sejam duas vezes diferenciáveis para aplicabilidade do método.

Lembramos que as condições necessárias e suficientes (minimização de funções) para que  $x_k$  seja mínimo local são:

Condições Necessárias:  $\nabla f(x_k) = 0$  e  $\nabla^2 f(x_k) \geq 0$  (semi definida positiva).

Condições Suficientes:  $\nabla f(x_k) = 0$  e  $\nabla^2 f(x_k) > 0$  (definida positiva)

Logo se  $\nabla^2 f(x_k)$  é definida positiva  $x_k$  será um ponto de mínimo da função 2.21.

Quando  $\nabla^2 f(x_k)$  não é definida positiva, a direção de Newton pode não ser uma direção de descida e  $(\nabla^2 f(x_k))^{-1}$  pode não existir, neste caso é necessário modificarmos a direção  $d_k$  de modo que a hessiana  $\nabla^2 f(x_k)$  seja definida positiva garantindo assim que  $d_k = x_{k+1} - x_k$  seja uma direção de descida. Para garantir que o algoritmo produza a diminuição do valor da função objetivo, é empregada uma variação do algoritmo de Newton : minimização unidimensional em cada direção, ou seja:

Encontramos  $t_k$  que minimiza a função unidimensional

$$\min_{t_k > 0} f(x_k + t_k d_k)$$

onde

$$x_{k+1} = x_k - t_k (\nabla^2 f(x_k))^{-1} \nabla f(x_k) \quad (2.24)$$

Para todo  $x$  na vizinhança da solução  $x_*$  tal que  $\nabla^2 f(x_*)$  é definida positiva, a hessiana  $\nabla^2 f(x)$  também será definida positiva. O método de Newton será bem definido nesta região e convergirá quadraticamente, com a condição de que o comprimento de passo  $t_k$  seja sempre 1.

A desvantagem de usarmos o método de Newton é a necessidade de calcularmos a hessiana  $\nabla^2 f(x)$ . O cálculo da segunda derivada algumas vezes pode ser caro, ter alguns erros e ser trabalhoso. Diferenças finitas e técnicas automática de diferenciação podem ser úteis para evitarmos o cálculo da hessiana.

**Teorema 2.24.** *Suponha que  $f$  é duas vezes diferenciável e que a hessiana  $\nabla^2 f(x)$  é Lipschitziana contínua numa vizinhança da solução  $x_*$  o qual a condição suficiente 2.10 é satisfeita. Considere a iteração  $x_{k+1} = x_k + d_k$ , onde  $d_k$  é dado por 2.24. Então*

- i) Se o ponto inicial esta suficientemente próximo da solução  $x_*$ , a sequência de iterações converge para  $x_*$ ;
- ii) a taxa de convergência de  $x_k$  é quadrática e;
- iii) a sequência da norma do gradiente  $\|\nabla f(x_k)\|$  converge quadraticamente para zero.

**Demonstração** Ver demonstração em [17]

**algoritmo 2.25.** Newton

Passo 1 Dado  $x_0 \in \mathbb{R}^n$   $k = 0$

Passo 2 Repita enquanto  $\nabla f(x_k) \neq 0$

Passo 3 Defina  $d_k = (-\nabla^2 f(x_k))^{-1} \nabla f(x_k)$

Passo 4 Determine o tamanho de passo  $t_k > 0$  através do *backtracking* ;

Passo 5 faça:  $x_{k+1} = x_k + t_k d_k$ ;

Passo 6  $k = k + 1$  ;

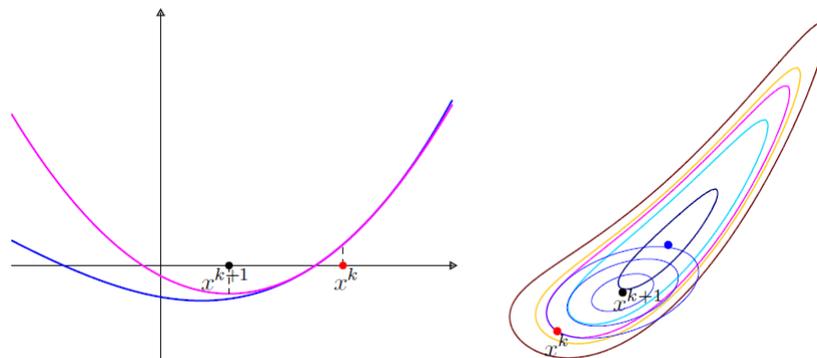


Figura 2.3: Iteração método de Newton

### Método Newton Corrigido

Para pontos distantes da solução a matriz hessiana  $\nabla^2 f(x)$  pode não ser definida positiva, assim não podemos garantir que a direção de Newton definida por 2.24 é uma direção de descida. Para solucionarmos este tipo de problema aplicamos um processo de positividade da matriz. O método mais simples é obtida somando uma matriz diagonal positiva junto a matriz hessiana  $\nabla^2 f(x_k)$ . Neste caso a nova matriz pode ser escrita da seguinte forma:

$$\nabla^2 f(x_k) + \epsilon_k I$$

Para testar a positividade tentamos escrever esta nova matriz na forma fatorada de Choleski. Caso for possível esta nova matriz será definida positiva.

$$\epsilon_k I + \nabla^2 f(x_k) = GG^T$$

Assim usamos esta matriz para encontrar a direção  $d_k$ . Ou seja  $d_k$  agora é calculado a partir da fórmula

$$(\epsilon_k I + \nabla^2 f(x^k))d_k = -\nabla f(x_k) \quad (2.25)$$

Caso  $\epsilon_k I + \nabla^2 f(x^k)$  continuar negativo aumentamos o valor de  $\epsilon$  e calculamos novamente até conseguirmos uma matriz definida positiva.

No método que foi implementado consideramos o valor de  $\epsilon_k = 1$  e caso a matriz continue negativa realizamos a soma da matriz identidade junto a matriz hessiana até conseguirmos uma matriz definida positiva. Segue abaixo o algoritmo que implementamos para Newton corrigido.

**algoritmo 2.26.** (Newton Corrigido)

Passo 1 Entradas: Dado  $f, gf, hf, tol$  e  $x_0$ ;

Passo 2 Fatore a matriz hessiana  $\nabla^2 f(x_k)$  na forma de cholesky. Caso for possível resolve

$$\nabla^2 f(x_k)d_k = -\nabla f(x_k)$$

Se não for possível faça  $\nabla^2 f(x_k)d_k + I$  e tente a fatoração outra vez até conseguir uma matriz definida positiva;

Passo 3 Determine o tamanho de passo  $t_k$  através do backtracking ;

Passo 4 Faça:  $x_{k+1} = x_k + t_k d_k$

Passo 5 Faça  $k = k + 1$ ;

## Método Quase Newton

Este método foi escrito baseado na referência [17]. O método de Quase Newton providencia uma direção alternativa ao método de Newton o qual não requer o cálculo da matriz Hessiana mantendo uma convergência superlinear. Este método é eficiente para problemas grandes e não é muito eficiente para problemas pequenos (1 mil, 2 mil variáveis).

No lugar da verdadeira matriz hessiana  $\nabla^2 f(x_k)$ , é usado uma aproximação  $B_k$ , o qual é necessário apenas o gradiente da função objetivo. Nós escolhemos a nova matriz (aproximação da Hessiana)  $B_{k+1}$  de maneira que satisfaça a propriedade 2.26.

$$\nabla^2 f(x_k)(x_{k+1} - x_k) \approx \nabla f(x_{k+1}) - \nabla f(x_k) \quad (2.26)$$

Satisfazer a condição 2.26 é equivalente satisfazer a seguinte condição, conhecida como equação secante:

$$B_{k+1}s_k = y_k$$

onde

$$s_k = x_{k+1} - x_k \quad \text{e} \quad y_k = \nabla f(x_{k+1}) - \nabla f(x_k)$$

O método proposto por Broyden, Fletcher, Goldfarb e Shanno é um dos mais conhecidos entre os métodos secantes. Segue abaixo a fórmula:

$$B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{y_k y_k^T}{y_k^T s_k} \quad (2.27)$$

A direção do método quase newton é obtida usando  $B_k$  no lugar da verdadeira matriz hessiana, ou seja:

$$d_k = -B_k^{-1} \nabla f(x_k)$$

Para implementarmos o método de quase newton evitamos fatorar  $B_k$  em cada iteração, atualizamos assim a inversa de  $B_k$ , em vez da própria  $B_k$ . De fato, a fórmula equivalente para 2.27 considerando a aproximação da matriz inversa  $B_k^{-1}$  como  $H_k$  é

$$H_{k+1} = (I - \rho_k s_k y_k^T) H_k (I - \rho_k y_k s_k^T) + \rho_k s_k s_k^T, \quad (2.28)$$

$$\rho_k = \frac{1}{y_k^T s_k}$$

Assim podemos redefinir a direção BFGS como:

$$d_k = -H_k \nabla f(x_k) \quad (2.29)$$

Antes de apresentar o algoritmo do método BFGS é necessário definir a aproximação inicial  $H_0$ .

Note que não temos uma aproximação inicial que funcione bem em todos os casos. Algumas alternativas são:

- Utilizar informação do problema, por exemplo,  $H_0$  pode ser definida como a inversa da aproximação por diferenças finitas da hessiana de  $f$  em  $x_0$ ;
- Utilizar a matriz identidade ou múltiplo da matriz identidade  $H_0 = \beta I$ .

**algoritmo 2.27.** (Quase Newton BFGS)

Passo 1  $k \leftarrow 0$ ,  $B_0 = \nabla^2 f(x_0)$  ou  $B_0 = I$ ;

Passo 2 Enquanto  $\|\nabla f(x_k)\| > \varepsilon$ ;

Passo 3 Determinar  $d_k$  tal que

$$B_k d_k = -\nabla f(x_k) \text{ ou } d_k = -H_k \nabla f(x_k)$$

Passo 4 Encontre o valor para  $t_k$  através do algoritmo retrocesso;

Passo 5 Faça  $x_{k+1} = x_k + t_k d$ .

Verifica o passo 2, caso ainda não for verdade atualiza a matriz  $B_k$

Passo 7 Faça  $k \leftarrow k + 1$  e volte para o passo 3.

### 2.1.8 Testes realizados com os Algoritmos Implementados

Realizamos testes em três funções quadráticas descritas nos exemplos 1, 2 e 3 . Para estes testes foram considerados quatro algoritmos : Gradiente , Newton, Newton-Corrigido e Quase-Newton-BFGS. Nas tabelas 2.1 e 2.2 podemos verificar os seguintes resultados: Para o primeiro exemplo, considerando os quatro métodos, obtemos a solução em duas iterações , o ponto mínimo encontrado foi  $[-0.5, -0.5]$  e o valor da função neste ponto foi  $-1$ . Para o segundo exemplo , considerando os métodos de Newton e Newton-Corrigido, obtemos a solução em duas iterações e 20 iterações com os métodos do Gradiente e Quase-Newton-BFGS. O ponto mínimo foi  $0.6667$  e o valor da função neste ponto foi  $2.6667$ . Para o último exemplo obtemos a solução em duas iterações considerando o método de Newton e dezessete iterações considerando o método de Quase-Newton-BFGS, já com os outros dois métodos não tivemos bom desempenho. O ponto mínimo encontrado para este último exemplo foi  $[-0.333, -0.5]$  e o valor da função neste ponto foi  $-0.333$ . Através da informação na tabela concluímos que a velocidade de convergência foi rápida para a maioria dos casos.

**Exemplo 1.**  $f(y) = 2y_1^2 + 2y_2^2 + 2y_1 + 2y_2$

Tabela 2.1: Exemplo 1

Rotina	n° iterações	tempo(ms)	f(x)	x	$x_0$
Gradiente	2	0.745593	-1	[-0.5,-0.5]	[0,0]
Newton	2	0.004630	-1	[-0.5,-0.5]	[0,0]
Newton-Corrigido	2	0.004486	-1	[-0.5,-0.5]	[0,0]
Quase-Newton-BFGS	2	0.019491	-1	[-0.5,-0.5]	[0,0]

**Exemplo 2.**  $f(y) = 12y_1^2 - 16y_1 + 8$

Tabela 2.2: Exemplo 2

Rotina	n° iterações	tempo(ms)	f(x)	x	$x_0$
Gradiente	22	0.002747	2.6667	0.6667	0
Newton	2	0.000565	2.6667	0.6667	0
Newton-Corrigido	2	0.000691	2.6667	0.6667	0
Quase-Newton-BFGS	18	0.002936	2.6667	0.6667	0

**Exemplo 3.**  $f(y_1, y_2) = 12y_1^2 + 4y_2^2 - 12y_1y_2 + 2y_1$

Tabela 2.3: Exemplo 3

Rotina	n° iterações	tempo(ms)	f(x)	x	$x_0$
Gradiente	118	0.698897	-0.333	[-0.333,-0.5]	0
Newton	2	0.004880	-0.333	[-0.3333,-0.5]	0
Newton-Corrigido	divergiu	-	-	-	0
Quase-Newton-BFGS	17	0.016002	-0.3333	[-0.3333,-0.5]	0

## 2.2 PROBLEMAS DE MÍNIMOS QUADRADOS

Nesta seção apresentamos a solução de problemas de mínimos quadrados não linear. Este problema está relacionado com problemas de minimização irrestrita e problemas de equações não lineares. Vamos explorar duas diferentes aproximações para resolver este tipo de problema: O primeiro está relacionado a resolver um sistema de equações não linear o qual utilizamos o método de Gauss Newton e o segundo está relacionado com problema de minimização irrestrita o qual utilizamos o método de Newton, os dois métodos estão relacionados entre si.

O problema de mínimo quadrado não linear é

$$\min f(x) = \frac{1}{2} R(x)^T R(x) = \frac{1}{2} \sum_{i=1}^m r_i^2(x) \quad (2.30)$$

onde cada  $m > n$ , a função resíduo  $\mathbb{R} : \mathbb{R}^n \rightarrow \mathbb{R}^m$  é não linear em  $x$ , e  $r_i(x)$  denota cada função componente de  $R(x)$ . O problema de mínimo quadrado não linear aparece mais comumente nas aplicações onde adequamos os dados observados  $(t_i, y_i)$ ,  $i = 1, \dots, m$  com uma função modelo  $m(x, t)$  que é não linear em  $x$ . Neste caso  $r_i(x) = m(x, t_i) - y_i$  e o problema de mínimo quadrado consiste na escolha de  $x$  de tal forma que a soma dos quadrados dos desvios seja mínima. É claro se a soma

$$\sum_{i=1}^m r_i^2(x) = \sum_{i=1}^m (m(x, t_i) - y_i)^2$$

é mínima, teremos que cada parcela  $(m(x, t_i) - y_i)^2$  é pequena, donde cada desvio  $(m(x, t_i) - y_i)$  é pequeno.

Observe que se o modelo ajustar exatamente os dados, o mínimo da função acima será zero. Quando  $m = n$  temos um caso especial de um problema de sistema de equações não linear, para o caso em que  $m > n$  temos um problema de minimização irrestrita. A principal razão de não resolvermos este tipo de problema pelo método geral de minimização irrestrita é devido a estrutura da da função dada por 2.30.

Sabemos que para obter um ponto mínimo de 2.30 temos de inicialmente encontrar seus pontos críticos, ou seja precisamos inicialmente encontrar as derivadas de  $R(x)$  e de  $f(x) = \frac{1}{2} R(x)^T R(x)$ .

A primeira derivada de  $R(x)$  é simplesmente a matriz Jacobiana  $\in \mathbb{R}^{m \times n}$  onde

$$J(x)_{ij} = \left[ \frac{\partial r_i(x)}{\partial x_j} \right]$$

Um modelo afim de  $\mathbb{R}(x)$  ao redor do ponto  $x_k$  é

$$M(x_{k+1}) = R(x_k) + J(x_k)(x_{k+1} - x_k)$$

Esta é a generalização para o caso  $m \neq n$  do modelo afim para sistema de equações não lineares. Este modelo será a base para o método de Gauss Newton o qual relaciona o problema de mínimo quadrado com um sistema de equações sobre determinado.

A primeira derivada de  $f(x) = \frac{1}{2}R(x)^T R(x)$  é

$$\nabla f(x) = \sum_{i=1}^m r_i(x) \nabla r_i(x) = J(x)^T R(x) \quad (2.31)$$

Similarmente, a segunda derivada é

$$\begin{aligned} \nabla^2 f(x) &= \sum_{j=1}^m \nabla r_j(x) \nabla r_j(x)^T + \sum_{j=1}^m r_j(x) \nabla^2 r_j(x) \\ &= J(x)^T J(x) + S(x) \end{aligned} \quad (2.32)$$

onde  $S(x) = \sum_{i=1}^m r_i(x) \nabla^2 r_i(x)$  denota a informação de segunda ordem em  $\nabla^2 f(x)$ . Assim o modelo quadrático de  $f(x)$  ao redor de  $x_k$  é

$$\begin{aligned} m_k(x) &= f(x_k) + \nabla f(x_k)^T (x - x_k) + \frac{1}{2} (x - x_k)^T \nabla^2 f(x_k) (x - x_k) \\ &= \frac{1}{2} R(x_k)^T R(x_k) + R(x_k)^T J(x_k) (x - x_k) + \frac{1}{2} (x - x_k)^T (J(x_k)^T J(x_k) \\ &\quad + S(x_k)) (x - x_k) \end{aligned}$$

Este modelo quadrático da série de Taylor é utilizado para minimizar função objetivo do tipo 2.30. De 2.33 o método de Newton aplicado para 2.30 é

$$x_{k+1} = x_k - \left( J(x_k)^T J(x_k) + \sum_{j=1}^m r_j(x_k) \nabla^2 r_j(x_k) \right)^{-1} J(x_k)^T r(x_k) \quad (2.33)$$

Certamente deveria ser o método mais rápido para resolver problemas de mínimos quadrados, desde que a função fosse quadraticamente convergente. O problema com a aproximação de Newton é que  $S(x)$  é usualmente inviável ou inconveniente e aproximar por diferenças finitas também é caro. Embora na prática costumam aproximar a matriz hessiana pelo método da secante aqui é indesejável porque a porção de  $J(x)^T J(x)$  já será avaliada, sendo  $J(x)$  calculada analiticamente ou por diferenças finitas a fim de obtermos o gradiente da  $f$ . Os termos zero resíduo, pequeno resíduo ou largo resíduo refere aos termos de  $R(x)$  ou  $f(x)$  no mínimo de 2.30. O problema para o qual  $R(x^*) = 0$  é chamado um problema de zero-residual.

**Exemplo 4.** Suponha que desejamos adequar os dados  $(t_i, y_i)$ ,  $i = 1, \dots, 4$  com o modelo  $m(x, t_i) = e^{t_i x_1} + e^{t_i x_2}$ , usando mínimos quadrados não linear. Então  $R : R^2 \rightarrow R^4$ ,  $r_i(x) =$

$e^{t_i x_1} + e^{t_i x_2} - y_i$ ,  $i = 1, \dots, 4$  e o problema de mínimo quadrado não linear é minimizar  $f(x) = \frac{1}{2} R(x)^T R(x)$ . Assim nós temos  $J(x) \in \mathbb{R}^{4 \times 2}$

$$J(x) = \begin{bmatrix} t_1 e^{t_1 x_1} & t_1 e^{t_1 x_2} \\ t_2 e^{t_2 x_1} & t_2 e^{t_2 x_2} \\ t_3 e^{t_3 x_1} & t_3 e^{t_3 x_2} \\ t_4 e^{t_4 x_1} & t_4 e^{t_4 x_2} \end{bmatrix}$$

Também  $\nabla f(x) \in \mathbb{R}^2$

$$\nabla f(x)^T = R(x)^T J(x) = \left( \sum_{i=1}^4 r_i(x) t_i e^{t_i x_1}, \sum_{i=1}^4 r_i(x) t_i e^{t_i x_2} \right)$$

e

$$\nabla^2 f(x) \in \mathbb{R}^{2 \times 2}.$$

Usando

$$\nabla^2 r_i(x) = \begin{bmatrix} t_i^2 e^{t_i x_1} & 0 \\ 0 & t_i^2 e^{t_i x_2} \end{bmatrix}$$

e

$$\nabla^2 f(x) = J(x)^T J(x) + \sum_{i=1}^4 r_i(x) \nabla^2 r_i(x)$$

nós temos

$$= \begin{bmatrix} \sum_{i=1}^4 t_i^2 e^{t_i x_1} (r_i(x) + e^{t_i x_1}) & \sum_{i=1}^4 t_i^2 e^{t_i (x_1 + x_2)} \\ \sum_{i=1}^4 t_i^2 e^{t_i (x_1 + x_2)} & \sum_{i=1}^4 t_i^2 e^{t_i x_2} (r_i(x) + e^{t_i x_2}) \end{bmatrix}$$

### 2.2.1 Método de Gauss Newton

O segundo método que consideramos para resolver o problema de mínimo quadrado usa o modelo afim de  $R(x)$  ao redor de  $x_k$  dado por

$$M_k(x_{k+1}) = R(x_k) + J(x_k)(x_{k+1} - x_k) \quad (2.34)$$

onde

$$M_k : \mathbb{R}^n \rightarrow \mathbb{R}^m$$

e  $m > n$ .

Nós podemos em geral esperar encontrar um  $x_{k+1}$  para o qual  $M_k(x_{k+1}) = 0$ , desde que o sistema de equação não linear seja sobre determinado. Entretanto um caminho para usar 2.34

para resolver um problema de mínimo quadrado não linear é escolher o próximo iterando  $x_{k+1}$  como solução do problema de mínimo quadrado linear

$$\min_{x \in \mathbb{R}^n} \frac{1}{2} \| M_k(x) \|_2^2 \triangleq \hat{m}_k(x) \quad (2.35)$$

Assuma agora que  $J(x_k)$  tenha coluna posto completo. Então a solução para 2.35 é

$$x_{k+1} = x_k - (J(x_k)^T J(x_k))^{-1} J(x_k)^T R(x_k) \quad (2.36)$$

O método iterativo que usa 2.36 é chamado de método de Gauss Newton. Note que a única diferença entre as equações 2.36 e 2.33 é o termo  $S(x)$  incluído pelo método de Newton na segunda derivada da matriz  $J(x_k)^T J(x_k) + S(x_k)$  mas omitido pelo método de Gauss Newton. Desde que o método de Newton é localmente quadraticamente convergente o método de Gauss Newton vai convergir quadraticamente se  $S(x_*) = 0$ . Isto ocorre quando  $R(x)$  é linear ou quando temos um problema de zero residual. Se  $S(x_*)$  é pequeno relativo a  $J(x_*)^T J(x_*)$  o método de Gauss Newton converge localmente linearmente. Entretanto, se  $S(x_*)$  é grande, o método de Gauss Newton pode não ser localmente convergente.

**Teorema 2.28.** *Seja  $\mathbb{R} : \mathbb{R}^n \rightarrow \mathbb{R}^m$ , e seja  $f(x) = \frac{1}{2} R(x)^T R(x)$  duas vezes diferenciável contínua num subconjunto aberto  $D \subset \mathbb{R}^n$ . Assuma que  $J(x) \in Lip_\gamma(D)$  com  $\| J(x) \|_2 \leq \alpha$  para todo  $x \in D$ , e que existe  $x_* \in D$  e  $\lambda, \sigma \geq 0$ , tal que  $J(x_*)^T R(x_*) = 0$ ,  $\lambda$  é o menor autovalor de  $J(x_*)^T J(x_*)$ , e*

$$\| ((J(x) - J(x_*))^T R(x_*)) \|_2 \leq \sigma \| x - x_* \|_2$$

para todo  $x \in D$ . Se  $\sigma < \lambda$ , então para algum  $c \in (1, \frac{\lambda}{\sigma})$ , existe  $\epsilon > 0$  tal que para todo  $x_0 \in N(x_*, \epsilon)$  a sequência gerada pelo método de Gauss Newton

$$x_{k+1} = x_k - (J(x_k)^T J(x_k))^{-1} J(x_k)^T R(x_k) \quad (2.37)$$

esta bem definida converge para  $x_*$ , e obedece

$$\| x_{k+1} - x_* \|_2 \leq \frac{c\sigma}{\lambda} \| x_k - x_* \|_2 + \frac{c\alpha\gamma}{2\lambda} \| x_k - x_* \|_2^2 \quad (2.38)$$

e também

$$\| x_{k+1} - x_* \|_2 \leq \frac{c\sigma + \lambda}{2\lambda} \| x_k - x_* \|_2 < \| x_k - x_* \|_2 \quad (2.39)$$

**Corolário 2.29.** *Suponha que as hipóteses do teorema 2.28 sejam satisfeitas. Se  $R(x_*) = 0$  então existe  $\epsilon > 0$  tal que para todo  $x_0 \in N(x_*, \epsilon)$  a sequência  $x_k$  gerada pelo método de Gauss Newton está bem definida e converge quadraticamente para  $x_*$ .*

**Demonstração** Ver demonstração em [16]

### Vantagens do método Gauss Newton

- Não realiza o cálculo da matriz hessiana da função ou das componentes da função resíduo, o que diminui o esforço computacional necessário;
- Convergência rápida e linear em problemas que não são linear e têm pequenos resíduos;
- Convergência local e rápida com ordem linear em problemas linear e com resíduo-pequeno;
- Resolve problemas de quadrados mínimos lineares em uma iteração;

### Desvantagens do método Gauss Newton

- Não podemos garantir sua convergência global;
- Não está bem definido quando a matriz Jacobiana não tem posto completo;
- Converge lentamente com ordem linear em problemas com resíduos grandes ou que são suficientemente não lineares;
- Não converge localmente sobre problemas que não são lineares e têm resíduo grande;

Esta seção foi baseada no livro [16].

#### algoritmo 2.30. Newton

Passo 1 Dado  $x_0 \in \mathbb{R}^n$   $k = 0$

Passo 2 Repita enquanto  $J(x_k) \neq 0$

Passo 3 Defina  $d_k = -(J(x_k)^T J(x_k))^{-1} J(x_k)^T f(x_k)$

Passo 4 Determine o tamanho de passo  $t_k > 0$  através do *backtracking* ;

Passo 5 faça:  $x_{k+1} = x_k + t_k d_k$ ;

Passo 6  $k = k + 1$  ;

### 2.3 EQUAÇÕES NÃO LINEARES

Nesta seção vamos estudar o problema de encontrar a solução de um sistema de equação não linear:

$$\text{dado } F : \mathbb{R}^n \rightarrow \mathbb{R}^n \text{ encontre } x_* \text{ tal que } F(x_*) = 0 \quad (2.40)$$

onde  $F$  é assumida ser contínua diferenciável. Para resolver o problema 2.40 vamos usar o método de Newton. A ideia do método de Newton para sistema não linear é construir um modelo linear nas proximidades de um ponto  $x_k$  e calcular a raiz deste modelo, que será usado como novo ponto central para outro modelo afim assim fazemos este procedimento sucessivamente até encontrar a solução.

Suponha que  $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$  seja diferenciavelmente contínua num subconjunto convexo  $D$  e que  $x$  e  $x + d$  são vetores em  $D$ . Pelo teorema fundamental do cálculo temos

$$F(x) = F(x_k) + \int_{x_k}^x J(z) dz \quad (2.41)$$

Podemos aproximar  $F(x)$  por uma função linear  $M_k$  resolvendo o segundo termo da equação 2.41. De fato,

Pelo Teorema do Valor Médio, existe  $d \in [x, x_k]$  tal que

$$F(x) - F(x_k) = J(d)(x - x_k) \quad (2.42)$$

Considerando que  $x_k$  esta convergindo para  $x$  temos também que  $x_k$  esta se aproximando de  $d$ , devido a continuidade de  $J$  temos que  $J(d) \approx J(x_k)$ , de onde segue que

$$\int_{x_k}^x J(z) dz = J(x_k)(x - x_k) \quad (2.43)$$

Assim, podemos escrever a  $F(x)$  como aproximação linear dada por,

$$F(x) \cong M_k(x_k + d) = F(x_k) + J(x_k) * d \quad (2.44)$$

onde  $d = x - x_k$ ,  $J(x)$  é a matriz Jacobiana de  $r$  e  $d \in \mathbb{R}^n$  é a direção. Para obtermos a direção  $d_k$  consideramos  $M_k(x_k + d_k) = 0$ . Assim, temos

$$M_k(x_k + d) = 0 \Leftrightarrow \quad (2.45)$$

$$F(x_k) + J(x_k) * d_k = 0 \quad (2.46)$$

$$d_k = -J(x_k)^{-1} F(x_k) \quad (2.47)$$

Este é o passo do método de Newton para sistemas não lineares. Assim podemos definir o processo iterativo considerando

$$x_{k+1} = x_k + t_k d_k \quad (2.48)$$

**algoritmo 2.31.** (Método de Newton para Equações Não Linear )

Passo 1 Dado  $x_0 \in \mathbb{R}^n$   $k = 0$

Passo 2 Repita enquanto  $J(x_k) \neq 0$

Passo 3 Defina  $J(x_k)d_k = -F(x_k)$

Passo 4 Determine o tamanho de passo  $t_k > 0$  através do *backtracking* ;

Passo 5 faça:  $x_{k+1} = x_k + t_k d_k$ ;

Passo 6  $k = k + 1$  ;

Passo 7 Fim;

**Exemplo 5.**

$$F(x) = \begin{bmatrix} x_1 + x_2 - 3 \\ x_1^2 + x_2^2 - 9 \end{bmatrix}$$

o qual tem as raízes  $(3, 0)^T$  e  $(0, 3)^T$ , e seja  $x_0 = (1, 5)^T$ . Então as duas primeiras iterações do método de Newton são:

$$J(x_0)d_0 = -F(x_0): \begin{bmatrix} 1 & 1 \\ 2 & 10 \end{bmatrix} d_0 = - \begin{bmatrix} 3 \\ 17 \end{bmatrix}, d_0 = \begin{bmatrix} \frac{-13}{8} \\ \frac{-11}{8} \end{bmatrix}$$

$$x_1 = x_0 + d_0 = (-0.625, 3.625)^T$$

$$J(x_1)d_1 = -F(x_1): \begin{bmatrix} 1 & 1 \\ -\frac{5}{4} & \frac{29}{4} \end{bmatrix} d_1 = - \begin{bmatrix} 0 \\ \frac{145}{32} \end{bmatrix}, d_1 = \begin{bmatrix} \frac{145}{272} \\ \frac{-145}{272} \end{bmatrix}$$

$$x_2 = x_1 + d_1 \cong (-0.625, 3.625)^T$$

Note que o método de Newton funcionou bem para este exemplo, pois  $x_2$  obtido na segunda iteração se aproximou bem da raiz. Esta é a vantagem do método de Newton. Se  $x_0$  está próximo suficiente da solução  $x_*$  e  $J(x_*)$  é não singular  $x_k$  gerado pelo algoritmo 2.31 converge quadraticamente para  $x_*$ .

### Vantagens

- Convergência quadrática quando o ponto inicial está próximo da solução ;
- Exata solução em uma única iteração para uma função afim de  $F$ ;

### Desvantagens

- Não é globalmente convergente;
- Requer  $J(x_k)$  em cada iteração;
- Cada iteração requer a solução do sistema de equações lineares que pode ser singular ou mal condicionado;

Esta seção foi baseada em [16].

Os exemplos a seguir descevem quatro sistemas não lineares. Vamos encontrar a solução destes exemplos através do método de Gauss Newton e Newton para sistema não-linear.

1° Exemplo

$$F(x) = \begin{bmatrix} x_1^2 + x_2^2 - 4 \\ x_1^2 - x_2^2 - 1 \end{bmatrix}$$

2° Exemplo

$$F(x) = \begin{bmatrix} x_1^2 + x_2 - 2 \\ x_1^2 - \frac{x_2^2}{9} - 1 \end{bmatrix}$$

3° Exemplo

$$F(x) = \begin{bmatrix} x_1^2 - x_2^2 - 0.2 \\ x_2^2 - x_1 + 1 \end{bmatrix}$$

4° Exemplo

$$F(x) = \begin{bmatrix} e^{2x_1+x_2} - x_1 \\ x_1^2 - x_2 \end{bmatrix}$$

Realizamos testes em quatro exemplos os quais encontramos a solução através dos métodos de Gauss Newton e Newton para Sistema Não Linear. Considerando os dois métodos observamos os seguintes resultados nas tabelas 2.4 e 2.5: A solução para o 1° exemplo foi obtida em cinco iterações e o ponto mínimo encontrado foi  $[1.5811, 1.2247]$ . Para o 2° exemplo a solução foi obtida em três iterações e o ponto mínimo encontrado foi  $[3.0244, -0.9954]$ . Para o 3° exemplo a solução foi obtida em três iterações e o ponto mínimo encontrado foi  $[0.5130, 0.4678]$ . Para o 4° exemplo a solução foi obtida em quatorze iterações considerando o método de Gauss Newton, a solução encontrada com este método foi  $[-0.0142, -0.6504]$ . Já com o método de Newton não encontramos solução.

Tabela 2.4: Método Gauss Newton

Problemas	n° iterações	tempo(ms)	Ponto inicial	solução
<i>ex1</i>	5	2	[1, 1]	[1.5811,1.2247]
<i>ex2</i>	3	2	[1, 1]	[3.0244,-0.9954]
<i>ex3</i>	3	3	[1, 1]	[0.5130,0.4678]
<i>ex4</i>	14	15	[1, 1]	[-0.0142,-0.6504]

Tabela 2.5: Método Newton para sistema não linear

Problemas	n° iterações	tempo(ms)	Ponto inicial	solução
<i>ex1</i>	5	5	[1, 1]	[1.5811,1.2247]
<i>ex2</i>	3	6	[1, 1]	[3.0244,-0.9954]
<i>ex3</i>	3	6	[1, 1]	[0.5130,0.4678]
<i>ex4</i>	estourou as iterações	666	[1, 1]	[-0.0142,-0.6504]

### 3 PROBLEMAS COM RESTRIÇÕES

#### 3.1 INTRODUÇÃO

Nesta seção apresentamos problema sujeito a restrição de igualdade e desigualdade sobre as variáveis. Segue abaixo o problema:

$$\begin{array}{ll} \min & f(x) \\ \text{sujeito a} & g_\beta(x) \leq 0 \\ & h_\xi(x) = 0 \end{array}$$

onde  $f : \mathbb{R}^n \rightarrow \mathbb{R}$ ,  $g_i : \mathbb{R}^n \rightarrow \mathbb{R}$  e  $h_i : \mathbb{R}^n \rightarrow \mathbb{R}$  e,  $i \in \xi \cup \beta$ , são funções de classes  $C^2$ . O conjunto

$$\Omega = \{x \in \mathbb{R}^n : h_i(x) = 0, i \in \xi \text{ e } g_i(x) \leq 0, i \in \beta\}$$

é chamado conjunto viável.

A suavidade da função objetivo e as restrições são características importantes na solução, justamente como no caso irrestrito. Isto garante que a função objetivo e as restrições todas se comportam de uma forma previsível, pelo menos localmente, permitindo desta forma que os algoritmos tomem decisões corretas com respeito às direções de busca. Tipicamente, fronteiras não suaves podem ser escritas por uma coleção de restrições suaves. A região factível  $\Omega$  pode ser escrita por uma única restrição não suave:

$$\|x\|_1 = |x_1| + |x_2| \leq 1$$

$\Omega$  também pode ser escrita pelas seguintes restrições suaves:

$$\begin{array}{ll} x_1 + x_2 \leq 1 & (3.1) \\ x_1 - x_2 \leq 1 \\ -x_1 + x_2 \leq 1 \\ -x_1 - x_2 \leq 1 \end{array}$$

### 3.2 CONDIÇÕES DE OTIMALIDADE PARA PROBLEMAS RESTRITOS

**Definição 3.1.** *Seja  $x_*$  um ponto satisfazendo as restrições*

$$\begin{aligned} h(x_*) &= 0 \\ g(x_*) &\leq 0 \end{aligned} \quad (3.2)$$

*e seja  $J$  o conjunto de índices  $j$  para o qual  $g_j(x_*) = 0$ . Então  $x_*$  é dito ser um ponto regular das restrições se os vetores gradientes  $\nabla h_i(x_*)$  para  $1 \leq i \leq m$  e  $\nabla g_j(x_*)$   $j \in J$  são linearmente independente.*

**Lema 3.2.** *Seja  $x_*$  um ponto regular das restrições e um mínimo local da  $f$  sujeita a estas restrições. Então para todo  $y \in E^n$  temos*

$$\nabla h(x_*)y = 0 \quad (3.3)$$

mas também satisfaz

$$\nabla f(x_*)y = 0 \quad (3.4)$$

**Demonstração** ver [8] página 326.

**Teorema 3.3.** *(Condição Necessária de 1º Ordem) Seja  $x_*$  mínimo local (máximo local) de  $f$  sujeita as restrições  $h(x) = 0$ . Considere que  $x_*$  é um ponto regular destas restrições. Então existe um  $\lambda \in E^m$  tal que*

$$\nabla f(x_*) + \lambda^T \nabla h(x_*) = 0 \quad (3.5)$$

**Demonstração** ver [8] página 327.

Podemos enunciar as condições necessárias de primeira ordem definindo a função Lagrangiana da seguinte forma:

$$l(x, \lambda) = f(x) + \lambda^T h(x) \quad (3.6)$$

Assim a condição necessária de primeira ordem passa a ser definida da seguinte maneira:

$$\nabla_x l(x, \lambda) = 0 \quad (3.7)$$

Esta observação sugere que podemos pesquisar por soluções procurando por pontos estacionários da função Lagrangiana.

Abaixo seguem alguns exemplos de problemas de otimização com restrições ativas.

**Exemplo 6.** Considere o problema

$$\begin{array}{ll} \min & x_1x_2 + x_2x_3 + x_1x_3 \\ \text{sujeito a} & x_1 + x_2 + x_3 = 3 \end{array}$$

Substituindo na função de Lagrange 3.6 obtemos:

$$l(x, \lambda) = x_1x_2 + x_2x_3 + x_1x_3 + \lambda^T(x_1 + x_2 + x_3) \quad (3.8)$$

Aplicando a condição de otimalidade, temos:

$$\nabla_x l(x, \lambda) = (x_2 + x_3 + \lambda, x_1 + x_3 + \lambda, x_2 + x_1 + \lambda) = 0$$

A solução para este problema é  $x_1 = x_2 = x_3 = 1$ , e  $\lambda = -2$

**Exemplo 7.** (Volume Máximo) Considere um problema que buscamos construir uma caixa de papelão de volume máximo, dado uma determinada área de papelão. Denotando as dimensões da caixa por  $x, y$  e  $z$  o problema pode ser expresso como

$$\begin{array}{ll} \max & xyz \\ \text{sujeito a} & (xy + yz + xz) = \frac{c}{2} \end{array}$$

onde  $c > 0$  é a área do papelão. Substituindo na função de Lagrange obtemos:

$$l(x, \lambda) = xyz + \lambda^T(xy + yz + xz) \quad (3.9)$$

Aplicando a condição de otimalidade obtemos

$$\nabla l(x, \lambda) = (yz + \lambda^T(y + z), xz + \lambda^T(x + z), xy + \lambda^T(x + y)) \quad (3.10)$$

Resolvendo o sistema abaixo, temos que  $x = y = z = \sqrt{\frac{c}{6}}$  é a única solução para a condição necessária.

### 3.2.1 Condições Necessária de Primeira Ordem Com Restrições De Igualdade e Desigualdade

Agora vamos voltar ao problema original, ou seja considerar o problema abaixo

$$\begin{array}{ll} \min & f(x) \\ \text{sujeito a} & g(x) \leq 0 \\ & h(x) = 0 \end{array}$$

Também vamos assumir que as dimensões para  $f, g$  e  $h$  são as mesmas consideradas anteriormente, e vamos considerar que estas funções são de classe  $C^2$ .

**Teorema 3.4** (Karush-Kuhn-Tucker Condições). *Seja  $x_*$  um mínimo relativo para o problema*

$$\begin{array}{ll} \min & f(x) \\ \text{sujeito a} & g(x) \leq 0 \\ & h(x) = 0 \end{array}$$

*e suponha que  $x_*$  é um ponto regular para a restrição. Então existe um vetor  $\lambda \in E^m$  e um vetor  $\mu \in E^p$  com  $\mu \geq 0$  tal que*

$$\nabla f(x_*) + \lambda^T \nabla h(x_*) + \mu^T \nabla g(x_*) = 0 \quad (3.11)$$

$$\mu^T g(x_*) = 0 \quad (3.12)$$

**Demonstração:** Ver página [8] 342.

**Exemplo 8.** Considere o problema

$$\begin{array}{ll} \min & 2x_1^2 + 2x_1x_2 + x_2^2 - 10x_1 - 10x_2 \\ \text{sujeito a} & x_1^2 + x_2^2 \leq 5 \\ & 3x_1 + x_2 \leq 6 \end{array}$$

A condição necessária de primeira ordem, em adição com as restrições são:

$$\begin{aligned}
4x_1 + 2x_2 - 10 + 2\mu_1x_1 + 3\mu_2 &= 0 \\
2x_1 + 2x_2 - 10 + 2\mu_1x_2 + \mu_2 &= 0 \\
\mu_1 &\geq 0, \mu_2 \geq 0 \\
\mu_1(x_1^2 + x_2^2 - 5) &= 0 \\
\mu_2(3x_1 + x_2 - 6) &= 0
\end{aligned}$$

Para encontrar a solução definimos várias combinações das restrições ativas. Para este problema podemos tentar definir nenhuma, uma ou duas restrições ativas. Assumindo que a primeira restrição é ativa e a segunda é inativa rende as seguintes equações:

$$4x_1 + 2x_2 - 10 + 2\mu_1x_1 = 0 \quad (3.13)$$

$$2x_1 + 2x_2 - 10 + 2\mu_1x_2 = 0 \quad (3.14)$$

$$x_1^2 + x_2^2 = 5 \quad (3.15)$$

o qual tem a solução  $x_1 = 1, x_2 = 2, \mu_1 = 1$

### 3.2.2 Condição de Qualificação

**Definição 3.5 (LICQ).** Dizemos que a condição de qualificação de independência linear (LICQ) é satisfeita em  $x_*$  quando o conjunto formado pelos gradientes das restrições de igualdade e das restrições de desigualdade ativas são linearmente independente, isto é,

$$\{\nabla g_i(x_*) | i \in \varepsilon \cup I(x_*)\}$$

é linearmente independente.

**Exemplo 9.** Considere duas restrições de desigualdades definidas por  $h_1, h_2 : \mathbb{R}^2 \rightarrow \mathbb{R}$ , onde  $h_1(x) = x_1^2 - 2x_1 - x_2$  e  $h_2 = x_1^2 - 2x_1 + x_2$ . Verifique que o ponto  $x_* = 0$  cumpre LICQ.

As duas restrições são ativas em  $x_* = 0$  e os vetores

$$\nabla h_1(x_*) = \begin{pmatrix} -2 \\ -1 \end{pmatrix}$$

e

$$\nabla h_2(x_*) = \begin{pmatrix} -2 \\ 1 \end{pmatrix}$$

são linearmente independentes.

Apesar desta simplicidade, LICQ tem a desvantagem de ser uma hipótese muito forte para garantir KKT. Existem muitos problemas em que temos KKT sem que LICQ seja satisfeita.

**Exemplo 10.** Considere o problema

$$\begin{aligned} \min \quad & f(x) = x_1 \\ & h_1(x) = x_1^2 - 2x_1 - x_2 \leq 0 \\ \text{sujeito a} \quad & h_2(x) = x_1^2 - 2x_1 + x_2 \leq 0 \\ & h_3(x) = -x_1 \leq 0 \end{aligned}$$

O ponto  $x_* = 0$  é o minimizador deste problema, cumpre as condições de KKT mas não satisfaz LICQ.

De fato, as três restrições são ativas em  $x_*$  e os vetores

$$\nabla h_1(x_*) = \begin{pmatrix} -2 \\ -1 \end{pmatrix}$$

e

$$\nabla h_2(x_*) = \begin{pmatrix} -2 \\ 1 \end{pmatrix}$$

e

$$\nabla h_3(x_*) = \begin{pmatrix} -1 \\ 0 \end{pmatrix}$$

são linearmente dependentes. Além disso,

$$-\nabla f(x_*) = \begin{pmatrix} -1 \\ 0 \end{pmatrix} = \nabla h_3(x_*)$$

ou seja, vale KKT.

Este exemplo motiva o estudo de hipóteses mais fracas mas que ainda sejam facilmente verificadas. Uma delas é a condição de Mangasarian e Fromovitz.

**Definição 3.6.** A condição de Mangasarian-Fromovitz (MFCQ) é satisfeita em  $x_*$  quando os gradientes das restrições de igualdade são linearmente independentes e se existir um vetor  $d \in \mathbb{R}^n$  tal que

$$\nabla h_i(x_*)^T d = 0$$

e

$$\nabla g_j(x_*)^T d < 0$$

para todos  $i \in \varepsilon$  e  $j \in I(x_*)$

As restrições do exemplo 10 cumprem *MCFQ* no ponto  $x_* = 0$ , pois o vetor

$$d = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

satisfaz  $\nabla h_i(x_*)^T d < 0$ ,  $i = 1, 2, 3$ .

Também é importante notar que a condição de *MCFQ*, apesar de ser uma hipótese mais fraca, não é necessária para termos *KKT*, conforme ocorre no seguinte exemplo.

**Exemplo 11.** Considere o problema

$$\begin{array}{ll} \min & f(x) = x_1 \\ & g_1(x) = -x_1^3 + x_2 \leq 0 \\ \text{sujeito a} & g_2(x) = -x_1^3 - x_2 \leq 0 \\ & g_3(x) = -x_1 \leq 0 \end{array}$$

O ponto  $x_* = 0$  é o minimizador e satisfaz *KKT*, mas não cumpre a condição *MFCQ*. De fato, as três restrições são ativas em  $x_* = 0$  e

$$\nabla g_1(x_*) = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \text{ e } \nabla g_2(x_*) = \begin{pmatrix} 0 \\ -1 \end{pmatrix} \text{ e } \nabla g_3(x_*) = \begin{pmatrix} -1 \\ 0 \end{pmatrix}.$$

Note que não existe um vetor  $d \in \mathbb{R}^2$  tal que  $\nabla h_i(x_*)^T d < 0$  para  $i = 1, 2, 3$ . Além disso, temos *KKT*, pois

$$-\nabla f(x_*) = \begin{pmatrix} -1 \\ 0 \end{pmatrix} = \nabla g_3(x_*)$$

## 4 MÉTODOS DE OTIMIZAÇÃO APLICADOS EM UMA EQUAÇÃO DIFERENCIAL COM MÚLTIPLOS PONTOS DE FRONTEIRA

Neste capítulo analisamos algumas perspectivas na determinação de solução numérica para uma equação diferencial de segunda ordem com múltiplos pontos de fronteira definida como:

$$\begin{cases} u'' + q(t)f(t, u, u') = 0, \\ u(0) = 0 \quad u(1) = g(u(\eta_1), u(\eta_2), \dots, u(\eta_{m-2})) \end{cases} \quad (4.1)$$

onde  $g : R^{m-2} \rightarrow R$ ,  $f : [0, 1] \times R \times R \rightarrow R$  e  $q : R_+ \rightarrow R$  são contínuas e, possivelmente, não lineares,  $\eta_1, \dots, \eta_{m-2} \in (0, 1)$ .

Este problema é conhecido na literatura como problema de segunda ordem com múltiplos pontos de fronteira, ou simplesmente, m-pontos. Os primeiros resultados de existência de solução foram apresentados por [23]. Segundo [27] variações deste problema surgem no contexto de modelos de fluxo elástico e viscoelástico, aplicações relacionadas com designe de pontes são apresentadas por [28].

Devido à importância dessa classe de problemas em diversas aplicações, muitos autores têm desenvolvido estudos considerando variações e generalizações. A maior parte desses estudos são relacionados à existência de solução, recomendamos as referências [15]; [7]; [20] e [26]. Diversos trabalhos despontaram tratando da obtenção de solução numérica, recomendamos ([5], [19] e [21]), os resultados numéricos geralmente não caminham como a existência de soluções e na sua maioria consideram apenas métodos baseados no teorema de ponto fixo de Banach, uma exceção é o trabalho de [6], no qual os autores exploram o uso de métodos de otimização na determinação de solução numérica, neste trabalho a equação diferencial de (4.1) é discretizada e esta da origem ao conjunto de restrições para o problema de programação não linear e utiliza-se como função objetivo uma variação da condição  $u(1) = g(u(\eta_1), u(\eta_2), \dots, u(\eta_{m-2}))$ , neste capítulo trabalhamos com a equação discretizada e aplicamos o método de Gauss-Newton também utilizamos o fato de que esta classe de problemas pode ser representada por uma equação integral para utilizar um método de programação não linear com restrições. Este trabalho motiva explorarmos alternativas que podem ser aplicadas a outras equações.

### 4.1 MÉTODOS NUMÉRICOS

Nesta seção apresentamos duas alternativas para determinação de solução numérica para o problema (1), a primeira alternativa se baseia na resolução do sistema não linear obtido com a discretização do problema, na segunda alternativa a equação discretizada forma o conjunto de restrições e utilizamos uma função objetivo baseada na representação integral do problema.

### 4.1.1 Algoritmo 1

Em nossa primeira alternativa, basicamente utilizamos o método de Gauss-Newton recomendamos [18], combinando com interpolação spline. Vamos considerar  $\{t_j\}$ ,  $j = 0, 1, \dots, n$  uma partição de  $[0, 1]$ , com  $t_{j+1} - t_j = h = \frac{1}{n}$  para todo  $j = 0, 1, \dots, n - 1$ , denotaremos por  $u_j$  uma aproximação para  $u(t_j)$ . Utilizando diferenças finitas obtemos uma discretização para a equação  $u'' = f(t, u, u')$ . Definimos a seguir o sistema não linear  $r(u) = 0$ :

$$r(u) = \begin{cases} \frac{u_{j+1} - 2u_j + u_{j-1}}{h^2} + f\left(t_j, u_j, \frac{u_{j+1} - u_{j-1}}{2h}\right) = 0 \\ u_1 - g(u(\eta_1), u(\eta_2), \dots, u(\eta_{m-2})) = 0 \end{cases} \quad (4.2)$$

Um algoritmo para resolver o problema (4.1) é apresentado a seguir.

#### ALGORITMO 1

Passo 1 Definir uniformemente a malha espaçada  $\{t_j\}$ ,  $j = 1, \dots, n$ ;

Passo 2 Escolher a aproximação inicial  $u_j^0 = u^0(t_j)$ ;

Passo 3 Discretizar o problema (1) por diferença finita, e computar para  $j = 2, \dots, n - 1$

$$u''^k(t_j) = \frac{u^k(t_{j+1}) - 2u^k(t_j) + u^k(t_{j-1}))}{h^2}, \quad u'^k(t_j) = \frac{u^k(t_{j+1}) - u^k(t_{j-1}))}{2h}$$

Calcular  $u^k(\eta)$ ,  $\eta = (\eta_1, \dots, \eta_{m-2})$  usando interpolação por Spline Cúbica.

Passo 4 Para  $k = 1, 2, 3, \dots$  (Gauss-Newton para o sistema  $r(u) = 0$ ,  $r$  foi definida em (4.1)).

- a) Compute  $r^k = (r_1, r_2, \dots, r_n)^T$  e  $A_k = (a_{ij})_{n \times n}$ ,  $r_i = r_i(u^k)$ ,  $a_{ij} = \nabla r_i(u^k)$
- b) Encontre  $d_k$  tal que  $(A_k^T A_k) d_k = -A_k^T r^k$
- c) Determine  $\alpha_k$  tal que a condição de Armijo seja satisfeita
- d) Compute  $u^{k+1} = u^k + \alpha_k d_k$

Passo 5 Teste a convergência.

### 4.1.2 Algoritmo 2

Na segunda alternativa, basicamente utilizamos um método de programação não linear para problemas com restrições. Assim como no algoritmo 1 consideramos  $\{t_j\}$ ,  $j = 0, 1, \dots, n$  uma partição de  $[0, 1]$ , com  $t_{j+1} - t_j = h = \frac{1}{n}$  para todo  $j = 0, 1, \dots, n - 1$ , denotaremos por  $u_j$  uma aproximação para  $u(t_j)$ . Consideremos como no algoritmo 1 o sistema não linear  $r(u) = 0$  para compor o conjunto das restrições do problema. Definimos uma função objetivo

que explora algumas propriedades do problema (4.1) utilizado nos métodos numéricos baseados em ponto fixo.

O problema definido em (4.1) pode ser modelado como uma equação integral, para isto basta observar que as soluções do problema (4.1) são pontos fixos do operador

$$(Tu)(x) = \int_0^1 G(x,t)f(t, u(t), u'(t))dt + g(u(\eta_1), \dots, u(\eta_{m-2}))x \quad (4.3)$$

onde  $G$  é uma função de Green dada por

$$G(x,t) = \begin{cases} t(1-x), & t \leq x \\ x(1-t), & x \leq t \end{cases} \quad (4.4)$$

Métodos numéricos baseados em ponto fixo, definem uma aproximação inicial para a solução discretizada de (4.1),  $u^0$ , então a sequência do método fica determinada como  $u^{k+1} = Tu^k$ , a convergência desta classe de métodos se baseia no Teorema de Ponto Fixo de Banach, ou seja, converge se  $T$  é uma contração em uma vizinhança da solução.

Observe que determinar  $u$  que satisfaz  $T(u) = u$  é equivalente a determinar  $u$  que minimiza  $\|T(u) - u\|_2^2$ , deste modo uma solução de (4.1) é uma solução do problema de programação não linear abaixo:

$$\begin{aligned} \min \quad & \|T(u) - u\|_2^2 \\ \text{sujeito a} \quad & \frac{u_{j+1} - 2u_j + u_{j-1}}{h^2} + f\left(t_j, u_j, \frac{u_{j+1} - u_{j-1}}{2h}\right) = 0 \\ & u_1 - g(u(\eta_1), u(\eta_2), \dots, u(\eta_{m-2})) = 0 \\ & Q(u) = 0 \end{aligned} \quad (5)$$

onde a restrição  $Q(u) = 0$  representa a possibilidade de inclusão de análise qualitativa do problema, ou seja, esta restrição permite a inclusão de restrições com relação a norma, positividade, concavidade da solução a ser determinada, caso tenhamos esta informação. Definimos como ALGORITMO 2 a resolução do problema de programação não linear (5) pela função *fmincon* do MatLab ([www.mathworks.com](http://www.mathworks.com)).

## ALGORITMO 2

Passo 1 Definir uniformemente a malha espaçada  $\{t_j\}, j = 1, \dots, n$ ;

Passo 2 Escolher a aproximação inicial  $u_j^0 = u^0(t_j)$ ;

Passo 3 Discretizar o problema (4.1) por diferença finita, e computar para  $j = 2, \dots, n - 1$

$$u''^k(t_j) = \frac{u^k(t_{j+1}) - 2u^k(t_j) + u^k(t_{j-1}))}{h^2}, \quad u'^k(t_j) = \frac{u^k(t_{j+1}) - u^k(t_{j-1}))}{2h}$$

Calcular  $u^k(\eta), \eta = (\eta_1, \dots, \eta_{m-2})$  usando interpolação por Spline Cúbica.

Passo 4 Aplique a função *fmincon* para o problema,

$$\begin{aligned} \min \quad & \|T(u) - u\|_2^2 \\ \text{sujeito a} \quad & \frac{u_{j+1} - 2u_j + u_{j-1}}{h^2} + f\left(t_j, u_j, \frac{u_{j+1} - u_{j-1}}{2h}\right) = 0 \\ & u_1 - g(u(\eta_1), u(\eta_2), \dots, u(\eta_{m-2})) = 0 \\ & Q(u) = 0 \end{aligned}$$

calcule as integrais do operador  $T$  com o método dos trapézios e  $u^k(\eta)$  utilizando Spline Cúbica.

Passo 5 Teste a convergência.

Uma motivação para definirmos os Algoritmo 1 e 2 consiste no fato de que métodos numéricos baseados em ponto fixo não permitem a inclusão de análise qualitativa da solução, além disso estes métodos apresentam forte tendência de convergir para a solução de menor norma, quando o problema possui múltiplas soluções, mesmo utilizando variedade de aproximações iniciais.

## 4.2 TESTES NUMÉRICOS

### 4.2.1 Testes realizados com 1° Algoritmo

Em seguida consta dez exemplos os quais realizamos testes. Estes exemplos foram retirados dos seguintes artigos: [12], [14], [11],[13]. Dos exemplares testados oito possuem soluções analíticas os demais possuem apenas limitações para existência de solução.

- Exemplo 1 retirado do artigo [12].

$$f(t, u, u') = -u - 2u'$$

$$g(s_1, s_2, s_3) = 64s_1^4 + 9s_2^3 + s_3^2 + \frac{e^{-1}}{6}$$

$$\eta = (\eta_1, \eta_2, \eta_3) = \left(\frac{1}{4}, \frac{1}{3}, \frac{1}{2}\right)$$

A solução numérica para o exemplo 1 foi obtida em cinco iterações com um tempo de 2.42s. As soluções analíticas para este problema foram  $u(x) = xe^{-x}$  e  $\bar{u}(x) = 0.176534437xe^{-x}$ .

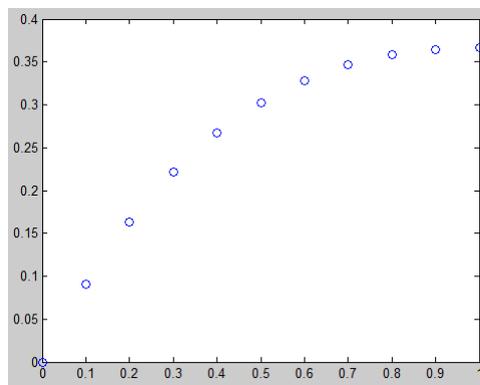


Figura 4.1: Solução exata exemplo 1

$$r(u^k) = \begin{cases} u''^k(t_j) + q(t_j)f(t_j, u^k(t_j), u'^k(t_j)) = 0; & j = 2, \dots, n-1 \\ u^k(1) - g(u^k(\eta_1), u^k(\eta_2), \dots, u^k(\eta_{m-2})) = 0 \end{cases} \quad (4.5)$$

$$r(u) = \begin{cases} \frac{u(t_{j+1}) - 2u(t_j) + u(t_{j-1}))}{h^2} + f\left(t_j, u(t_j), \frac{u(t_{j+1}) - u(t_{j-1}))}{2h}\right) = 0 \\ u(1) - g(u(\eta_1), u(\eta_2), u(\eta_3)) = 0 \end{cases} \quad (4.6)$$

$$r(u) = \begin{cases} u(t_{j+1}) - 2u(t_j) + u(t_{j-1}) + h^2 \left[ -u(t_j) - \frac{u(t_{j+1}) - u(t_{j-1}))}{h} \right] = 0 \\ u(1) - 64(u(0, 25))^4 - 9(u(0, 33))^3 - (u(0, 5))^2 - \frac{e^{-1}}{6} = 0 \end{cases} \quad (4.7)$$

$$r(u) = \begin{cases} u(t_{j+1}) - 2u(t_j) + u(t_{j-1}) - h^2u(t_j) - hu(t_{j+1}) + hu(t_{j-1}) = 0 \\ u(1) - 64(u(0, 25))^4 - 9(u(0, 33))^3 - (u(0, 5))^2 - \frac{e^{-1}}{6} = 0 \end{cases} \quad (4.8)$$

Como citamos anteriormente, o método de Gauss Newton exige apenas a primeira derivada de  $r(u)$ , ou seja a matriz Jacobiana . Somente para visualizarmos vamos escrever a  $r(u) = (r_2(u), r_3(u), r_4(u), r_5(u), r_6(u))$  na forma matricial e escrever a sua Jacobiana considerando  $t_j = 2, \dots, 6$ . Substituindo os valores de  $(r_2(u), r_3(u), \dots, r_6(u))$  obtemos

$$\begin{pmatrix} r_2(u) \\ r_3(u) \\ r_4(u) \\ r_5(u) \\ r_6(u) \end{pmatrix} \quad (4.9)$$

=

$$\begin{pmatrix} \frac{u(t_3) - 2u(t_2) + u(t_1) - h^2u(t_2) - hu(t_3) + hu(t_1)}{h^2} \\ \frac{u(t_4) - 2u(t_3) + u(t_2) - h^2u(t_3) - hu(t_4) + hu(t_2)}{h^2} \\ \frac{u(t_5) - 2u(t_4) + u(t_3) - h^2u(t_4) - hu(t_5) + hu(t_3)}{h^2} \\ \frac{u(t_6) - 2u(t_5) + u(t_4) - h^2u(t_5) - hu(t_6) + hu(t_4)}{h^2} \\ \frac{u(t_7) - 2u(t_6) + u(t_5) - h^2u(t_6) - hu(t_7) + hu(t_5)}{h^2} \end{pmatrix} \quad (4.10)$$

$$= \begin{pmatrix} \frac{1+h}{h^2} & \frac{-2-h^2}{h^2} & \frac{1-h}{h^2} & 0 & 0 & 0 & 0 \\ 0 & \frac{1+h}{h^2} & \frac{-2-h^2}{h^2} & \frac{1-h}{h^2} & 0 & 0 & 0 \\ 0 & 0 & \frac{1+h}{h^2} & \frac{-2-h^2}{h^2} & \frac{1-h}{h^2} & 0 & 0 \\ 0 & 0 & 0 & \frac{h+1}{h^2} & \frac{-2-h^2}{h^2} & \frac{1-h}{h^2} & 0 \\ 0 & 0 & 0 & 0 & \frac{1+h}{h^2} & \frac{-2-h^2}{h^2} & \frac{1-h}{h^2} \end{pmatrix} \quad (4.11)$$

considerando  $h = 0.1$  obtemos

$$= \begin{pmatrix} 110 & -201 & 90 & 0 & 0 & 0 & 0 \\ 0 & 110 & -201 & 90 & 0 & 0 & 0 \\ 0 & 0 & 110 & -201 & 90 & 0 & 0 \\ 0 & 0 & 0 & 110 & -201 & 90 & 0 \\ 0 & 0 & 0 & 0 & 110 & -201 & 90 \end{pmatrix} \quad (4.12)$$

- Exemplo 2 retirado do artigo [12].

$$u'' = -4u - 2tu'$$

$$g(s) = 15s^4 + \frac{e^{-1}}{16}$$

$$\eta = 0.5$$

A solução numérica para o exemplo 2 foi obtida em cinco iterações com um tempo de 2.11s. As soluções analíticas encontradas para este problema foram  $u(x) = xe^{-(x^2)}$  e  $\bar{u}(x) = 0.062514318227981xe^{-(x^2)}$ .

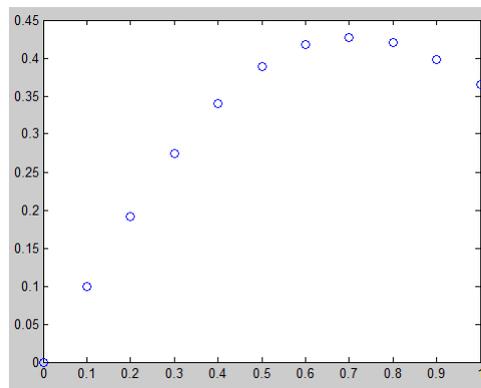


Figura 4.2: Solução exata exemplo 2

$$r(u) = \begin{cases} \frac{u(t_{j+1}) - 2u(t_j) + u(t_{j-1}))}{h^2} + f\left(t_j, u(t_j), \frac{u(t_{j+1}) - u(t_{j-1}))}{2h}\right) = 0 \\ u(1) - g(u(\eta_1), u(\eta_2), \dots, u(\eta_{m-2})) = 0 \end{cases} \quad (4.13)$$

Substituindo o valor da  $f$  e da  $g$  obtemos :

$$r(u) = \begin{cases} u(t_{j+1}) - 2u(t_j) + u(t_{j-1}) + h^2 \left[ -4u(t_j) - t_j \frac{u(t_{j+1}) - u(t_{j-1}))}{h} \right] = 0 \\ u(1) - 15(u(0.5))^4 - \frac{e^{-1}}{16} = 0 \end{cases} \quad (4.14)$$

$$r(u) = \begin{cases} u(t_{j+1}) - 2u(t_j) + u(t_{j-1}) - 4h^2u(t_j) - t_jhu(t_{j+1}) + t_jhu(t_{j-1}) = 0 \\ u(1) - 15(u(0.5))^4 - \frac{e^{-1}}{16} = 0 \end{cases} \quad (4.15)$$

- Exemplo 3 retirado do artigo [14].

$$f(t, u, u') = (u')^2 - u^2 + 2tu' + t^2 - 2t + 1$$

$$g(y) = 2y$$

$$\eta = 0.5$$

$$q(t) = -1$$

Considerando a identidade como ponto inicial a solução numérica para o exemplo 3 foi obtida em seis iterações com um tempo de 6.21s. A solução analítica foi  $u(x) = x$ .

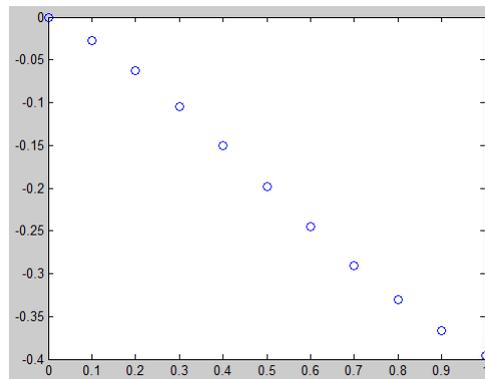


Figura 4.3: Solução exata exemplo 3

- Exemplo 4 retirado do artigo [11].

$$f(t, u, u') = (u')^2 - u^2 + 2tu' + t^2 - 2t + 1$$

$$g(y) = \frac{4}{3}y$$

$$\eta = \frac{3}{4}$$

$$q(t) = -1$$

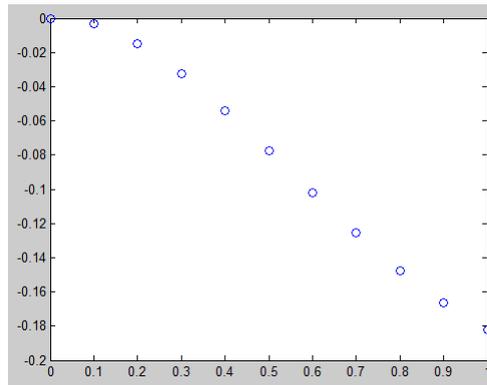


Figura 4.4: Solução exata exemplo 4

Considerando o ponto nulo a solução numérica para o exemplo 4 foi obtida em quatro iterações com um tempo de 7.06s. A solução analítica foi  $u(x) = x$ .

- Exemplo 5 retirado do artigo [14].

$$f(t, u, u') = (u')^2 - u^2 + 2tu' + t^2 - 2t + 1$$

$$g(y) = y(1) + y(2) + y(3) + (y(4))^2$$

$$\eta = [1/6, 1/4, 1/3, 1/2]$$

$$q(t) = -1$$

Considerando o ponto nulo a solução numérica para o exemplo 5 foi obtida em quatro iterações com um tempo de 6.12s. A solução analítica foi  $u(x) = x$ .

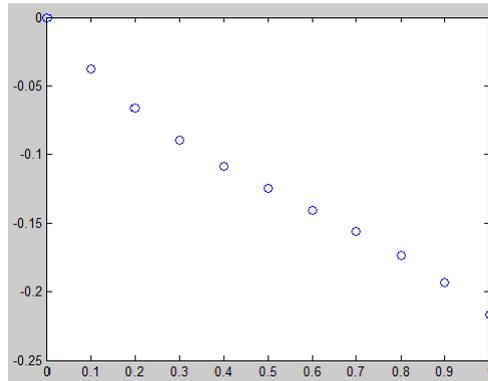


Figura 4.5: Solução exata exemplo 5

- Exemplo 6 retirado do artigo [13].

$$f(t, u, v) = u(u^2 + u^{2'})$$

$$g(y) = 0.591470984807897 + (u(\pi/6))^2$$

$$\eta = [\pi/6]$$

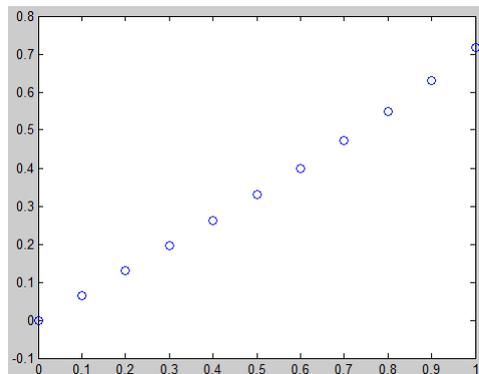


Figura 4.6: Solução exata exemplo 6

Considerando zero como ponto inicial a solução numérica para o exemplo 6 foi obtida em cinco iterações com um tempo de 6.98s. A solução analítica encontrada para este problema foi  $u(x) = \text{sen}(x)$ .

$$r(u) = \begin{cases} \frac{u(t_{j+1}) - 2u(t_j) + u(t_{j-1}))}{h^2} + f\left(t_j, u(t_j), \frac{u(t_{j+1}) - u(t_{j-1}))}{2h}\right) = 0 \\ u(1) - g(u(\eta_1), u(\eta_2), \dots, u(\eta_{m-2})) = 0 \end{cases} \quad (4.16)$$

Substituindo o valor da  $f$  da  $g$  da  $u'$  e da  $u''$  obtemos:

$$r(u) = \begin{cases} \frac{u(t_{j+1}) - 2u(t_j) + u(t_{j-1}))}{h^2} + u(t_j) * \left( u(t_j)^2 + \frac{u^2(t_{j+1}) - u^2(t_{j-1}))}{4h^2} \right) \\ u(1) - g(u(\eta_1), u(\eta_2), \dots, u(\eta_{m-2})) = 0 \end{cases} \quad (4.17)$$

$$r(u) = \begin{cases} \frac{4u(t_{j+1}) - 8u(t_j) + 4u(t_{j-1}))}{4h^2} + u(t_j) * \left( \frac{4h^2 u(t_j)^2 + u^2(t_{j+1}) - u^2(t_{j-1}))}{4h^2} \right) \\ u(1) - g(u(\eta_1), u(\eta_2), \dots, u(\eta_{m-2})) = 0 \end{cases} \quad (4.18)$$

- Exemplo 7 retirado do artigo [13].

$$f(t, u, v) = u(u^2 + v^2)$$

$$g(y) = 0.420735492403948 + 0.392943630388474(u(1/3)) + 0.472478473157369(u(2/3))$$

$$\eta = [1/3, 2/3]$$

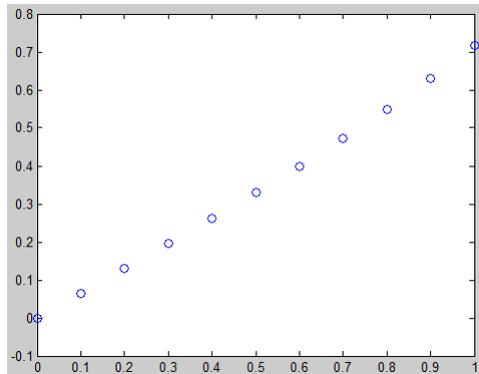


Figura 4.7: Solução exata exemplo 7

Considerando a identidade como ponto inicial a solução numérica para o exemplo 7 foi obtida em quatro iterações com um tempo de 8.10s. A solução analítica encontrada para este problema foi  $u(x) = \text{sen}(x)$ .

### Exemplos sem respostas analíticas

- Exemplo 8 retirado do artigo [12].

$$f(t, u, v) = \begin{cases} t + \frac{24}{27}u^3 + \left(\frac{v}{80}\right)^2, & 0 \leq u \leq 3 \\ t + 24 + u - 3 + \left(\frac{v}{80}\right)^2, & 3 \leq u \end{cases}$$

$$g(y) = \frac{1}{3} + \frac{1}{4k} \sum_{i=1}^k y_i$$

$$q(t) = 1$$

$$\eta = \left[ \frac{1}{10}, \frac{1}{5}, \frac{2}{3}, \frac{3}{4} \right]$$

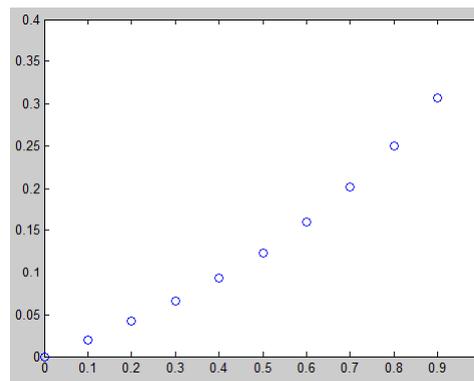


Figura 4.8: Solução exata exemplo 8

Considerando a nulo como ponto inicial a solução numérica para o exemplo 8 foi obtida em quatro iterações com um tempo de 9.6s.

- Exemplo 9 retirado do artigo [14].

$$f(t, u, v) = 2t + \frac{3}{10}u + \frac{1}{20}v^2$$

$$q(t) = e^t$$

$$g(y(1), y(2), y(3)) = \frac{1}{12} (y(1) + y(2) + y(3))$$

$$\eta = [1/3, 1/2, 3/4]$$

Considerando o ponto nulo como ponto inicial a solução numérica para o exemplo 9 foi obtida em três iterações com um tempo de 7.05s.

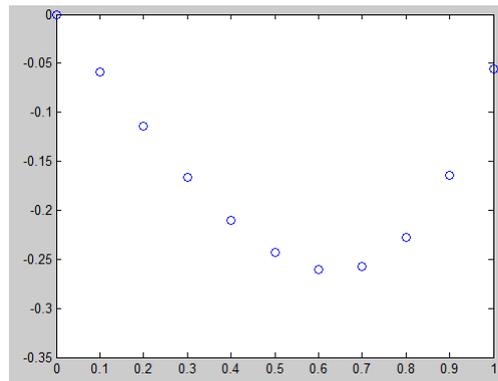


Figura 4.9: Solução exata exemplo 9

- Exemplo 10 retirado do artigo [14].

$$f(t, u, v) = \frac{5}{3}t + \frac{1}{20}u|v| + \frac{1}{3}|v| \sin^2(u)$$

$$q(t) = e^t$$

$$g(y(1), y(2), y(3)) = \sin(y(1), y(2), y(3)) + \frac{1}{20}y(2) + \frac{1}{10}$$

$$\eta = [1/3, 1/2, 3/4]$$

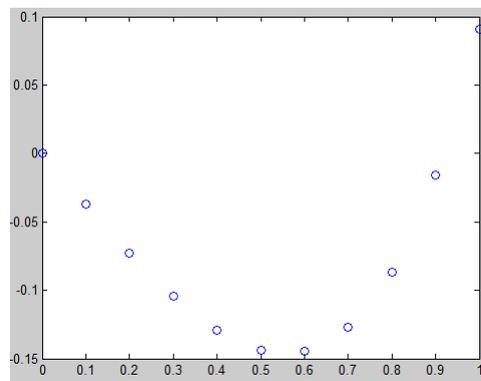


Figura 4.10: Solução exata exemplo 10

Considerando a identidade como ponto inicial a solução numérica para o exemplo 10 foi obtida em cinco iterações com um tempo de 7.55s.

### 4.2.2 Testes realizados com 2º Algoritmo

Repetimos os testes utilizando o 2º algoritmo, a seguir apresentamos os detalhes:

- Exemplo 1

$$f(t, u, u') = -u - 2u'$$

$$g(s_1, s_2, s_3) = 64s_1^4 + 9s_2^3 + s_3^2 + \frac{e^{-1}}{6}$$

$$\eta = (\eta_1, \eta_2, \eta_3) = \left(\frac{1}{4}, \frac{1}{3}, \frac{1}{2}\right)$$

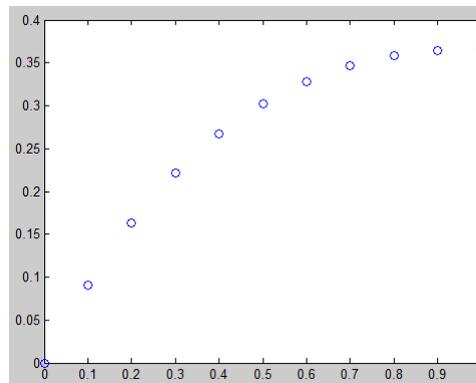


Figura 4.11: Solução exata exemplo 1

Considerando a identidade como ponto inicial a solução numérica para o exemplo 1 foi obtida em seis iterações com um tempo de 1.28s. As soluções analíticas são  $u(x) = xe^{-x}$  e  $\bar{u}(x) = 0.176534437xe^{-x}$ .

- Exemplo 2

$$u'' = -4u - 2tu'$$

$$g(s) = 15s^4 + \frac{e^{-1}}{16}$$

$$\eta = 0.5$$

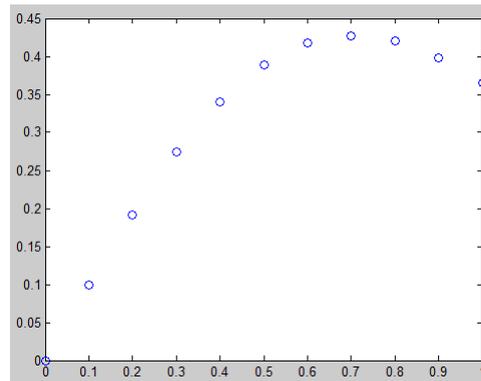


Figura 4.12: Solução exata exemplo 2

Considerando a identidade como ponto inicial a solução numérica para o exemplo 2 foi obtida em seis iterações com um tempo de 1.66s. As soluções analíticas são  $u(x) = xe^{-x}$  e  $\bar{u}(x) = 0.0062514318227981xe^{-x}$ .

- Exemplo 3 retirado do artigo [14].

$$f(t, u, u') = (u')^2 - u^2 + 2tu' + t^2 - 2t + 1$$

$$g(y) = 2y$$

$$\eta = 0.5$$

$$q(t) = -1$$

Considerando o ponto nulo como ponto inicial a solução numérica para o exemplo 3 foi obtida em sete iterações com um tempo de 2.15s. A solução analítica é  $u(x) = x$ .

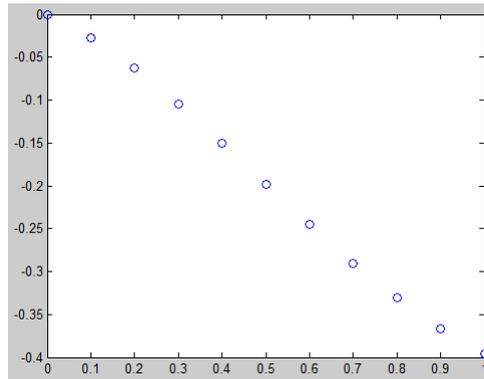


Figura 4.13: Solução exata exemplo 3

- Exemplo 4 retirado do artigo [11].

$$f(t, u, u') = (u')^2 - u^2 + 2tu' + t^2 - 2t + 1$$

$$g(y) = \frac{4}{3}y$$

$$\eta = \frac{3}{4}$$

$$q(t) = -1$$

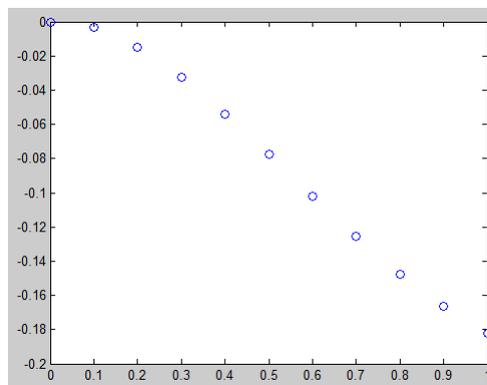


Figura 4.14: Solução exata exemplo 4

Considerando o zero como ponto inicial a solução numérica para o exemplo 4 foi obtida em cinco iterações com um tempo de 2.44s. A solução analítica é  $u(x) = x$ .

- Exemplo 5 retirado do artigo [14].

$$f(t, u, u') = (u')^2 - u^2 + 2tu' + t^2 - 2t + 1$$

$$g(y) = y(1) + y(2) + y(3) + (y(4))^2$$

$$\eta = [1/6, 1/4, 1/3, 1/2]$$

$$q(t) = -1$$

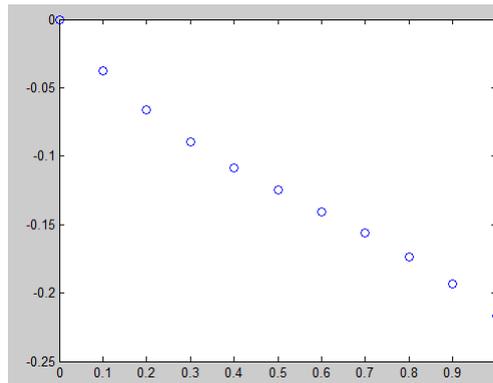


Figura 4.15: Solução exata exemplo 5

Considerando a identidade como ponto inicial a solução numérica para o exemplo 5 foi obtida em seis iterações com um tempo de 2.22s. A solução analítica é  $u(x) = x$ .

- Exemplo 6 retirado do artigo [13].

$$f(t, u, v) = u * (u^2 + u^{2'})$$

$$g(y) = 0.591470984807897 + (u(\pi/6))^2$$

$$\eta = [\pi/6]$$

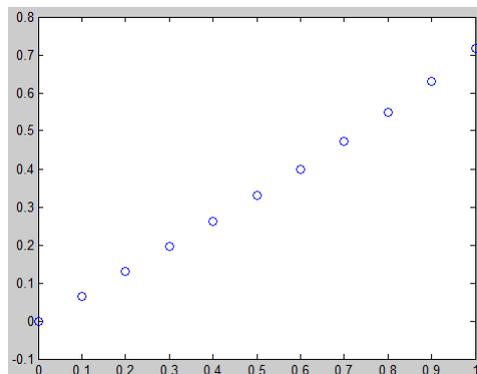


Figura 4.16: Solução exata exemplo 6

Considerando a identidade como ponto inicial a solução numérica para o exemplo 6 foi obtida em sete iterações com um tempo de 1.79s. A solução analítica é  $u(x) = \text{sen}(x)$ .

- Exemplo 7 retirado do artigo [13].

$$f(t, u, v) = u(u^2 + v^2)$$

$$g(y) = 0.420735492403948 + 0.392943630388474(u(1/3)) + 0.472478473157369(u(2/3))$$

$$\eta = [1/3, 2/3]$$

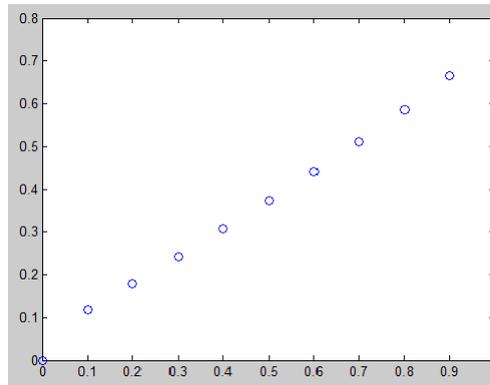


Figura 4.17: Solução exata exemplo 7

Considerando a identidade como ponto inicial a solução numérica para o exemplo 7 foi obtida em seis iterações com um tempo de 2.59s. A solução analítica é  $u(x) = \text{sen}(x)$ .

### Exemplos sem respostas analíticas

- Exemplo 8

$$f(t, u, v) = \begin{cases} t + \frac{24}{27}u^3 + \left(\frac{v}{80}\right)^2, & 0 \leq u \leq 3 \\ t + 24 + u - 3 + \left(\frac{v}{80}\right)^2, & 3 \leq u \end{cases}$$

$$g(y) = \frac{1}{3} + \frac{1}{4k} \sum_{i=1}^k y_i$$

$$q(t) = 1$$

$$\eta = \left[ \frac{1}{10}, \frac{1}{5}, \frac{2}{3}, \frac{3}{4} \right]$$

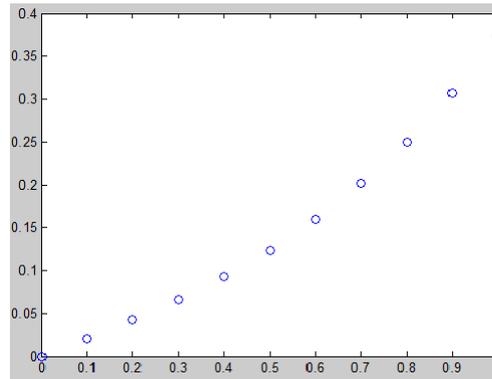


Figura 4.18: Solução exata exemplo 8

Considerando a identidade como ponto inicial a solução numérica para o exemplo 8 foi obtida em quatro iterações com um tempo de 2.18s.

- Exemplo 9 retirado do artigo [14]

$$f(t, u, v) = 2t + \frac{3}{10}u + \frac{1}{20}v^2$$

$$q(t) = e^t$$

$$g(y(1), y(2), y(3)) = \frac{1}{12}(y(1) + y(2) + y(3))$$

$$\eta = [1/3, 1/2, 3/4]$$

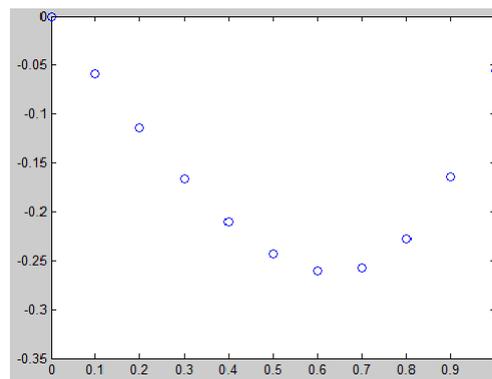


Figura 4.19: Solução exata exemplo 9

Considerando a identidade como ponto inicial a solução numérica para o exemplo 9 foi obtida em cinco iterações com um tempo de 2.42s.

- Exemplo 10 retirado do artigo [14]

$$f(t, u, v) = \frac{5}{3}t + \frac{1}{20}u|v| + \frac{1}{3}|v| \sin^2(u)$$

$$q(t) = e^t$$

$$g(y(1), y(2), y(3)) = \sin(y(1), y(2), y(3)) + \frac{1}{20}y(2) + \frac{1}{10}$$

$$\eta = [1/3, 1/2, 3/4]$$

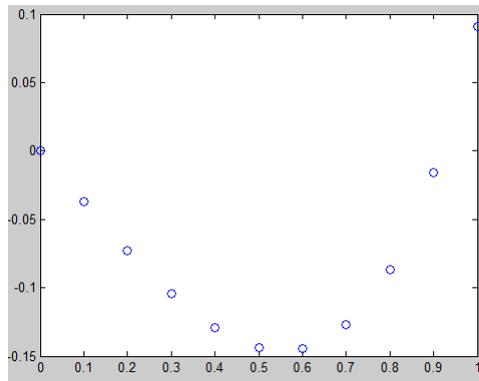


Figura 4.20: Solução exata exemplo 10

Considerando a identidade como ponto inicial a solução numérica para o exemplo 10 foi obtida em seis iterações com um tempo de 1.69s.

### 4.2.3 Comparativo

Com intuito de comparar os Algoritmo 1 e 2, realizamos testes nos quais não incluímos a restrições qualitativas no Algoritmo 2. Utilizamos a função *fmincom* do MatLab ([www.mathworks.com](http://www.mathworks.com)) para aplicar tanto o Algoritmo 1 quanto o Algoritmo 2, especificamente no caso do Algoritmo 1 consideramos uma função constante como função objetivo na função *fmincon* e implementamos  $r(u) = 0$  como restrição de igualdade, deste modo consideramos o tempo gasto por cada algoritmo nos problemas como um parâmetro de comparação.

Apresentamos resultados após realizar testes com dez problemas utilizando os algoritmos 1 e 2. Primeiramente detalharemos o exemplo 8, no qual as funções componentes do problema (4.1) são apresentadas a seguir:

$$f(t, u, v) = \begin{cases} t + \frac{24}{27}u^3 + (\frac{v}{80})^2, & 0 \leq u \leq 3 \\ t + 24 + u - 3 + (\frac{v}{80})^2, & 3 \leq u \end{cases}$$

e

$$g(y) = \frac{1}{3} + \frac{1}{4k} \sum_{i=1}^k y_i.$$

$$\eta = \left( \frac{1}{10}, \frac{1}{5}, \frac{2}{3}, \frac{3}{4} \right)$$

Pode-se mostrar utilizando o teorema de Avery-Peterson, recomendamos [3], que o problema acima possui no mínimo três soluções no espaço de Banach  $E = C^1[0, 1]$  das funções continuamente diferenciáveis em  $[0, 1]$  com norma

$$\|u\|_E = \max\{\|u\|_\infty, \|u'\|_\infty\}.$$

Segundo o teorema podemos ainda caracterizar as soluções como concavas, positivas e com  $\|u\|_E \leq 80$ . Deste modo podemos incluir no Algoritmo 2 função  $Q$  afim de explorar estas características qualitativas do problema, deste modo obtemos as seguintes restrições adicionais

$$\begin{aligned} -u_j &\leq 0, \forall j = 1, \dots, n \\ u_j &\leq 80, \forall j = 1, \dots, n \end{aligned}$$

Aplicamos os Algoritmos 1 e 2, utilizamos dez aproximações iniciais diferentes, definidas pela equação

$$u^0(x) = x(n_r + 1 - x),$$

onde  $n_r$  é um número randômico no intervalo  $[0, 10]$ .

Naturalmente, este processo pode retornar múltiplas respostas. Assim precisamos estabelecer um critério afim de comparar soluções para estabelecer quais seriam distintas. Note que a amplitude das soluções podem ser diferentes. Neste sentido dizemos que soluções  $u^*$  e  $u^{**}$  são

equivalentes se

$$\|u^* - u^{**}\| \leq \max\{10^{-8}, 10^{-6} \min\{\|u^*\|, \|u^{**}\|\}\}.$$

Os resultados obtidos pelos algoritmos 1 e 2 foram equivalentes, utilizando o critério acima para comparar as soluções obtidas, concluímos que pelo menos cinco soluções são diferentes (ver Figura 4.21).

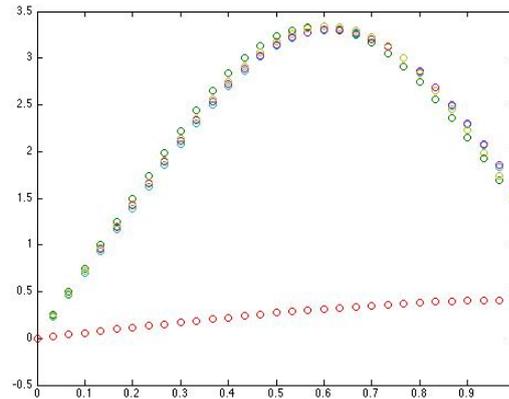


Figura 4.21: Soluções numéricas obtidas.

Podemos observar que as soluções encontradas são possivelmente concavas, positivas e com  $\|u\|_E \leq 80$ , como era esperado.

Para este problema também aplicamos o Algoritmo 2 do artigo [21], baseado no teorema de Ponto Fixo de Banach, aplicamos as mesmas condições anteriores utilizando 10 aproximações iniciais diferentes. O método convergiu apenas para uma solução distinta, possivelmente a de menor norma (ver Figura 4.21).

Ambos os algoritmos convergiram nos onze problemas testados, esperávamos que o Algoritmo 2 fosse consideravelmente mais lento, pois o custo do cálculo de  $\|Tu - u\|_2^2$  deveria implicar em um número maior de operações, porém o tempo gasto entre os algoritmos foi equilibrado como podemos observar na Tabela 4.1, isto se deve principalmente ao fato de que o Algoritmo 2 consumiu um número menor de iterações internas.

Tabela 4.1: Testes comparativos

	$t$ (1 ° Problema )	$t$ (2° Problema )	ponto inicial
ex1	2,42 s	2,34 s	identidade
ex2	2,11 s	2,76 s	identidade
ex4	1,75 s	1,67 s	nulo
ex5	1,59 s	1,95 s	nulo
ex6	2,16 s	2,06 s	nulo
ex7	2,07 s	1,90 s	identidade
ex8	2,49 s	2,33 s	identidade
ex9	1,67 s	1,62 s	nulo
ex10	1,79 s	1,79 s	nulo
ex11	1,82 s	2,11 s	identidade

## 5 CONCLUSÃO

Neste trabalho realizamos uma breve revisão bibliográfica sobre otimização não linear a fim de aplicarmos métodos de otimização não linear para resolver o problema 1.1. Iniciamos a revisão bibliográfica no capítulo dois, no qual focamos em problemas irrestritos. O primeiro problema foi maximizar, minimizar uma função sobre um conjunto  $\Omega$  que é o próprio  $R^n$ . Para este problema apresentamos quatro algoritmos os quais diferem pela direção  $d_k$ . As direções que consideramos foram a do gradiente 2.21, a de newton 2.23 a de newton corrigido 2.25 e do método BFGS dada pela equação 2.29. Com os testes que realizamos, apesar de serem poucos exemplos, chegamos a seguinte conclusão: O método baseado na direção do gradiente apesar de não exigir o cálculo da matriz hessiana mostrou ser mais lento e o número de iterações foi maior comparado com os outros métodos. O método baseado na direção de Newton apresentou melhor desempenho, isto por que o ponto inicial está próximo da solução e as funções são quadráticas. O segundo problema que estudamos foi de mínimos quadrados, este problema consiste em encontrar  $x$  que minimiza o problema 2.30. O método que utilizamos para este problema foi de Gauss Newton. Realizamos testes em quatro exemplos dos quais apresentaram uma convergência rápida com um número de iteração pequeno. O último problema que estudamos foi encontrar a solução de um sistema não linear 2.40. Para este problema consideramos a direção  $d_k$  dada pelo método de Newton para sistema não linear 2.47. Usamos os mesmos exemplos os quais testamos o método de Gauss Newton. O resultado em relação ao número de iteração foi bem semelhante ao método anterior mas o tempo de convergência foi maior. Concluímos assim que é melhor optarmos pelo método de Gauss Newton nos casos que pudermos utilizar os dois métodos.

No capítulo três apresentamos uma breve revisão bibliográfica a respeito de problemas com restrições, neste capítulo concentramos na apresentação das condições de KKT (apresentadas na equação 3.13) para um problema geral de programação não linear (problema 3.3).

No último capítulo apresentamos duas alternativas para encontrar a solução do problema 1.1. A primeira alternativa considera o problema original o qual discretizamos obtendo assim um sistema não linear  $r(u) = 0$ . Para resolver este sistema  $r(u) = 0$  utilizamos o algoritmo baseado no método de Gauss Newton combinado com interpolação spline. A segunda alternativa considera o sistema não linear  $r(u) = 0$  como restrição de igualdade e a função objetivo definida se baseia nos resultados do teorema do ponto fixo de Banach. Para resolver este problema usamos um algoritmo do *matlab* o qual resolve problemas não lineares com restrições. Os resultados obtidos foram promissores, pois os dois algoritmos apresentaram bom desempenho visto que para os dez exemplares encontramos soluções com um número de iterações pequena. Observe que esperávamos que o algoritmo 2 fosse consideravelmente mais lento, porém o tempo gasto entre os algoritmos foi equilibrado, isto se deve principalmente ao fato de que o Algoritmo 2 consumiu um número menor de iterações internas. Como perspectivas

futuras pretendemos explorar a eficiência destes algoritmos com outras equações diferenciais que permitem representação através de uma equação integral, bem como explorar as possíveis vantagens de se considerar o Algoritmo 2 realizando mais testes. Podemos ressaltar que quando comparamos os Algoritmos 1 e 2 com métodos numéricos baseados em ponto fixo, uma grande vantagem é que em problemas com múltiplas soluções podemos determinar todas as soluções e não ficamos necessariamente restritos a solução de menor norma como ocorreu no exemplo 8.

**REFERÊNCIAS**

- [1] A.A.RIBEIRO, AND E.W.KARAS. *Um Curso De Otimização*. Curitiba, (2012).
- [2] AGARWAL, R., MEEHAN, M., AND O'REGAN, D. *Fixed Point Theory and Applications*. Cambridge University Press, 2001.
- [3] AVERY, R. I., AND PETERSON, A. C. Three positive fixed points of nonlinear operators in ordered banach spaces. *Computers and Mathematics with Applications* 42 (2001), 313–322.
- [4] C.A.SASSI. Métodos numéricos em fluido térmica. Notas de aula, Universidade Estadual Paulista, 2007.
- [5] CASTELANI, E. V., AND MA, T. F. Numerical solutions for a three-point boundary value problem. *Communications in Applied Analysis* 11 (2007), 87–96.
- [6] CASTELANI, E. V., AND MARTINEZ, A. L. M. Resolvendo uma equação de múltiplos pontos utilizando métodos de otimização. *XXXV CNMAC* (2014).
- [7] C.P. GUPTA, S. T. A sharper condition for the solvability of a three-point second order boundary value problem. *J. Math. Anal. Appl.* 205 ((1992)), 586–597.
- [8] D.G.LUENBERGER, AND YE, Y. *Linear and Nonlinear Programming*, 3 ed. Springer, New York, (2008).
- [9] E.L.LIMA. *Curso de Análise volume 2*, 11 ed. IMPA, Rio de Janeiro.
- [10] E.V.CASTELANI. Uma equação diferencial de segunda ordem com múltiplos pontos de fronteira via teorema de krasnoselskii. Mestrado, Dissertação de Mesrado ,Universidade Estadual de Maringá,Maringá, 2006.
- [11] E.V.CASTELANI, AND MARTINEZ, A. Existence and multiplicity of positive solutions for a m-point boundary value problem. *International Journal of Applied Mathematics* 38 (2012), 44.
- [12] E.V.CASTELANI, R.HOTO, AND A.L.M.MARTINEZ. Solving a second order m-point boundary value problem. *Elsevier* (2013).
- [13] E.V.CASTELANI, A.L.M.MARTINEZ, C., AND T.S.PINTO. *Um estudo de soluções para um problema de segunda ordem com múltiplos pontos de fronteira*, vol. 2. pp. 255–263.

- [14] E.V.CASTELANI, A.L.M.MARTINEZ, C., AND T.S.PINTO. Remarks on positive solutions for an m-point boundary value problem. *International Journal of Applied Mathematics* 25 ((2012)), 281–297.
- [15] GUPTA, C. Solvability of a three-point nonlinear boundary value problem for a second order ordinary differential equation. *J. Math. Anal. Appl* 168 ((1992)), 540–551.
- [16] J. E.DENNIS, J., AND SCHNABEL, R. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, 1 ed. SIAMr, Philadelphia, (1983).
- [17] J.NOCEDAL, AND WRIGHT, S. *Numerical Optimization*, 2 ed. Springer, New York, (2006).
- [18] KALLEY, C. T. Iterative methods for optimization. *SIAM* (1999).
- [19] LIN, Y., AND CUI, M. A numerical solution to nonlinear multi-point boundary value problems in the reproducing kernel space. *Mathematical Methods in the Applied Sciences* 34 (2011), 44–47.
- [20] MA, R. Existence theorems for a second order three point boundary value problem. *J. Math. Anal. Appl* 212 (1997), 430–442.
- [21] MARTINEZ, A. L. M., MARTINEZ, C. P. A., CASTELANI, E. V., AND PINTO, T. S. Remarks on positive solutions for an m-point boundary value problem. *Int. Journal of Appl. Math.* 25 (2012), 281–297.
- [22] R.MA. Multiplicity of positive solutions for second-order three-point boundary value problems. *Comput.and Math. with Appl* 40 (7/2000), 193–204.
- [23] V.A.IL'IN, AND E.I.MOISEEV. A nonlocal boundary-value problem of the second kind for a Sturm-Liouville operator. *Differential Equations* 23 (1987), 979–987.
- [24] V.A.IL'IN, AND E.I.MOISEEV. *A nonlocal boundary value problem for the Sturm-Liouville operator in the differential and difference treatments*, vol. 34. (1987).
- [25] V.A.IL'IN, AND E.I.MOISEEV. Nonlocal boundary-value problem of the second kind for a Sturm-Liouville operator. *Differential Equations* 23 (1987), 979–987.
- [26] WONG, P. J. Y., AND AGARWAL, R. P. Existence and uniqueness of solutions for three-point boundary value problems for second order difference equations. *Proceedings of Dynamic Systems and Applications* (1995).
- [27] Y.LIN, AND M.CUI. A numerical solution to nonlinear multi-point boundary value problems in the reproducing kernel space. *Mathematical Methods in the Applied Sciences* 34 (2011), 44–47.

- [28] ZOU, Y., HU, Q., AND ZHANG, R. On numerical studies of multi-point boundary value problem and its fold bifurcation. *App. Math. and Comp.* 185 (2007), 527–537.