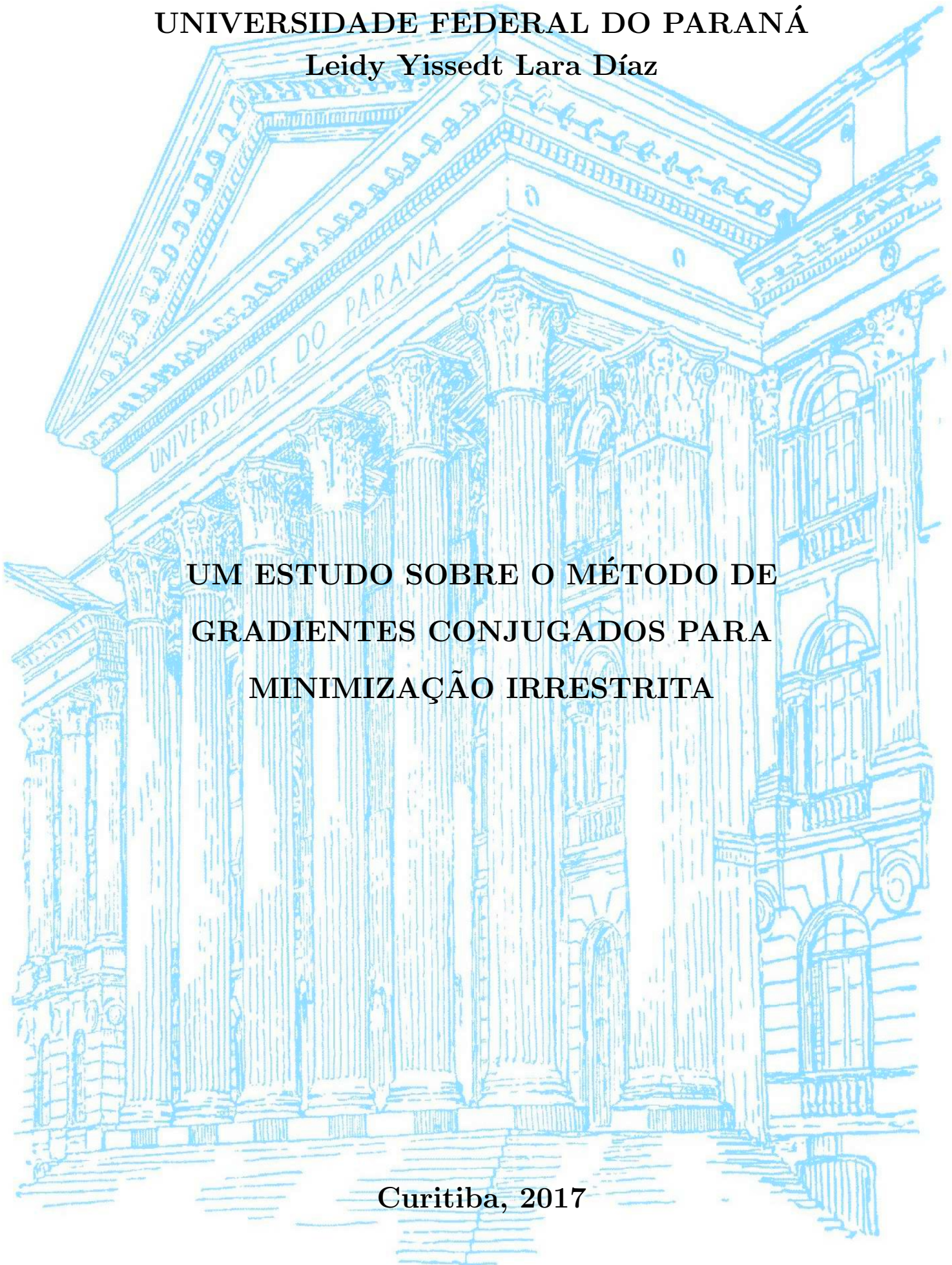


UNIVERSIDADE FEDERAL DO PARANÁ

Leidy Yissedt Lara Díaz

**UM ESTUDO SOBRE O MÉTODO DE
GRADIENTES CONJUGADOS PARA
MINIMIZAÇÃO IRRESTRITA**

Curitiba, 2017



UNIVERSIDADE FEDERAL DO PARANÁ
Leidy Yissedt Lara Díaz

**UM ESTUDO SOBRE O MÉTODO DE
GRADIENTES CONJUGADOS PARA
MINIMIZAÇÃO IRRESTRITA**

Dissertação apresentada ao Programa de Pós-Graduação em Matemática da Universidade Federal do Paraná, como requisito parcial à obtenção do grau de Mestre em Matemática.

Orientador: Dr. Lucas Garcia Pedroso

Coorientador: Dr. Luiz Carlos Matioli

Curitiba, 2017

Díaz, Leidy Yissedt Lara

Um estudo sobre o método de gradientes conjugados para minimização irrestrita / Leidy Yissedt Lara Díaz. – Curitiba, 2017. 59 f. : il.

Dissertação (mestrado) – Universidade Federal do Paraná, Setores de Tecnologia e de Ciências Exatas, Programa de Pós-Graduação Matemática.

Orientador: Lucas Garcia Pedroso

Coorientador: Luiz Carlos Matioli

Bibliografia: p. 56-59

1. Otimização matemática. 2. Algoritmos. I. Pedroso, Lucas Garcia. II. Matioli, Luiz Carlos. III. Título.

CDD 519.3



MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DO PARANÁ
PRÓ-REITORIA DE PESQUISA E PÓS-GRADUAÇÃO
Setor CIÊNCIAS EXATAS
Programa de Pós-Graduação MATEMÁTICA

TERMO DE APROVAÇÃO

Os membros da Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em MATEMÁTICA da Universidade Federal do Paraná foram convocados para realizar a arguição da dissertação de Mestrado de **LEIDY YISSEDT LARA DÍAZ** intitulada: **UM ESTUDO SOBRE O MÉTODO DE GRADIENTES CONJUGADOS PARA MINIMIZAÇÃO IRRESTRITA**, após terem inquirido a aluna e realizado a avaliação do trabalho, são de parecer pela sua aprovação.

Curitiba, 24 de Fevereiro de 2017.

Lucas Garcia Pedroso
LUCAS GARCIA PEDROSO

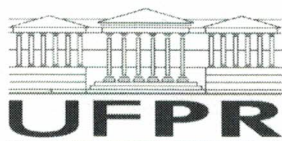
Presidente da Banca Examinadora (UFPR)

Mael Sachine
MAEL SACHINE

Avaliador Interno (UFPR)

Rodolfo Gotardi Begiato
RODOLFO GOTARDI BEGIATO

Avaliador Externo (UTFPR)



MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DO PARANÁ
PRÓ-REITORIA DE PESQUISA E PÓS-GRADUAÇÃO
Setor CIÊNCIAS EXATAS
Programa de Pós-Graduação MATEMÁTICA

ATA Nº080

ATA DE SESSÃO PÚBLICA DE DEFESA DE MESTRADO PARA A OBTENÇÃO DO GRAU DE MESTRE EM MATEMÁTICA

No dia vinte e quatro de Fevereiro de dois mil e dezessete às 14:00 horas, na sala Anfiteatro A, Rua Cel. Francisco H. dos Santos, 100 - Jardim das Américas, foram instalados os trabalhos de arguição da mestranda **LEIDY YISSEDT LARA DÍAZ** para a Defesa Pública de sua dissertação intitulada **UM ESTUDO SOBRE O MÉTODO DE GRADIENTES CONJUGADOS PARA MINIMIZAÇÃO IRRESTRITA**. A Banca Examinadora, designada pelo Colegiado do Programa de Pós-Graduação em MATEMÁTICA da Universidade Federal do Paraná, foi constituída pelos seguintes Membros: LUCAS GARCIA PEDROSO (UFPR), MAEL SACHINE (UFPR), RODOLFO GOTARDI BEGIATO (UTFPR). Dando início à sessão, a presidência passou a palavra a discente, para que a mesma expusesse seu trabalho aos presentes. Em seguida, a presidência passou a palavra a cada um dos Examinadores, para suas respectivas arguições. A aluna respondeu a cada um dos arguidores. A presidência retomou a palavra para suas considerações finais e, depois, solicitou que os presentes e a mestranda deixassem a sala. A Banca Examinadora, então, reuniu-se sigilosamente e, após a discussão de suas avaliações, decidiu-se pela aprovação da aluna. A mestranda foi convidada a ingressar novamente na sala, bem como os demais assistentes, após o que a presidência fez a leitura do Parecer da Banca Examinadora. Nada mais havendo a tratar a presidência deu por encerrada a sessão, da qual eu, LUCAS GARCIA PEDROSO, lavrei a presente ata, que vai assinada por mim e pelos membros da Comissão Examinadora.

Curitiba, 24 de Fevereiro de 2017.


LUCAS GARCIA PEDROSO

Presidente da Banca Examinadora (UFPR)


MAEL SACHINE

Avaliador Interno (UFPR)


RODOLFO GOTARDI BEGIATO

Avaliador Externo (UTFPR)

*Dedico este trabalho
à minha família*

Agradecimentos

Agradeço aos meus pais, Edith e Juan, e aos meus irmãos, Juan e Eimmy, porque estiveram sempre comigo apesar da distância, e me deram todo o seu apoio e amor, que me permitiu continuar nos momentos de dificuldade e de muita saudade.

Aos meus orientadores, os professores Lucas e Luiz, que procuraram sempre me guiar, ensinar e apoiar em cada passo de meu mestrado e de meu trabalho de dissertação.

Obrigada aos professores que foram parte de minha formação, ao programa de pós-graduação PPGM e aos meus colegas que me mostraram o caminho para conseguir com sucesso o título de mestre.

Agradeço também aos membros da banca, por sua participação e por suas sugestões enriquecedoras.

À OEA por seu suporte financeiro.

Aos meus amigos no Brasil, em especial Pamela, Gerald, John, Claudia, Aura, Elvis, Ariel, Guido, Lucia e Nash, que sempre me incentivarem e me ajudaram em todo este caminhar. Finalmente, agradeço ao meu namorado Gabriel, por todo seu carinho e companhia, que me permitiram sonhar e continuar com o doutorado no Chile.

Leidy Yissedt Lara Díaz

Resumo

Dentre os métodos para a minimização irrestrita de funções contínuas e diferenciáveis encontramos o Método de Gradientes Conjugados, que é o foco deste trabalho. Revisamos várias das suas versões, que diferem principalmente na escolha do parâmetro β_k da atualização na direção de busca. Além disso, estudamos também as propriedades teóricas dos algoritmos clássicos, como os propostos por Hestenes e Stiefel, Fletcher e Reeves e Polak, Rebière e Polyak. Em seguida, analisamos o método proposto por Dai e Kou em 2013, que utiliza elementos da regra de atualização BFGS para construir as direções de busca, bem como traz melhorias para as condições de Wolfe. Ao final do texto, apresentamos alguns experimentos numéricos para avaliar o desempenho do método de Gradientes Conjugados para algumas escolhas do parâmetro β_k e do critério de busca linear utilizado.

Palavras chaves: Otimização irrestrita, Gradientes Conjugados, Critério de Wolfe, Experimentos numéricos.

Abstract

Among the methods for unconstrained optimization of continually differentiable functions we find the Conjugate Gradient Method, which is the subject of this work. We revise many of its versions, that differ mainly in the choice of the parameter β_k of the search direction update. We also study the theoretical properties of some classical algorithms, such as the ones proposed by Hestenes and Stiefel, Fletcher and Reeves and Polak, Rebière and Polyak. After that, we analyze the method proposed by Dai and Kou in 2013, which uses elements from the BFGS update rule to build the search directions as well as brings some improvements to the Wolfe conditions. At the end of the text, we present some numerical experiments to evaluate the performance of the Conjugate Gradient method for some choices of the parameter β_k and of the linear search criterion used.

Keywords: Unconstrained optimization, Conjugate Gradient, Wolfe criteria, Numerical experiments.

Sumário

Resumo	i
Abstract	ii
Sumário	iii
Lista de Tabelas	v
Lista de Figuras	vi
Introdução	1
1 Método de Gradientes Conjugados para Problemas Quadráticos	3
1.1 Direções Conjugadas	3
1.2 Método de Gradientes Conjugados	7
1.3 Análise de Convergência	10
2 Método de Gradientes Conjugados para Problemas Não Quadráticos	14
2.1 Um Breve Histórico	15
2.2 Condições de Parada na Busca Linear	16
2.3 Método de Fletcher-Reeves	19
2.4 Método de Polak-Ribière-Polyak e Variantes	24
2.4.1 Outros Métodos com $g_{k+1}^T y_k$ no Numerador de β_k	25
2.5 Resultados de Convergência	28
3 Um Método de Gradientes Conjugados com Direções Baseadas na Atualização de BFGS	32

3.1	Gradientes Conjugados Baseados na Atualização BFGS	32
3.2	Busca de Wolfe Melhorada	35
3.3	Algoritmo e Análise de Convergência	39
4	Experimentos Numéricos	41
4.1	Algoritmo Implementado	41
4.2	Metodologia de teste	44
4.3	Resultados Numéricos	45
5	Conclusões	54
	Referências Bibliográficas	56

Lista de Tabelas

2.1	Resumo cronológico de diferentes escolhas do parâmetro β_k	15
4.1	Resumo das escolhas do parâmetro β_k testadas.	43
4.2	Tabela de resultados para a busca de Armijo.	47
4.3	Tabela de resultados para a busca de Wolfe Forte.	49
4.4	Tabela de resultados para a busca de Wolfe melhorada.	51

Lista de Figuras

4.1	Perfil de desempenho em relação ao número de iterações para a busca de Armijo.	48
4.2	Perfil de desempenho em relação ao número de avaliações de funções para a busca de Armijo.	48
4.3	Perfil de desempenho em relação ao número de iterações para a busca de Wolfe Forte.	50
4.4	Perfil de desempenho em relação ao número de avaliações de função para a busca de Wolfe Forte.	50
4.5	Perfil de desempenho em relação ao número de iterações para a busca de Wolfe melhorada.	52
4.6	Perfil de desempenho em relação ao número de avaliações de função para a busca de Wolfe melhorada.	52
4.7	Os melhores betas em robustez e eficiência com a busca de Wolfe Forte.	53

Introdução

A Otimização é uma linha de pesquisa relevante em Matemática Aplicada, já que tem um forte potencial na solução de problemas práticos em diferentes áreas, como por exemplo, nas Ciências Biológicas, nas Ciências Econômicas e na Engenharia. Com ela, podemos abordar problemas de minimização ou de maximização com ou sem restrições, tanto para funções lineares como não lineares, assim como diferenciáveis ou não diferenciáveis. É de especial interesse para nosso trabalho o problema de minimizar funções sem restrições, que sejam continuamente diferenciáveis, isto é,

$$\begin{aligned} & \text{Minimizar } f(x) \\ & \text{sujeito a } x \in \mathbb{R}^n, \end{aligned} \tag{1}$$

onde $f : \mathbb{R}^n \rightarrow \mathbb{R}$ é continuamente diferenciável. Dentre os diversos algoritmos e técnicas utilizadas para resolver (1), daremos central atenção a métodos que utilizam buscas lineares, os quais têm a característica de serem iterativos com a seguinte regra de formação: sendo $x_0 \in \mathbb{R}^n$ um ponto inicial, os métodos geram uma sequência iterativa de pontos $\{x_k\}$ da forma,

$$x_{k+1} = x_k + \alpha_k d_k, \text{ para } k = 0, 1, 2, \dots$$

onde $\alpha_k \in \mathbb{R}$ é o tamanho do passo e $d_k \in \mathbb{R}^n$ é a direção de busca. A direção d_k e o parâmetro α_k podem ser escolhidos de formas diferentes, o que origina métodos diferentes. Por exemplo, os clássicos: Método do Gradiente, Método de Newton, Método de Quase-Newton e Método de Gradientes Conjugados (ver [23]) diferem principalmente (mas não apenas) na definição da direção de busca. Por exemplo, o Método do Gradiente tem como direção d_k a de máxima descida, ou seja, $d_k = -\nabla f(x_k)$, enquanto os Métodos de Gradientes Conjugados, que são os de nosso interesse, têm a direção d_k da forma

$$d_{k+1} = -g_{k+1} + \beta_k d_k, \text{ para } k = 0, 1, 2, \dots \tag{2}$$

sendo g_{k+1} o gradiente da função f no ponto x_{k+1} e $\beta_k \in \mathbb{R}$ o parâmetro que permite a conjugação das direções d_k e d_{k+1} , as quais apresentaremos no decorrer do trabalho. Por

outro lado, podemos notar em (2) que, dependendo das escolhas do parâmetro β_k , existirá uma variedade de algoritmos de Gradientes Conjugados, fato que vamos explorar no presente texto.

A história dos Métodos de Gradientes Conjugados se inicia nos anos 50, com a primeira proposta devida a Hestenes e Stiefel [19]. Eles apresentaram um método iterativo para resolver sistemas de equações lineares com matrizes de coeficientes definidas positivas. Até hoje este é tido, além do método de eliminação de Gauss, como uma forma adequada para resolver diretamente sistemas de equações lineares de grande porte. Já nos anos 60, o primeiro algoritmo de Gradientes Conjugados para o problema (1) foi introduzido por Fletcher e Reeves [12], sendo uma das primeiras técnicas para resolver problemas de grande escala de otimização não-linear. Ao longo dos anos, muitas variantes destes métodos têm sido amplamente utilizadas na prática. As características mais atraentes desses algoritmos são a economia, uma vez que não necessitam do armazenamento de matrizes, e a rapidez, pois são mais rápidos que o Método do Gradiente. Isso motiva o contínuo estudo do método, tornando-o um assunto atual.

Para termos condições de desenvolver o estudo e análise dos algoritmos de Gradientes Conjugados para problemas gerais (1), faremos no Capítulo 1 uma revisão do método para o caso quadrático, discutindo suas qualidades e características teóricas e computacionais. No Capítulo 2, generalizamos o método para o caso não quadrático em que existe uma gama de possibilidades para o parâmetro beta. Estudaremos também as propriedades, vantagens e desvantagens, tanto computacionais como de convergência, das versões mais importantes do método. No Capítulo 3, discutiremos o trabalho recente de Dai e Kou [5], onde são introduzidas uma nova forma de gerar direções conjugadas e algumas modificações na busca linear de Wolfe e do Método de Gradientes Conjugados. No Capítulo 4, apresentaremos os resultados numéricos obtidos em nossos testes computacionais para os problemas da coleção de Moré, Garbow e Hillstom [22] e, finalmente, as conclusões são apresentadas no Capítulo 5.

Capítulo 1

Método de Gradientes Conjugados para Problemas Quadráticos

O estudo do método de Gradientes Conjugados para o caso quadrático, ou seja, em que f no problema (1) é uma função quadrática, é de grande importância, já que fornece uma técnica que permite resolver sistemas de equações lineares e, além disso, podem ser adaptados para resolver problemas de Otimização não quadráticos.

No presente capítulo, pretendemos apresentar as propriedades mais importantes do método no caso quadrático, sendo finalmente o caso não-quadrático o objeto de nosso interesse nos próximos capítulos.

Para começar o estudo, apresentaremos definições, teoremas e outros resultados importantes para o desenvolvimento inicial do método, para logo discutir o caso quadrático. As principais referências usadas no presente capítulo são [21] e [23], onde todos os resultados podem ser encontrados.

1.1 Direções Conjugadas

Definição 1.1 ([21], Pág. 263). *Dada uma matriz simétrica $B \in \mathbb{R}^{n \times n}$, dizemos que dois vetores $d_1, d_2 \in \mathbb{R}^n$ são B -conjugados se $d_1^T B d_2 = 0$.*

Da definição dada acima, podemos ver que se a matriz B é a matriz nula, então todos os vetores serão B -conjugados, já se B é a matriz identidade, recuperamos a definição de

ortogonalidade.

Se considerarmos agora um conjunto de vetores não nulos $\{d_i\}_{i=0}^{n-1}$, estes serão chamados de B -conjugados se $d_i^T B d_j = 0$, para todo $i \neq j$, os quais têm a propriedade de serem linearmente independentes, resultado que mostramos na seguinte proposição.

Proposição 1.2 ([21], Pág. 264). *Se B é uma matriz simétrica definida positiva e o conjunto de vetores não nulos $\{d_i\}_{i=0}^{n-1}$ é B -conjugado, então $\{d_i\}_{i=0}^{n-1}$ é linearmente independente.*

Demonstração. Consideremos constantes $\alpha_i, i = 0, \dots, n - 1$, tais que

$$\alpha_0 d_0 + \dots + \alpha_{n-1} d_{n-1} = 0.$$

Multiplicando a expressão acima por $d_i^T B$ vemos pelo fato das direções serem B -conjugadas que

$$\alpha_i d_i^T B d_i = 0,$$

e como $d_i^T B d_i > 0$ pela positividade da matriz B , obtemos que $\alpha_i = 0$, isto é, que o conjunto $\{d_i\}_{i=0}^{n-1}$ é linearmente independente. \square

Uma consequência direta da proposição acima é que dada uma matriz $B \in \mathbb{R}^{n \times n}$ simétrica definida positiva, não é possível obter mais de n vetores não nulos B -conjugados. Por isso, quando falamos de um conjunto de vetores B -conjugados estaremos supondo que nenhum dos vetores é nulo, salvo menção contrária.

Seja o problema quadrático

$$\text{Minimizar } f(x) = \frac{1}{2} x^T B x - b^T x \quad (1.1)$$

onde $B \in \mathbb{R}^{n \times n}$ é simétrica definida positiva e $b \in \mathbb{R}^n$. Sabemos que a solução deste problema é também solução do sistema linear

$$Bx = b, \quad (1.2)$$

de modo que resolver o problema de minimização quadrática é equivalente a resolver um sistema linear. Consideremos um conjunto de vetores B -conjugados $\{d_i\}_{i=0}^{n-1} \subset \mathbb{R}^n$. Como tal conjunto é linearmente independente pela Proposição 1.2, podemos escrever a solução

x_* de (1.1) ou (1.2) como combinação linear de d_0, d_1, \dots, d_{n-1} , ou seja existem escalares $\alpha_0, \dots, \alpha_{n-1}$ tais que

$$x_* = \alpha_0 d_0 + \alpha_1 d_1 + \dots + \alpha_{n-1} d_{n-1}. \quad (1.3)$$

Para $i = 0, \dots, n-1$, multiplicando (1.3) por $d_i^T B$, obtemos pela conjugação que

$$\alpha_i = \frac{d_i^T B x_*}{d_i^T B d_i} = \frac{d_i^T b}{d_i^T B d_i}, \quad i = 0, \dots, n-1.$$

Portanto, a solução x_* pode ser expressa como

$$x_* = \sum_{i=0}^{n-1} \frac{d_i^T b}{d_i^T B d_i} d_i. \quad (1.4)$$

A expressão (1.4) pode ser considerada como o resultado de um processo iterativo de n passos, onde no i -ésimo passo $\alpha_i d_i$ é adicionado. Vendo o procedimento desta forma e permitindo um ponto inicial arbitrário, o método básico de direções conjugadas é obtido, conforme o teorema abaixo.

Teorema 1.3 (Método de Direções Conjugadas [21], Pág. 265). *Considere $B \in \mathbb{R}^{n \times n}$ uma matriz simétrica definida positiva e $\{d_i\}_{i=0}^{n-1}$ um conjunto de vetores B -conjugados. Para todo $x_0 \in \mathbb{R}^n$ a sequência $\{x_k\}$ gerada por*

$$x_{k+1} = x_k + \alpha_k d_k, \quad k = 0, \dots, n-1 \quad (1.5)$$

com

$$\alpha_k = \frac{-g_k^T d_k}{d_k^T B d_k} \quad (1.6)$$

e

$$g_k = Bx_k - b \quad (1.7)$$

converge à única solução x_* do sistema $Bx = b$, isto é, $x_n = x_*$.

Demonstração. Como os vetores d_k , $k = 0, \dots, n-1$, são linearmente independentes pela Proposição 1.2, podemos escrever

$$x_* - x_0 = \beta_0 d_0 + \beta_1 d_1 + \dots + \beta_{n-1} d_{n-1}$$

para um conjunto de coeficientes $\beta_0, \beta_1, \dots, \beta_{n-1}$ adequados. Multiplicando a expressão acima por B , obtemos

$$B(x_* - x_0) = \beta_0 B d_0 + \beta_1 B d_1 + \dots + \beta_{n-1} B d_{n-1}$$

e tomando o produto escalar com d_k , vemos da hipótese de B -conjugação que

$$d_k^T B(x_* - x_0) = \beta_k d_k^T B d_k.$$

Logo,

$$\beta_k = \frac{d_k^T B(x_* - x_0)}{d_k^T B d_k}.$$

Por outro lado, por (1.5) temos que

$$x_k - x_0 = \alpha_0 d_0 + \alpha_1 d_1 + \dots + \alpha_{k-1} d_{k-1},$$

a qual multiplicando por $d_k^T B$ e usando a B -conjugação implica

$$d_k^T B(x_k - x_0) = 0,$$

de onde podemos ver que

$$d_k^T B x_k = d_k^T B x_0. \quad (1.8)$$

Substituindo (1.8) na expressão de β_k e usando (1.7), obtemos

$$\beta_k = \frac{d_k^T B(x_* - x_0)}{d_k^T B d_k} = \frac{d_k^T B x_* - d_k^T B x_0}{d_k^T B d_k} = \frac{d_k^T B x_* - d_k^T B x_k}{d_k^T B d_k} = \frac{d_k^T [b - B x_k]}{d_k^T B d_k} = \frac{-d_k^T g_k}{d_k^T B d_k},$$

logo

$$\beta_k = \frac{-g_k^T d_k}{d_k^T B d_k} = \alpha_k,$$

ou seja, $x_n - x_0 = x_* - x_0$, completando a prova. \square

Até o momento derivamos o método de Direções Conjugadas essencialmente da observação de que resolver (1.1) é equivalente a resolver (1.2). Vejamos agora algumas propriedades do método.

Lema 1.4 ([21], Corolário Pág 267). *No método de Direções Conjugadas (Teorema 1.3) o gradiente g_k , $k = 0, 1, \dots, n$ satisfaz*

$$g_k^T d_i = 0, \text{ para } i < k.$$

Demonstração. Faremos a prova por indução. Claramente a tese é válida para $k = 0$ por vacuidade. Vamos supor que vale para k , ou seja, que $g_k^T d_i = 0$ para $i = 1, \dots, k - 1$. Precisamos provar que a tese é válida para $k + 1$, ou seja, $g_{k+1}^T d_i = 0$ para $i = 1, \dots, k$.

Multiplicando (1.5) por B , subtraindo b e multiplicando por d_i^T , vemos que

$$d_i^T g_{k+1} = d_i^T g_k + \alpha_k d_i^T B d_k.$$

Para $i = 1, \dots, k - 1$, temos que ambas as parcelas da soma do lado direito da expressão acima se anulam, uma vez que $d_i^T g_k = 0$ por hipótese indutiva e $d_i^T B d_k = 0$ pela B -conjugação. Já para $i = k$, basta substituímos a expressão de α_k dada em (1.6) para chegarmos à conclusão que $d_k^T g_{k+1} = 0$, o que completa a prova. \square

Definindo \mathbf{K}_k como o subespaço de \mathbb{R}^n gerado por $\{d_i\}_{i=0}^{k-1}$, vamos provar que o iterando x_k do método de direções conjugadas minimiza a função objetivo de (1.1) sobre a variedade linear k -dimensional $x_0 + \mathbf{K}_k$.

Teorema 1.5 (Subespaços Expandidos [21], Pág. 266). *Considere B uma matriz simétrica definida positiva e $\{d_i\}_{i=0}^{n-1}$ um conjunto de vetores não-nulos B -conjugados em \mathbb{R}^n . Então, no método de Direções Conjugadas, para todo $x_0 \in \mathbb{R}^n$ a sequência gerada por*

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

com

$$\alpha_k = \frac{-g_k^T d_k}{d_k^T B d_k}$$

tem a propriedade de que x_k é um minimizador de $f(x) = \frac{1}{2}x^T B x - b^T x$ sobre a variedade linear $x_0 + \mathbf{K}_k$, e também sobre a reta $x = x_{k-1} + \alpha d_{k-1}$, $\alpha \in \mathbb{R}$.

Demonstração. Para a prova, só precisamos mostrar que x_k minimiza f sobre a variedade linear $x_0 + \mathbf{K}_k$, já que esta contém a reta $x = x_{k-1} + \alpha d_{k-1}$, $\alpha \in \mathbb{R}$. Como a função f é estritamente convexa, basta mostrarmos que g_k é ortogonal a \mathbf{K}_k (Corolário 3.14 de [33]). No entanto, isso é consequência direta do Lema 1.4 e da definição de \mathbf{K}_k . \square

O resultado acima às vezes é chamado de Teorema de subespaços expandidos, já que $\mathbf{K}_{k+1} \supset \mathbf{K}_k$ e x_k minimiza a função objetivo sobre $x_0 + \mathbf{K}_k$. Como $x_0 + \mathbf{K}_n = \mathbb{R}^n$, obtemos um dos resultados mais importantes desta seção.

Apesar de termos o Teorema 1.3, que apresenta um resultado forte de convergência em um número finito de iterações, não discutimos ainda como conseguir um conjunto de n direções B -conjugadas. Para construirmos tal conjunto, o Lema 1.4 será muito importante, conforme discutiremos na próxima seção.

1.2 Método de Gradientes Conjugados

O método de Gradientes Conjugados é um método de direções conjugadas com propriedades interessantes. Na k -ésima iteração, calculamos uma direção d_k como uma combinação

linear conveniente dos vetores g_k e d_{k-1} . Apesar de usarmos apenas essas duas direções, conseguimos provar que d_k é B-conjugada não apenas em relação a d_{k-1} , mas também a d_0, \dots, d_{k-2} , que é um dos requerimentos de um método de Direções Conjugadas. Além disso, os cálculos envolvidos são bastante simples, o que é um dos fatores que tornam o método atraente. Podemos definir um algoritmo de Gradientes Conjugados para resolver os problemas (1.1) e (1.2) como segue.

Algoritmo 1.1. (*Gradientes Conjugados*)

Dado $x_0 \in \mathbb{R}^n$, defina $d_0 = -g_0 = b - Bx_0$. Para $k = 0, 1, \dots, n - 1$, faça

$$\alpha_k = \frac{-g_k^T d_k}{d_k^T B d_k}, \quad (1.9)$$

$$x_{k+1} = x_k + \alpha_k d_k, \quad (1.10)$$

$$\beta_k = \frac{g_{k+1}^T B d_k}{d_k^T B d_k}, \quad (1.11)$$

$$d_{k+1} = -g_{k+1} + \beta_k d_k, \quad (1.12)$$

onde $g_k = Bx_k - b$.

Note que os passos (1.9)-(1.10) são iguais aos do algoritmo de máxima descida ([21] Seção 8.6). Uma característica importante do Algoritmo 1.1 é a formulação simples dada por (1.11) e (1.12) para a atualização da direção. Isso deixa o método um pouco mais complicado que o método de máxima descida, mas convergente em um número finito de passos. O seguinte teorema mostra algumas propriedades importantes do Algoritmo 1.1.

Teorema 1.6 (Teorema de Gradientes Conjugados. [23], Teorema 5.3). *O Algoritmo 1.1 é um método de direções conjugadas. Se não terminar em x_k , então*

$$(a) [g_0, g_1, \dots, g_k] = [g_0, Bg_0, \dots, B^k g_0]$$

$$(b) [d_0, d_1, \dots, d_k] = [g_0, Bg_0, \dots, B^k g_0]$$

$$(c) d_k^T B d_i = 0, \forall i \leq k - 1$$

$$(d) g_k^T g_i = 0, \forall i \leq k - 1$$

$$(e) \alpha_k = \frac{g_k^T g_k}{d_k^T B d_k}$$

$$(f) \beta_k = \frac{g_{k+1}^T g_{k+1}}{g_k^T g_k}$$

Demonstração. Provaremos primeiro (a), (b) e (c) simultaneamente por indução. Para $k = 0$ as expressões (a), (b) são válidas e (c) vale para $k = 0$ por vacuidade. Suponhamos agora que são verdadeiras as teses até k e vejamos que também valem para $k + 1$.

Ao multiplicar $x_{k+1} = x_k + \alpha_k d_k$ por B e subtrair b , temos que $g_{k+1} = g_k + \alpha_k B d_k$ e pelas hipóteses de indução g_k e d_k pertencem a $[g_0, B g_0, \dots, B^k g_0]$, logo $g_{k+1} \in [g_0, B g_0, \dots, B^{k+1} g_0]$. Além disso, $g_{k+1} \notin [g_0, B g_0, \dots, B^k g_0] = [d_0, d_1, \dots, d_k]$ pois, caso contrário teríamos $g_{k+1} = 0$, já que pelo Lema 1.4 g_{k+1} , é ortogonal a $[d_0, \dots, d_k]$ e a hipótese indutiva para (c) garante que o método é de direções conjugadas até o ponto x_{k+1} .

Por outro lado, podemos ver que a relação (b) é válida para $k+1$ uma vez que, da definição de d_{k+1} , hipótese de indução em (b) e item (a),

$$\begin{aligned} [d_0, d_1, \dots, d_{k+1}] &= [d_0, \dots, d_k, g_{k+1}] \\ &= [g_0, \dots, B^k g_0, g_{k+1}] \\ &= [g_0, \dots, B^k g_0, B^{k+1} g_0]. \end{aligned}$$

Agora vamos provar (c). Da expressão $d_{k+1} = -g_{k+1} + \beta_k d_k$ obtemos que para $i = 0, 1, \dots, k$

$$d_{k+1}^T B d_i = -g_{k+1}^T B d_i + \beta_k d_k^T B d_i. \quad (1.13)$$

Pela definição de β_k em (1.11), podemos ver que o lado direito de (1.13) se anula quando $i = k$. Portanto $d_{k+1}^T B d_k = 0$. Para o caso $i < k$, temos que ambos os termos do lado direito de (1.13) se anulam. De fato, o primeiro termo é nulo, uma vez que $B d_i \in [d_1, \dots, d_{i+1}] \subset [d_1, \dots, d_{k+1}]$ e o Teorema do Subespaço Expandido garante que g_{k+1} é ortogonal a $[d_0, \dots, d_{k+1}]$. Já o segundo termo se anula pela hipótese de indução em (c). Assim, $d_{k+1}^T B d_i = 0$, $i = 1, \dots, k$ como queríamos demonstrar.

Para provar (d), por (c) e pelo Lema 1.4 temos que $g_k^T d_i = 0$ para $i = 0, 1, \dots, k-1$ e $k = 1, 2, \dots, n-1$. De (1.12) temos $g_i = -d_i + \beta_{i-1} d_{i-1}$, logo $g_i \in [d_{i-1}, d_i]$, para todo $i = 1, \dots, k-1$, concluindo que $g_k^T g_i = 0$, para todo $i = 1, \dots, k-1$.

Para provar (e) temos que por (1.12) e Lema 1.4

$$-g_k^T d_k = -g_k^T (-g_k + \beta_{k-1} d_{k-1}) = g_k^T g_k.$$

Portanto, da definição de α_k em (1.9) segue que

$$\alpha_k = \frac{g_k^T g_k}{d_k^T B d_k}. \quad (1.14)$$

Finalmente, para provar (f), multiplicamos (1.10) por B e isolamos $B d_k$, obtendo

$$B d_k = \frac{g_{k+1} - g_k}{\alpha_k}.$$

Multiplicando esta última equação por g_{k+1}^T e usando o fato que $g_{k+1}^T g_k = 0$, obtemos

$$g_{k+1}^T B d_k = \frac{g_{k+1}^T (g_{k+1} - g_k)}{\alpha_k} = \frac{g_{k+1}^T g_{k+1}}{\alpha_k}.$$

Logo, β_k ficará da seguinte forma

$$\beta_k = \frac{g_{k+1}^T g_{k+1}}{\alpha_k d_k^T B d_k},$$

e, substituindo α_k dado por (1.14), obtemos

$$\beta_k = \frac{g_{k+1}^T g_{k+1}}{g_k^T g_k}.$$

□

As novas opções de fórmulas para o cálculo de α_k e β_k fornecidas pelo Teorema 1.6 são mais econômicas, uma vez que apenas dois produtos internos precisam ser calculados na iteração k : $g_{k+1}^T g_{k+1}$ e $d_k^T B d_k$ (note que o numerador de α na iteração k é o mesmo de β na iteração $k - 1$). Com elas, portanto, obtemos o algoritmo de Gradientes Conjugados em sua forma mais conhecida.

Algoritmo 1.2. (*Gradientes Conjugados*)

Dado $x_0 \in \mathbb{R}^n$, defina $d_0 = -g_0 = b - Bx_0$. Para $k = 0, \dots, n - 1$, faça

$$\alpha_k = \frac{g_k^T g_k}{d_k^T B d_k}, \tag{1.15}$$

$$x_{k+1} = x_k + \alpha_k d_k, \tag{1.16}$$

$$\beta_k = \frac{g_{k+1}^T g_{k+1}}{g_k^T g_k},$$

$$d_{k+1} = -g_{k+1} + \beta_k d_k,$$

onde $g_k = Bx_k - b$.

1.3 Análise de Convergência

Pelo Corolário 1.6 temos que a solução de (1.1) é atingida pelo algoritmo no máximo em n passos. No entanto, podemos ter limitantes melhores caso os autovalores da matriz B tenham uma certa distribuição. Para provar isso, nesta seção estudaremos de maneira mais aprofundada o comportamento do método.

Seguindo as ideias apresentadas em [23], mostraremos que o Algoritmo 1.2 é, em um

certo sentido importante, ótimo. Considerando a regra (1.16) temos que a sequência $\{x_k\}$ gerada pelo algoritmo pode ser vista como

$$x_0, x_1 = x_0 + \alpha_0 d_0, x_2 = x_1 + \alpha_1 d_1 = x_0 + \alpha_0 d_0 + \alpha_1 d_1, \dots, x_{k+1} = x_0 + \alpha_0 d_0 + \dots + \alpha_k d_k,$$

e pela igualdade (b) do Teorema 1.6, obtemos que existem constantes γ_i , para $i = 0, 1, \dots, k$ tais que

$$x_{k+1} = x_0 + \alpha_0 d_0 + \dots + \alpha_k d_k = x_0 + \gamma_0 g_0 + \dots + \gamma_k B^k g_0.$$

Seja o polinômio $P_k^*(\cdot)$ de grau k e coeficientes $\gamma_0, \gamma_1, \dots, \gamma_k$. Como qualquer polinômio, P_k^* pode tomar um escalar ou uma matriz quadrada como seu argumento. Então, para a matriz B , temos que

$$P_k^*(B) = \gamma_0 I \dots + \gamma_k B^k,$$

de onde vemos que $x_{k+1} = x_0 + \gamma_0 I g_0 \dots + \gamma_k B^k g_0 = x_0 + P_k^*(B) g_0$. Veremos que entre todos os métodos cujos primeiros k passos estão restritos ao subespaço \mathbf{K}_{k+1} gerado pelos vetores $[g_0, Bg_0, \dots, B^k g_0]$, o algoritmo de Gradientes Conjugados faz o melhor trabalho de minimizar a distância à solução depois de k passos, quando essa distância é mensurável pela norma com peso B , definida por

$$\|z\|_B^2 = z^T B z.$$

Agora, usando essa norma e o problema

$$\text{Minimizar } f(x) = \frac{1}{2} x^T B x - b^T x$$

com o fato de que x_* minimiza a função f , é fácil ver que

$$\frac{1}{2} \|x - x_*\|_B^2 = \frac{1}{2} (x - x_*)^T B (x - x_*) = f(x) - f(x_*).$$

O Teorema 1.5 estabelece que x_{k+1} minimiza a função f , e conseqüentemente $\|x - x_*\|_B^2$, sobre o conjunto $x_0 + [d_0, d_1, \dots, d_k]$, o qual é igual a $x_0 + [g_0, Bg_0, \dots, B^k g_0]$. Isto mostra que o polinômio P_k^* resolve o seguinte problema,

$$\min_{P_k} \|x_0 + P_k(B) g_0 - x_*\|_B, \quad (1.17)$$

em que o mínimo é tomado sobre o subespaço de todos os possíveis polinômios de grau menor ou igual a k .

Como $g_0 = Bx_0 - b = Bx_0 - Bx_* = B(x_0 - x_*)$, temos que

$$\begin{aligned} x_{k+1} - x_* &= x_0 + P_k^*(B) g_0 - x_* \\ &= [I + P_k^*(B) B](x_0 - x_*). \end{aligned} \quad (1.18)$$

Agora, sejam $0 < \lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n$ os autovalores de B , e sejam v_1, v_2, \dots, v_n os autovetores ortonormais correspondentes, logo $B = \sum_{i=1}^n \lambda_i v_i v_i^T$. Como os autovetores de B geram todo o espaço \mathbb{R}^n , existem coeficientes $\xi_1, \dots, \xi_n \in \mathbb{R}$ tais que

$$x_0 - x_* = \sum_{i=1}^n \xi_i v_i. \quad (1.19)$$

É simples mostrar que cada autovetor de B é também um autovetor de $P_k(B)$ para cada polinômio P_k . Assim, se $\lambda_1, \dots, \lambda_n$ são autovalores de B com os correspondentes autovetores v_i , temos que

$$P_k(B)v_i = P_k(\lambda_i)v_i, \quad i = 1, \dots, n.$$

Substituindo (1.19) em (1.18), temos que

$$x_{k+1} - x_* = \sum_{i=1}^n [1 + \lambda_i P_k^*(\lambda_i)] \xi_i v_i.$$

Agora, usando o fato que $\|z\|_B^2 = z^T B z = \sum_{i=1}^n \lambda_i (v_i^T z)^2$, obtemos

$$\|x_{k+1} - x_*\|_B^2 = \sum_{i=1}^n \lambda_i [1 + \lambda_i P_k^*(\lambda_i)]^2 \xi_i^2.$$

Como P_k^* é solução de (1.17), temos que

$$\|x_{k+1} - x_*\|_B^2 = \min_{P_k} \sum_{i=1}^n \lambda_i [1 + \lambda_i P_k(\lambda_i)]^2 \xi_i^2.$$

Extraindo o maior dos termos $[1 + \lambda_i P_k(\lambda_i)]^2$ desta expressão, obtemos que

$$\begin{aligned} \|x_{k+1} - x_*\|_B^2 &\leq \min_{P_k} \max_{1 \leq i \leq n} [1 + \lambda_i P_k(\lambda_i)]^2 \left(\sum_{j=1}^n \lambda_j \xi_j^2 \right) \\ &= \min_{P_k} \max_{1 \leq i \leq n} [1 + \lambda_i P_k(\lambda_i)]^2 \|x_0 - x_*\|_B^2. \end{aligned} \quad (1.20)$$

onde usamos o fato que $\|x_0 - x_*\|_B^2 = \sum_{j=1}^n \lambda_j \xi_j^2$.

O resultado seguinte traz uma interessante consequência dessa discussão.

Teorema 1.7 ([23], Teorema 5.4). *Se B tem exatamente r autovalores distintos, então o método de Gradientes Conjugados encontra a solução em no máximo r iterações.*

Demonstração. Suponhamos que os autovalores $\lambda_1, \lambda_2, \dots, \lambda_n$ de B tomam r valores distintos $\sigma_1 < \sigma_2 < \dots < \sigma_r$. Definimos o polinômio $Q_r(\lambda)$ por

$$Q_r(\lambda) = \frac{(-1)^r}{\sigma_1 \sigma_2 \dots \sigma_r} (\lambda - \sigma_1)(\lambda - \sigma_2) \dots (\lambda - \sigma_r)$$

e notemos que $Q_r(\sigma_i) = 0$ para $i = 1, 2, \dots, r$ e $Q_r(0) = 1$.

Da última observação, deduzimos que $Q_r(\lambda) - 1$, é um polinômio de grau r com uma raiz em $\lambda = 0$, logo pela divisão polinomial, a função \bar{P}_{r-1} definida por

$$\bar{P}_{r-1} = \frac{Q_r(\lambda) - 1}{\lambda},$$

é um polinômio de grau $r - 1$. Por outro lado, temos que

$$0 \leq \min_{P_{r-1}} \max_{1 \leq i \leq n} [1 + \lambda_i P_{r-1}(\lambda_i)]^2 \leq \max_{1 \leq i \leq n} [1 + \lambda_i \bar{P}_{r-1}(\lambda_i)]^2 = \max_{1 \leq i \leq n} Q_r^2(\lambda_i) = 0.$$

Deste fato, fazendo $k = r - 1$ em (1.20), vemos que $\|x_r - x_*\|_B^2 = 0$ e portanto $x_r = x_*$, como queríamos. \square

Por fim, apresentamos um limitante de erro que dá uma caracterização útil do comportamento do método de Gradientes Conjugados.

Teorema 1.8 ([23], Teorema 5.5). *Se B tem autovalores $\lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n$ temos que*

$$\|x_{k+1} - x_*\|_B^2 \leq \left(\frac{\lambda_{n-k} - \lambda_1}{\lambda_{n-k} + \lambda_1} \right)^2 \|x_0 - x_*\|_B^2.$$

Demonstração. Ver em [21] demonstração do Teorema do Método de Gradientes Conjugados Parcial, nas páginas 275 – 276. \square

A estimativa dada no Teorema 1.8 pode ser usada para prever o comportamento do método de Gradientes Conjugados em problemas específicos. Por exemplo, supondo que temos dois grupos de autovalores de B : $n - m$ autovalores próximos de 1 ($\lambda_1 \leq \dots \leq \lambda_{n-m}$) e m autovalores grandes ($\lambda_{n-m+1} \leq \dots \leq \lambda_n$). Definindo $\varepsilon = \lambda_{n-m} - \lambda_1$, pelo Teorema 1.8 temos que depois de $m + 1$ passos do algoritmo de Gradientes Conjugados

$$\|x_{m+1} - x_*\|_B \approx \varepsilon \|x_0 - x_*\|_B.$$

Sendo ε pequeno, concluímos que as iterações do método de Gradientes Conjugados já proporcionam neste caso uma boa estimativa de solução depois de apenas $m + 1$ passos.

Capítulo 2

Método de Gradientes Conjugados para Problemas Não Quadráticos

No capítulo anterior, apresentamos o Método de Gradientes Conjugados para minimizar uma função quadrática convexa $f : \mathbb{R}^n \rightarrow \mathbb{R}$ dada por $f(x) = \frac{1}{2}x^T Bx - b^T x$, sendo $B \in \mathbb{R}^{n \times n}$ uma matriz simétrica definida positiva. Neste capítulo, trataremos do caso não-quadrático, ou seja, de minimizar uma função $f : \mathbb{R}^n \rightarrow \mathbb{R}$, com f continuamente diferenciável. Vamos considerar iterações da forma

$$x_{k+1} = x_k + \alpha_k d_k, \quad (2.1)$$

$$d_k = \begin{cases} -g_k & \text{para } k = 0 \\ -g_k + \beta_{k-1} d_{k-1} & \text{para } k \geq 1 \end{cases} \quad (2.2)$$

onde β_k é um escalar e α_k é o tamanho do passo obtido por alguma busca linear.

É de importância mencionar que, para este caso, o algoritmo gerado é muito similar ao Algoritmo 1.2, mas com modificações nos parâmetros α_k e β_k . O primeiro deles define o tamanho do passo na busca linear, o que no caso quadrático é feito de modo exato através de (1.15). Nos demais casos, a busca exata raramente pode ser aplicada, de modo que buscas inexatas são necessárias. Quanto ao parâmetro β_k , utilizado na atualização da busca, diversas expressões para calculá-lo podem ser encontradas na literatura. Nos algoritmos clássicos, tais expressões são equivalentes se a função objetivo f é quadrática, mas isso não pode ser garantido no caso geral.

As principais referências para este capítulo são [23] e [18].

2.1 Um Breve Histórico

Na busca por ampliar o método para outros tipos de problemas, neste caso para os problemas de otimização irrestrita não-quadrática, desde os anos 50 pesquisadores vêm desenvolvendo maneiras diferentes de escolher os parâmetros de atualização no Algoritmo 1.2. Para calcular o parâmetro α_k , existem as buscas lineares inexatas, que serão mencionadas mais adiante. Já em relação ao parâmetro β_k , veremos que este pode ser atualizado de várias maneiras distintas. Assim, os diferentes métodos existentes de Gradientes Conjugados para o caso não-quadrático são definidos pelas diferentes escolhas do parâmetro β_k . As mais importantes, de acordo com [18], estão apresentadas na Tabela 2.1.

$$\beta_k^{HS} = \frac{g_{k+1}^T y_k}{d_k^T y_k} \quad (1952) \text{ Hestenes e Stiefel [19] com } y_k = g_{k+1} - g_k.$$

$$\beta_k^{FR} = \frac{\|g_{k+1}\|^2}{\|g_k\|^2} \quad (1964) \text{ Fletcher e Reeves [12].}$$

$$\beta_k^D = \frac{g_{k+1}^T \nabla^2 f(x_k) d_k}{d_k^T \nabla^2 f(x_k) d_k} \quad (1967) \text{ Daniel [11].}$$

$$\beta_k^{PRP} = \frac{g_{k+1}^T y_k}{\|g_k\|^2} \quad (1969) \text{ Polak-Ribière-Polyak [30][29].}$$

$$\beta_k^{CD} = \frac{\|g_{k+1}\|^2}{-d_k^T g_k} \quad (1987) \text{ Fletcher, Chamado de Descida Conjugado [13].}$$

$$\beta_k^{LS} = \frac{g_{k+1}^T y_k}{-d_k^T g_k} \quad (1991) \text{ Liu e Storey [20].}$$

$$\beta_k^{DY} = \frac{\|g_{k+1}\|^2}{d_k^T y_k} \quad (1999) \text{ Dai e Yuan [9].}$$

$$\beta_k^N = \left(y_k - 2d_k \frac{\|y_k\|^2}{d_k^T y_k} \right)^T \frac{g_{k+1}}{d_k^T y_k} \quad (2005) \text{ Hager e Zhag [16].}$$

Tabela 2.1: Resumo cronológico de diferentes escolhas do parâmetro β_k .

O primeiro método de Gradientes Conjugados não-quadrático foi proposto por Hestenes-Stiefel em 1952 e a definição para β_k proposta consta na Tabela 2.1. É notável que a maioria das escolhas do parâmetro β_k tem como numerador $\|g_{k+1}\|^2$ ou $g_{k+1}^T y_k$ e como denominador $d_k^T y_k$, $-d_k^T g_k$ ou $\|g_k\|^2$. Portanto, destas 2 escolhas diferentes de numerador e 3 de denominador, nos resultam 6 diferentes combinações, que já estão na tabela. Além

desses, temos na tabela o proposto por Daniel, que não é recomendável para a maioria dos problemas pelo fato de utilizar a matriz Hessiana da função, e o proposto por Hager e Zhang. Uma característica importante de todas estas diferentes escolhas para β_k é que, se tomamos uma função quadrática convexa, todos os métodos são equivalentes quando a busca do tamanho de passo é exata. Mas para o caso não-quadrático, que é o tema de nosso interesse neste trabalho, o comportamento de cada escolha do parâmetro β_k varia consideravelmente, fato que mostraremos e desenvolveremos em futuras seções deste capítulo.

Ao ver que alguns métodos têm mais vantagens que outros, tanto computacionais como de convergência, tem-se desenvolvido métodos de Gradientes Conjugados híbridos, os quais misturam dois ou mais parâmetros β_k (dados na Tabela 2.1) para equilibrar as vantagens e as desvantagens, produzindo melhores algoritmos de Gradientes Conjugados. Em diferentes experimentos numéricos, tem-se encontrado que dentro dos melhores comportamentos estão o método proposto por Hager e Zhang [18] e o método híbrido entre Dai-Yuan [3] e Hestenes-Stiefel [19]. Nestes testes em geral é utilizada uma busca linear baseada nas condições de Wolfe.

Assim, para começar, vamos ver as diferentes condições usadas na busca linear, dentre as quais temos as condições de Wolfe e as de Zoutendijk, para depois fazer uma análise da convergência de algumas das diferentes versões do método para o caso não-quadrático.

2.2 Condições de Parada na Busca Linear

Em cada iteração do método de Gradientes Conjugados com busca linear inexata, o tamanho de passo α_k é escolhido para produzir uma solução aproximada para o problema

$$\min_{\alpha \geq 0} f(x_k + \alpha d_k)$$

em que a direção d_k satisfaz a condição de descida

$$g_k^T d_k < 0 \quad \text{para todo } k \geq 0. \quad (2.3)$$

Por outro lado, se temos que existe uma constante $c > 0$ tal que

$$g_k^T d_k < -c \|g_k\|^2 \quad \text{para todo } k \geq 0, \quad (2.4)$$

dizemos que a direção de busca satisfaz a condição suficiente de descida.

As condições de parada para a busca linear nos métodos de Gradientes Conjugados são frequentemente baseadas em alguma versão das condições de Wolfe, dentre as quais citamos:

Condições padrão de Wolfe [18]

$$\begin{aligned}f(x_k + \alpha_k d_k) - f(x_k) &\leq c_1 \alpha_k g_k^T d_k \\g_{k+1}^T d_k &\geq c_2 g_k^T d_k\end{aligned}$$

onde d_k é uma direção de descida e $0 < c_1 \leq c_2 < 1$.

Condições fortes de Wolfe [18]

$$\begin{aligned}f(x_k + \alpha_k d_k) - f(x_k) &\leq c_1 \alpha_k g_k^T d_k \\|g_{k+1}^T d_k| &\leq -c_2 g_k^T d_k\end{aligned}\tag{2.5}$$

onde d_k é uma direção de descida e $0 < c_1 \leq c_2 < 1$.

Condições generalizadas de Wolfe [18]

$$\begin{aligned}f(x_k + \alpha_k d_k) - f(x_k) &\leq c_1 \alpha_k g_k^T d_k \\c_2 g_k^T d_k \leq g_{k+1}^T d_k &\leq -c_3 g_k^T d_k\end{aligned}$$

onde d_k é uma direção de descida, $0 < c_1 < c_2 < 1$ e $c_3 \geq 0$. Se $c_2 = c_3$, estas recaem nas condições fortes de Wolfe.

Gostaríamos que fosse interrompida a busca linear nos algoritmos de Gradientes Conjugados quando as condições padrão de Wolfe fossem satisfeitas, mas para alguns algoritmos de Gradientes Conjugados alguma outra condição é necessária para garantir a convergência e melhorar a estabilidade. Por essa razão, também são introduzidas as condições aproximadas de Wolfe.

Condições aproximadas de Wolfe [18]

$$\begin{aligned} f(x_k + \alpha_k d_k) - f(x_k) &\leq c_1 \alpha_k g_k^T d_k \\ c_2 g_k^T d_k &\leq g_{k+1}^T d_k \leq (2c_1 - 1) g_k^T d_k \end{aligned}$$

onde d_k é uma direção de descida e $0 < c_1 < 1/2$ e $c_1 < c_2 < 1$.

As versões das condições padrão e generalizada de Wolfe são muito utilizadas para provar a convergência de algoritmos de Gradientes Conjugados. Já a versão aproximada tem sido útil para mostrar eficiência nas implementações de algoritmos de Gradientes Conjugados para os quais não existem uma teoria de convergência, mas apresentam bom desempenho prático [17].

A seguir, listamos duas hipóteses que são utilizadas em diversos resultados desta seção.

Hipótese de Lipschitz

Em alguma vizinhança \mathcal{N} do conjunto de nível $\mathcal{L} = \{x \in \mathbb{R}^n : f(x) \leq f(x_0)\}$, o gradiente de f é Lipschitz contínuo, isto é, existe uma constante $0 < L < \infty$ tal que

$$\|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\| \quad \forall x, y \in \mathcal{N}. \quad (2.6)$$

Hipótese de Limitação

O conjunto de nível \mathcal{L} é limitado, isto é, existe uma constante $0 < B < \infty$ tal que

$$\|x\| \leq B \quad \forall x \in \mathcal{L}. \quad (2.7)$$

Tais condições e hipóteses fornecem a estrutura necessária para garantir a convergência dos métodos de Gradientes Conjugados para o caso não-linear. Assim, obtemos o seguinte teorema, conhecido como condição de Zoutendijk, o qual é utilizado para provar a convergência dos métodos de nosso interesse.

Teorema 2.1. *Considerando uma iteração da forma (2.1) onde d_k satisfaz a condição de descida (2.3), α_k satisfaz a condição padrão de Wolfe e as Hipóteses de Lipschitz (2.6) e Limitação (2.7) são satisfeitas, então*

$$\sum_{k=0}^{\infty} \frac{(g_k^T d_k)^2}{\|d_k\|^2} < +\infty.$$

Demonstração. Ver [23], Teorema 3.2. □

2.3 Método de Fletcher-Reeves

Este método foi proposto para estender o método de Gradientes Conjugados para funções não quadráticas, fazendo para isso algumas mudanças no Algoritmo 1.1 [23, 18]. Uma delas é no tamanho do passo α_k (1.9), já que agora precisamos de uma busca linear que determine um minimizador aproximado da função não-quadrática f ao longo de d_k a partir de x_k . Estas mudanças são mostradas no Algoritmo 2.1.

Algoritmo 2.1. (*Gradientes Conjugados de Fletcher-Reeves*) Dado $x_0 \in \mathbb{R}^n$, calcular $f_0 = f(x_0)$ e $g_0 = \nabla f(x_0)$.

Definir $d_0 = -g_0$ e $k = 0$.

Enquanto $g_k \neq 0$,

Calcular α_k e $x_{k+1} = x_k + \alpha_k d_k$,

Avaliar $g_{k+1} = \nabla f(x_{k+1})$;

$$\beta_k^{FR} = \frac{g_{k+1}^T g_{k+1}}{g_k^T g_k}; \quad (2.8)$$

$$d_{k+1} = -g_{k+1} + \beta_k^{FR} d_k; \quad (2.9)$$

$$k = k + 1$$

Fim (Enquanto)

Observe que o algoritmo é econômico, pois a cada iteração somente uma avaliação da função objetivo e do seu gradiente são necessárias. Além disso, não são necessárias operações matriciais para o cálculo do passo β_k^{FR} , sendo apenas requeridas operações e armazenamento de vetores.

A dificuldade está presente no cálculo do parâmetro α_k , e dependendo dele, a direção de busca d_{k+1} pode resultar em não ser de descida. De (2.9) temos que

$$g_k^T d_k = -\|g_k\|^2 + \beta_{k-1}^{FR} g_k^T d_{k-1}. \quad (2.10)$$

Se a busca linear é exata, então o termo α_{k-1} vai ser o minimizador de f ao longo da direção d_{k-1} , obtendo assim que $g_k^T d_{k-1} = 0$, resultando de (2.10) que efetivamente $g_k^T d_k < 0$. Por outro lado, se a busca não é exata, o segundo termo de (2.10) pode dominar o primeiro, resultando $g_k^T d_k > 0$, onde não teríamos uma direção de descida. Logo, para evitar essa situação, podemos exigir que o tamanho de passo α_k cumpra as condições fortes de Wolfe, expostas na seção anterior.

Lema 2.2 ([23], Lema 5.6). *Suponha que o Algoritmo 2.1 (de Fletcher-Reeves) é implementado com um tamanho de passo α_k que satisfaz as condições fortes de Wolfe com $0 < c_2 < 1/2$. Então o método gera direções de descida d_k que satisfazem as seguintes desigualdades*

$$\frac{-1}{1-c_2} \leq \frac{g_k^T d_k}{\|g_k\|^2} \leq \frac{2c_2-1}{1-c_2}, \quad \text{para } k = 0, 1, \dots \quad (2.11)$$

Demonstração. Considere a função $t(\xi) = \frac{2\xi-1}{1-\xi}$, que satisfaz $t(0) = -1$ e $t(\frac{1}{2}) = 0$. Como $c_2 \in (0, \frac{1}{2})$ e t é monótona e crescente no intervalo $[0, \frac{1}{2}]$, temos que

$$-1 \leq \frac{2c_2-1}{1-c_2} < 0. \quad (2.12)$$

Se vale (2.11), por (2.12) temos que

$$\frac{g_k^T d_k}{\|g_k\|^2} \leq \frac{2c_2-1}{1-c_2} < 0,$$

logo, $g_k^T d_k < 0$, garantindo de imediato a condição de descida. Provaremos (2.11) por indução.

Para $k = 0$, temos que $d_0 = -g_0$, logo $\frac{g_0^T d_0}{\|g_0\|^2} = -1$. Como $c_2 \in (0, \frac{1}{2})$, um pouco de manipulação algébrica nos faz ver que

$$\frac{-1}{1-c_2} \leq -1 \leq \frac{2c_2-1}{1-c_2} < 0,$$

logo (2.11) vale para $k = 0$.

Vamos assumir como hipótese indutiva que para k a inequação (2.11) é válida. Multiplicando a igualdade (2.9) por g_{k+1}^T , vemos que

$$g_{k+1}^T d_{k+1} = -\|g_{k+1}\|^2 + \beta_k^{FR} g_{k+1}^T d_k,$$

a qual pode ser reescrita como

$$\frac{g_{k+1}^T d_{k+1}}{\|g_{k+1}\|^2} = -1 + \beta_k^{FR} \frac{g_{k+1}^T d_k}{\|g_{k+1}\|^2}.$$

Substituindo β_k^{FR} , dada em (2.8), nesta última equação e simplificando, chegamos a

$$\frac{g_{k+1}^T d_{k+1}}{\|g_{k+1}\|^2} = -1 + \frac{g_{k+1}^T d_k}{\|g_k\|^2}. \quad (2.13)$$

Usando a segunda condição forte de Wolfe (2.5), é possível obter a expressão

$$\frac{c_2 g_k^T d_k}{\|g_k\|^2} \leq \frac{g_{k+1}^T d_k}{\|g_k\|^2} \leq \frac{-c_2 g_k^T d_k}{\|g_k\|^2},$$

da qual subtraindo 1 e considerando o resultado (2.13) temos que

$$-1 + \frac{c_2 g_k^T d_k}{\|g_k\|^2} \leq \frac{g_{k+1}^T d_{k+1}}{\|g_{k+1}\|^2} \leq -1 + \frac{-c_2 g_k^T d_k}{\|g_k\|^2}.$$

Substituindo $g_k^T d_k / \|g_k\|^2$ pelo termo do lado esquerdo de (2.11), obtemos

$$-1 - \frac{c_2}{1 - c_2} \leq \frac{g_{k+1}^T d_{k+1}}{\|g_{k+1}\|^2} \leq -1 + \frac{c_2}{1 - c_2}.$$

Finalmente, simplificando obtemos o resultado desejado. \square

Na demonstração do resultado anterior, somente foi usada a segunda condição forte de Wolfe. No entanto, logo será vista a importância da primeira condição para a convergência do presente método.

Por enquanto, vamos supor que o método gera uma direção d_k ruim, isto é, que seja quase ortogonal ao gradiente g_k . Considerando θ_k como o ângulo entre d_k e a direção de máxima descida $-g_k$, sabemos que

$$\cos(\theta_k) = \frac{-g_k^T d_k}{\|g_k\| \|d_k\|}. \quad (2.14)$$

Então, se d_k é uma direção de busca ruim no sentido que $\theta_k \approx \frac{\pi}{2}$, temos que $\cos(\theta_k) \approx 0$.

Multiplicando as inequações (2.11) por $\|g_k\| / \|d_k\|$ obtemos que,

$$\frac{-\|g_k\|}{(1 - c_2)\|d_k\|} \leq \frac{g_k^T d_k \|g_k\|}{\|d_k\| \|g_k\|^2} \leq \frac{\|g_k\|(2c_2 - 1)}{\|d_k\|(1 - c_2)},$$

e usando (2.14), vemos que

$$\frac{\|g_k\|}{\|d_k\|} \frac{1 - 2c_2}{1 - c_2} \leq \cos(\theta_k) \leq \frac{\|g_k\|}{\|d_k\|} \frac{1}{1 - c_2}, \quad k = 0, 1, \dots, \quad (2.15)$$

de onde podemos deduzir que $\cos(\theta_k) \approx 0$, se e somente se $\|g_k\| \ll \|d_k\|$. Nesse sentido, $x_{k+1} \approx x_k$, já que d_k é uma direção com pouco decréscimo. E se $x_{k+1} \approx x_k$ acontecer, teremos que $g_{k+1} \approx g_k$ pela continuidade do gradiente, além disso, da fórmula (2.8), teremos que $\beta_k^{FR} \approx 1$. Usando esta aproximação junto com $\|g_{k+1}\| \approx \|g_k\| \ll \|d_k\|$, concluímos que $d_{k+1} \approx d_k$. Então um grande número de iterações improdutivas são necessárias para convergir à solução, o que é ruim, pois torna o método muito lento.

Além do método proposto por Fletcher-Reeves, Dai-Yuan [9] e Fletcher “Conjugate Descent” [13] propuseram algoritmos de Gradiente Conjugados cujas atualizações para β_k têm $\|g_{k+1}\|^2$ no numerador. Esse três métodos compartilham da importante característica de precisarem apenas da hipótese de Lipschitz em suas demonstrações de convergência.

O primeiro resultado de convergência global foi mostrado para o método de Fletcher-Reeves no ano 1970 por Zoutendijk [37], para o caso em que a busca do tamanho do passo seja exata. Já no ano de 1985 foi apresentado por Al-Baali o primeiro resultado de convergência global para a busca linear inexata, usando as condições fortes de Wolfe com $c_2 < \frac{1}{2}$. Ele provou que o método de Fletcher-Reeves gera direções suficientes de descida. Mais precisamente, mostrou que

$$\frac{1 - 2c_2 + c_2^{k+1}}{1 - c_2} \leq \frac{-g_k^T d_k}{\|g_k\|^2} \leq \frac{1 - c_2^{k+1}}{1 - c_2}, \text{ para todo } k \geq 0,$$

tendo como consequência a validade da condição de Zoutendijk. Em Liu e Storey [20], a prova de convergência de Al-Baali é estendida para o caso $c_2 = \frac{1}{2}$. Mais tarde, Dai e Yuan em [7] aprofundaram essa linha de pesquisa e mostraram que, em consecutivas iterações de Fletcher-Reeves, pelo menos uma iteração vai satisfazer a propriedade suficiente de descida, pois vale

$$\max \left\{ \frac{-g_k^T d_k}{\|g_k\|^2}, \frac{-g_{k-1}^T d_{k-1}}{\|g_{k-1}\|^2} \right\} \geq \frac{1}{2}.$$

Em [10], Dai e Yuan mostram que o método de Fletcher-Reeves, com as condições fortes de Wolfe, pode não produzir uma direção de descida quando $c_2 > \frac{1}{2}$, de onde concluímos que a limitação $c_2 \leq \frac{1}{2}$ deve ser imposta para garantir que as direções geradas sejam de descida.

Em relação ao método de Descida Conjugado proposto por Fletcher, vemos que está estreitamente relacionado com o método de Fletcher-Reeves, pois quando a busca linear é exata temos $\beta_k^{FR} = \beta_k^{CD}$. Diferentemente do método de Fletcher-Reeves, no método de Descida Conjugado a convergência é garantida sem a necessidade de que o parâmetro c_2 da busca forte de Wolfe satisfaça $c_2 \leq \frac{1}{2}$. Além disso, para uma busca linear com condições generalizadas de Wolfe, com $c_2 < 1$ e $c_3 = 0$, temos que $0 \leq \beta_k^{CD} \leq \beta_k^{FR}$. Por outro lado, para $c_2 \geq 1$ e $c_3 > 0$, Dai e Yuan apresentam em [8] um exemplo de um problema para o qual $\|d_k\|^2$ aumenta exponencialmente no método de Descida Conjugado de Fletcher e, por essa razão, o algoritmo converge a um ponto em que o gradiente não se anula. Em outras palavras, o método de Descida Conjugado pode não convergir a um ponto estacionário para uma busca linear forte de Wolfe.

Já o método de Dai e Yuan, desenvolvido em [9], é fundamentalmente diferente dos métodos de Fletcher-Reeves e de Descida Conjugado de Fletcher. O método de Dai e Yuan sempre vai gerar direções de descida se a busca linear cumpre as condições padrão

de Wolfe. O método converge globalmente quando a hipótese de Lipschitz é satisfeita. Dai em [3] analisa o método de Dai e Yuan e apresenta um resultado de convergência aplicável a qualquer método para o qual β_k possa ser escrito como

$$\beta_k = \frac{\Phi_{k+1}}{\Phi_k}. \quad (2.16)$$

Notemos que o método de Fletcher-Reeves corresponde à escolha de $\Phi_k = \|g_k\|^2$, e como $d_{k+1} = -g_{k+1} + \beta_k d_k$, podemos ver que β_k^{DY} pode ser escrito como,

$$\beta_k^{DY} = \frac{g_{k+1}^T g_{k+1}}{d_k^T (g_{k+1} - g_k)} = \frac{g_{k+1}^T (\beta^{DY} d_k - d_{k+1})}{d_k^T (\beta^{DY} d_k - d_{k+1} - g_k)},$$

de onde vemos que

$$(\beta_k^{DY})^2 d_k^T d_k - \beta^{DY} d_k^T d_{k+1} - \beta_k^{DY} d_k^T g_k = \beta^{DY} g_{k+1}^T d_k - g_{k+1}^T d_{k+1}.$$

Como $d_{k+1} = -g_{k+1} + \beta_k^{DY} d_k$, podemos cancelar alguns termos como segue,

$$\beta_k^{DY} (\cancel{\beta^{DY} d_k} - \cancel{d_{k+1}} - g_k - \cancel{g_{k+1}})^T d_k = -g_{k+1}^T d_{k+1}$$

$$\beta_k^{DY} (-g_k^T d_k) = -g_{k+1}^T d_{k+1}.$$

Finalmente, isolando β_k^{DY} , temos

$$\beta_k^{DY} = \frac{g_{k+1}^T d_{k+1}}{g_k^T d_k}.$$

Logo o método de Dai e Yuan tem a forma (2.16) com $\Phi_k = g_k^T d_k$. Desse modo, a ele também se aplica o seguinte teorema.

Teorema 2.3 ([18], Teorema 4.2). *Considere o algoritmo (2.1)-(2.2) em que β_k tem a forma (2.16) e d_k satisfaz a condição de descida (2.3) e a hipótese de Lipschitz. Se a condição de Zoutendijk é satisfeita, e além disso*

$$\sum_{k=0}^{\infty} \frac{(g_k^T d_k)^2}{\Phi_k^2} = \infty \quad \text{ou} \quad \sum_{k=0}^{\infty} \frac{\|g_k\|^2}{\Phi_k^2} = \infty \quad \text{ou} \quad \sum_{k=0}^{\infty} \prod_{i=1}^k \beta_i^{-2} = \infty,$$

então as iterações são globalmente convergentes.

Como um corolário deste resultado, o método de Dai e Yuan é globalmente convergente quando é implementado com busca linear padrão de Wolfe, pois $\Phi_k = g_k^T d_k$ e

$$\sum_{k=0}^N \frac{(g_k^T d_k)^2}{\Phi_k^2} = N + 1.$$

O método de Fletcher-Reeves também é globalmente convergente quando é implementado com busca linear forte de Wolfe e $c_2 \leq 1/2$, pois $\Phi_k = \|g_k\|^2$ e

$$\sum_{k=0}^N \frac{\|g_k\|^2}{\Phi_k^2} = N + 1.$$

2.4 Método de Polak-Ribière-Polyak e Variantes

O método de Polak-Ribière [29, 30] é uma variante importante do método de Fletcher-Reeves, onde o parâmetro β é atualizado pela expressão

$$\beta_k^{PR} = \frac{g_{k+1}^T (g_{k+1} - g_k)}{\|g_k\|^2} \quad (2.17)$$

em que $g_k = \nabla f(x_k)$. O algoritmo de Polak-Ribière é o Algoritmo 2.1 mudando somente o parâmetro β^{FR} por (2.17).

Para o caso em que a função objetivo f é estritamente convexa e quadrática e a busca linear é exata, vamos ter que os termos β dos dois algoritmos são iguais, isto é $\beta^{FR} = \beta^{PR}$, já que os gradientes g_{k+1} e g_k são mutuamente ortogonais pelo Teorema 1.6. No entanto, quando são aplicados para funções não-quadráticas gerais, com busca inexata, os algoritmos se comportam de maneiras bem diferentes.

Alguns estudos têm mostrado que o algoritmo de Polak-Ribière tende a ser mais eficiente entre os dois. Outro fato sobre este algoritmo é que as condições fortes de Wolfe não garantem que a direção d_k seja sempre de descida. Por isso, redefine-se o parâmetro β como $\beta_k^+ = \max\{\beta_k^{PR}, 0\}$, obtendo-se um novo algoritmo (PR+) que, com algumas adaptações das condições fortes de Wolfe, cumpre a propriedade de descida.

Além disso, repetimos a análise feita para o método de Fletcher-Reeves sobre o estudo do ângulo θ_k . Se considerarmos as mesmas condições, isto é, que a direção de busca d_k satisfaz que $\cos(\theta_k) \approx 0$ para algum k , e se o passo subsequente é pequeno, vemos que se $g_k \approx g_{k+1}$, então $\beta_k^{PR} \approx 0$ e da fórmula $d_{k+1} = -g_{k+1} + \beta_k^{PR}d_k$, encontramos que a nova direção de busca d_{k+1} estará perto da direção de máxima descida $-g_{k+1}$ e $\cos(\theta_{k+1}) \approx 1$. Temos então que, em vez de estagnar, o algoritmo executa um reinício depois de encontrar uma direção ruim, argumento que pode ser aplicado também para o Algoritmo PR+.

O algoritmo dado pela fórmula do parâmetro $\beta_k^{HS} = \frac{g_{k+1}^T y_k}{d_k^T y_k}$ de Hestenes e Stiefel é similar ao algoritmo de Polak-Ribière tanto em termos das suas propriedades de convergência teóricas quanto no seu desempenho prático. A fórmula de β_k^{HS} pode ser derivada exigindo que direções de busca consecutivas sejam conjugadas com respeito ao Hessiano médio no

segmento de reta $[x_k, x_{k+1}]$, definido como

$$G_k = \int_0^1 [\nabla^2 f(x_k + \tau \alpha_k d_k)] d\tau.$$

Levando em conta as propriedades teóricas do algoritmo de Fletcher-Reeves e as numéricas de Polak-Ribière, vários autores fazem combinações de betas das duas famílias para tentar obter algoritmos ainda melhores. Isso sugeriu a seguinte modificação para o método de Polak-Ribière, que tem um bom comportamento prático e chamaremos de FR-PR: para todo $k \geq 2$, definimos

$$\beta_k = \begin{cases} -\beta_k^{FR}, & \text{se } \beta_k^{PR} < -\beta_k^{FR} \\ \beta_k^{PR}, & \text{se } |\beta_k^{PR}| \leq \beta_k^{FR} \\ \beta_k^{FR}, & \text{se } \beta_k^{PR} > \beta_k^{FR}. \end{cases} \quad (2.18)$$

Entre as demais variantes estão também as propostas de Hager-Zhang [18] e Dai-Yuan [9]. Estas duas últimas escolhas garantem que d_k é uma direção de descida se o tamanho do passo α_k satisfaz as condições de Wolfe.

2.4.1 Outros Métodos com $g_{k+1}^T y_k$ no Numerador de β_k .

Dentre os métodos que têm $g_{k+1}^T y_k$ no numerador de β_k estão o método de Polak-Ribière-Polyak, de Hestenes-Stiefel e de Liu-Storey. Tais métodos possuem um recurso de reinicialização incorporado, pois, quando o passo $x_{k+1} - x_k$ é pequeno, o termo y_k no numerador tende a zero. Assim, β_k torna-se pequeno e a nova direção de busca d_k é próxima da direção de máxima descida. Nesse sentido, temos que estes métodos automaticamente ajustam β_k para evitar sucessivos passo ruins quando $x_{k+1} \approx x_k$, de modo que, em geral, o comportamento destes é melhor que o comportamento dos métodos com $\|g_{k+1}\|^2$ no numerador de β_k .

Sobre a convergência global, em [29] é estabelecido o estudo para o método de Polak-Ribière quando a função f é estritamente convexa e a busca linear é exata. Em [31], Powell provou que para uma função não linear geral, se

- (a) O tamanho do passo $s_k = x_{k+1} - x_k$ tende a zero,
- (b) A busca linear é exata,
- (c) A hipóteses de Lipschitz é satisfeita,

então o método de Polak-Ribière é globalmente convergente. Além disso, mais tarde Powell mostra em [32] um exemplo em \mathbb{R}^3 para o qual o método de Polak-Ribière com uma busca linear exata é infinitamente cíclico, não convergindo assim a um ponto estacionário. Portanto, a hipótese de que o tamanho do passo tenda a zero é necessária para a convergência. Sob a suposição de que a direção de busca é uma direção de descida, Yuan em [36] estabelece a convergência global do método de Polak-Ribière para funções estritamente convexas e busca linear de Wolfe. Em [2], para uma busca linear forte de Wolfe, Dai deu um exemplo no qual a função objetivo é estritamente convexa mas o método de Polak-Ribière gera uma direção de subida.

Em resumo, a convergência do método de Polak-Ribière para funções não lineares gerais é incerta. O exemplo de Powell mostrou que quando a função não é estritamente convexa então o método pode não convergir, mesmo que a busca linear seja exata. E o exemplo de Dai mostrou que mesmo que a função seja estritamente convexa, o método pode não gerar direções de descida com uma busca linear inexata. Baseado neste conhecimento adquirido Powell sugere em [32] uma modificação para a atualização do parâmetro para o método de Polak-Ribière, o qual já denotamos antes por PR+ e tem a forma

$$\beta_k^{PR+} = \max\{\beta_k^{PR}, 0\}.$$

Em [14], Gilbert e Nocedal provaram a convergência do método PR+. A análise feita por Gilbert e Nocedal é aplicada a uma classe de algoritmos de Gradientes Conjugados que cumprem a seguinte propriedade: considere um método da forma (2.1)-(2.2) e suponhamos que $0 < \gamma \leq \|g_k\| \leq \bar{\gamma}$, para todo $k \geq 0$, onde γ e $\bar{\gamma}$ são constantes positivas. Sob esta hipótese, dizemos que um método tem a Propriedade (\star) se existem constantes $b > 1$ e $\lambda > 0$ tais que, para todo k , $|\beta_k| \leq b$ e $\|s_k\| \leq \lambda$ implicam $|\beta_k| \leq \frac{1}{2b}$, onde $s_{k-1} := x_k - x_{k-1}$.

Teorema 2.4 ([18], Teorema 5.1). *Considere o método de Gradientes Conjugados (2.1)-(2.2) com as seguintes propriedades:*

- (a) $\beta_k \geq 0$, para todo k .
- (b) $\{x_k\} \subset \mathcal{L}$, onde $\mathcal{L} = \{x : f(x) \leq f(x_0)\}$ é limitada.
- (c) A condição suficiente de descida, $g_k^T d_k \leq -\sigma_3 \|g_k\|^2$ com $0 < \sigma_3 \leq 1$, é satisfeita.
- (d) A condição de Zoutendijk é satisfeita.

(e) A Propriedade (\star) se verifica.

Então $\liminf \|g_k\| = 0$.

Demonstração. Ver Teorema 4.3 em [14]. □

Como uma consequência deste resultado, temos que quando a direção de busca satisfaz a condição suficiente de descida e a busca linear padrão de Wolfe é implementada, o método de PR+ é globalmente convergente.

Em [6], Dai e Yuan exibem exemplos para mostrar que a Hipótese de Limitação é realmente necessária para obter a convergência global do Teorema 2.4. Por outro lado, a limitação $\beta_k \geq 0$ pode ser estendida para $\max\{\beta_k^{PR}, -\epsilon\}$ para qualquer $\epsilon > 0$. Já em [4], os autores mostram que a condição suficiente de descida no Teorema 2.4 pode ser estendida à condição de descida se na busca linear inexata são usadas as condições fortes de Wolfe.

O método PR+ foi introduzido para corrigir as falhas na convergência do método de Polak-Ribière quando é implementado com busca linear de Wolfe. Outra técnica para corrigir as falhas na convergência está em manter a fórmula para a atualização do parâmetro, mas modificar a busca linear. Por exemplo, Grippo e Lucidi em [?] propõem uma nova busca linear inspirada na de Armijo, com a qual provaram a convergência global do método de Polak-Ribière. Em outras abordagens de pesquisa, eles mostram em [10] que o método de Polak-Ribière é globalmente convergente quando a busca linear usa um tamanho do passo constante $\alpha_k = \eta < 1/4L$, sendo L a constante de Lipschitz para ∇f . Em [35], Sun e Zhang apresentam um resultado para a convergência global para a escolha de passo $\alpha_k = -\delta \frac{g_k^T d_k}{d_k^T Q_k d_k}$, onde Q_k é uma matriz definida positiva com menor autovalor $\nu_{min} > 0$, $\delta \in (0, \nu_{min}/L)$.

Por outro lado, temos para o método de Hestenes-Stiefel, que a condição de conjugação $d_{k+1}^T y_k = 0$ sempre é satisfeita, independentemente da busca linear. Para uma busca linear exata, temos que $\beta_k^{HS} = \beta_k^{PR}$. Além disso, as propriedades de convergência do método de Hestenes-Stiefel são similares às propriedades do método de Polak-Ribière. Em particular, pelo exemplo de Powell em [32], o método de Hestenes-Stiefel com busca linear exata pode não convergir para uma função não quadrática geral. Isto pode ser evitado se as direções de busca satisfazem a condição suficiente de descida e se as condições padrão de Wolfe são implementadas. Com isso, o método de Hestenes-Stiefel cumpre a Propriedade (\star) , como

acontece com o método de Polak-Ribière. Definindo

$$\beta_k^{HS+} = \max \{ \beta_k^{HS}, 0 \},$$

pelo Teorema 2.4 temos que o método de HS+ é globalmente convergente.

Finalmente, o método de Liu-Storey é idêntico ao método de Polak-Ribière se a busca linear é exata. Por outro lado, embora não se tenha uma extensa pesquisa sobre escolhas para a atualização do parâmetro, espera-se que as técnicas usadas para a análise do método de Polak-Ribière também sejam aplicáveis ao método de Liu-Storey.

2.5 Resultados de Convergência

Ao contrário do método de Gradientes Conjugados quadráticos, cujas propriedades de convergência são bem entendidas e que é conhecido por ser ótimo, o caso não quadrático possui propriedades de convergência muitas vezes surpreendentes. Nesta seção, vamos apresentar algumas dessas propriedades para os métodos de Fletcher-Reeves e de Polak-Ribière.

Estamos supondo que valem as hipóteses de Limitação e de Lipschitz, as quais implicam que existe uma constante γ tal que

$$\|g\| \leq \gamma, \text{ para todo } x \in \mathcal{L} = \{x : f(x) \leq f(x_0)\}, \quad (2.19)$$

assim como também a validade do teorema de Zoutendijk, onde teremos que

$$\sum_{k=0}^{\infty} \cos^2(\theta_k) \|g_k\|^2 < \infty. \quad (2.20)$$

Antes de provarmos a convergência global do método de Fletcher-Reeves, apresentaremos um pequeno Lema, de nossa autoria, que surgiu para corrigir um possível erro encontrado na demonstração do Teorema 5.7 de [23].

Lema 2.5. *Seja $c_3 \geq 1$ e considere a sequência $\{\psi_k\}_{k=1}^{\infty}$ dada por*

$$\psi_1 = c_3 \|g_1\|^2 + \beta_0^2 \|g_0\|^2, \quad \psi_2 = c_3 \|g_2\|^2 + c_3 \beta_1^2 \|g_1\|^2 + \beta_1^2 \beta_0^2 \|g_0\|^2,$$

$$\psi_3 = c_3 \|g_3\|^2 + c_3 \beta_2^2 \|g_2\|^2 + c_3 \beta_2^2 \beta_1^2 \|g_1\|^2 + \beta_2^2 \beta_1^2 \beta_0^2 \|g_0\|^2,$$

$$\psi_4 = c_3 \|g_4\|^2 + c_3 \beta_3^2 \|g_3\|^2 + c_3 \beta_3^2 \beta_2^2 \|g_2\|^2 + c_3 \beta_3^2 \beta_2^2 \beta_1^2 \|g_1\|^2 + \beta_3^2 \beta_2^2 \beta_1^2 \beta_0^2 \|g_0\|^2, \text{ etc,}$$

onde $\beta_k = \|g_{k+1}\|^2/\|g_k\|^2$. Então vale

$$\psi_k \leq c_3 \|g_k\|^4 \sum_{j=0}^k \|g_j\|^{-2}. \quad (2.21)$$

Demonstração. A prova segue por indução sobre k . Para $k = 1$, o lado direito de (2.21) é $c_3 \|g_1\|^2 + c_3 \|g_1\|^4 \|g_0\|^{-2}$. A definição de β_0 em ψ_1 e o fato que $c_3 \geq 1$ garantem o resultado. Suponhamos que a desigualdade vale para k . Da definição de ψ_k podemos ver que

$$\psi_{k+1} = c_3 \|g_{k+1}\|^2 + \beta_k^2 \psi_k.$$

Da hipótese indutiva e definição de β_k , temos que

$$\psi_{k+1} \leq c_3 \|g_{k+1}\|^2 + c_3 \frac{\|g_{k+1}\|^4}{\|g_k\|^4} \|g_k\|^4 \sum_{j=0}^k \|g_j\|^{-2} = c_3 \|g_{k+1}\|^4 \sum_{j=0}^{k+1} \|g_j\|^{-2},$$

o que conclui a demonstração. \square

Teorema 2.6 (Al-Baali [23], Teorema 5.7). *Suponha que as hipóteses (2.6) e (2.7) são válidas e o algoritmo de Fletcher-Reeves é implementado com busca linear que cumpra as condições fortes de Wolfe com $0 < c_1 < c_2 < \frac{1}{2}$. Então*

$$\liminf_{k \rightarrow \infty} \|g_k\| = 0. \quad (2.22)$$

Demonstração. A prova será por contradição. Assim, suponhamos que o oposto de (2.6) é válido, ou seja, existe $\epsilon > 0$ tal que para $k = 0, 1, \dots$, tem-se

$$\|g_k\| \geq \epsilon. \quad (2.23)$$

Agora, sob a condição de Zoutendijk e a desigualdade à esquerda de (2.15), vemos que

$$\sum_{k=0}^{\infty} \frac{\|g_k\|^4}{\|d_k\|^2} < \infty. \quad (2.24)$$

Usando (2.5) e (2.11), obtemos

$$|g_k^T d_{k-1}| \leq -c_2 g_{k-1}^T d_{k-1} \leq \frac{c_2}{1 - c_2} \|g_{k-1}\|^2.$$

Por (2.8) e (2.9), podemos ver que,

$$\begin{aligned} \|d_k\|^2 &= (-g_k + \beta_{k-1}^{FR} d_{k-1})^T (-g_k + \beta_{k-1}^{FR} d_{k-1}) \\ &= \|g_k\|^2 - 2\beta_{k-1}^{FR} g_k^T d_{k-1} + (\beta_{k-1}^{FR})^2 \|d_{k-1}\|^2 \\ &\leq \|g_k\|^2 + 2\beta_{k-1}^{FR} |g_k^T d_{k-1}| + (\beta_{k-1}^{FR})^2 \|d_{k-1}\|^2 \\ &\leq \|g_k\|^2 + \frac{2c_2}{1 - c_2} \beta_{k-1}^{FR} \|g_{k-1}\|^2 + (\beta_{k-1}^{FR})^2 \|d_{k-1}\|^2 \\ &= \|g_k\|^2 + \frac{2c_2}{1 - c_2} \frac{\|g_k\|^2}{\|g_{k-1}\|^2} \|g_{k-1}\|^2 + (\beta_{k-1}^{FR})^2 \|d_{k-1}\|^2 \\ &= \|g_k\|^2 \left(\frac{1 + c_2}{1 - c_2} \right) + (\beta_{k-1}^{FR})^2 \|d_{k-1}\|^2. \end{aligned}$$

Aplicando várias vezes esta relação e definindo

$$c_3 = \frac{1 + c_2}{1 - c_2} \geq 1,$$

temos que

$$\|d_k\|^2 \leq \psi_k, \quad (2.25)$$

onde a definição de ψ_k é dada no Lema 2.5. Por este mesmo lema e relação (2.25), obtemos

$$\|d_k\|^2 \leq c_3 \|g_k\|^4 \sum_{j=0}^k \|g_j\|^{-2}. \quad (2.26)$$

Assim, de (2.19), (2.23) e (2.26), vemos que

$$\|d_k\|^2 \leq \frac{c_3 \gamma^4}{\epsilon^2} (k + 1),$$

o qual implica que

$$\sum_{k=0}^{\infty} \frac{1}{\|d_k\|^2} \geq \frac{\epsilon^2}{c_3 \gamma^4} \sum_{k=0}^{\infty} \frac{1}{k + 1}. \quad (2.27)$$

Por outro lado, usando (2.23) e (2.24) temos que

$$\sum_{k=1}^{\infty} \frac{1}{\|d_k\|^2} < \infty.$$

No entanto, se usamos esta inequação com (2.27), obtemos que $\sum_{k=1}^{\infty} 1/k < \infty$, o qual é um absurdo. Temos assim que (2.23) não se cumpre e (2.22) é provada, concluindo a demonstração do Teorema. \square

Este resultado de convergência global pode ser estendido para qualquer escolha de β_k que satisfaça $|\beta_k| \leq |\beta_k^{PR}|$, em particular para o método híbrido de FR-PR dado em (2.18).

Em geral, se podemos mostrar que existem constante $c_4, c_5 > 0$ tais que

$$\cos(\theta_k) \geq c_4 \frac{\|g_k\|}{\|d_k\|} \quad \text{e} \quad \frac{\|g_k\|}{\|d_k\|} \geq c_5 > 0, \quad \text{para } k = 1, 2, \dots$$

segue de (2.20) que

$$\lim_{k \rightarrow \infty} \|g_k\| = 0.$$

De fato, este resultado pode ser estabelecido para o Método de Polak-Ribière supondo que f seja uma função estritamente convexa e que a busca linear usada seja exata. Mas para o caso de funções gerais, nem sempre é possível provar este resultado como no Teorema de Al-Baali para o Algoritmo de Fletcher-Reeves. O seguinte resultado mostra que o método de Polak-Ribière pode ser infinitamente cíclico.

Teorema 2.7 ([23], Teorema 5.8). *Considere o método de Polak-Ribière com uma busca linear “ideal” (aqui “ideal” significa que a busca linear retorna um valor α_k que é o primeiro ponto estacionário positivo para a função $f(x_k + \alpha d_k)$). Existem uma função contínua e duas vezes diferenciável $f : \mathbb{R}^3 \rightarrow \mathbb{R}$ e um ponto inicial $x_0 \in \mathbb{R}^3$ tais que a sequência de gradientes $\{\|g_k\|\}$ é limitada longe de zero.*

Demonstração. A prova deste resultado é bastante complexa e pode ser vista em [32]. \square

O resultado acima é interessante, uma vez que o tamanho do passo assumido na tese pode ser aceito por qualquer um dos algoritmos de busca linear atualmente usados. No caso de busca linear ideal, isso acontece apenas se $\beta_k < 0$, de modo que a análise sugere o método de PR+, no qual redefinimos β_k como zero sempre que se torna negativo. Mencionamos anteriormente que uma estratégia de busca linear baseada em uma ligeira modificação das condições de Wolfe garantem que todas as direções de busca geradas pelo Algoritmo PR+ são direções de descida. Usando estes fatos, é possível provar convergência global como o Teorema 2.6 para o Algoritmo PR+. Uma propriedade atrativa das fórmulas de β_k para os métodos de Hager-Zhang e de Dai-Yuan é que a convergência global pode ser estabelecida sem introduzir modificações para a busca linear baseada nas condições de Wolfe.

Capítulo 3

Um Método de Gradientes Conjugados com Direções Baseadas na Atualização de BFGS

Como visto no capítulo anterior, várias possibilidades foram consideradas na literatura para a atualização do parâmetro β_k que define a direção de busca em algoritmos de Gradientes Conjugados para funções não quadráticas. Uma linha de pesquisa nesse sentido é a que tenta incorporar ideias trazidas do bem sucedido algoritmo Quase-Newton com atualização BFGS para minimização irrestrita. Para exemplificar essa importante classe de algoritmos, neste capítulo, estudaremos a pesquisa desenvolvida recentemente por Dai e Kou [5] que, baseando-se em diversos trabalhos anteriores, introduzem um novo algoritmo de Gradientes Conjugados, juntamente com uma modificação nas condições de Wolfe que visa deixá-lo mais eficiente.

3.1 Gradientes Conjugados Baseados na Atualização BFGS

A ideia de incorporar elementos da atualização de BFGS no método de Gradientes Conjugados não é recente. Inspirados nela, em particular nos trabalhos de Perry [28] e Shanno [34], Dai e Kou [5] apresentam uma nova família de métodos de Gradiente Conjugados baseados no método de BFGS sem memória autoescalado. A direção de busca, neste caso,

é definida por

$$d_{k+1}^{PS} = -H_{k+1}g_{k+1}, \quad (3.1)$$

onde estamos considerando a matriz H_{k+1} como a matriz resultante da fórmula de atualização BFGS da matriz escalonada $\frac{1}{\tau_k}I$, sendo τ_k o parâmetro de escalamento, da forma

$$\begin{aligned} H_{k+1} &= \left(I - \frac{s_k y_k^T}{s_k^T y_k} \right) \left(\frac{1}{\tau_k} I \right) \left(I - \frac{y_k s_k^T}{s_k^T y_k} \right) + \frac{s_k s_k^T}{s_k^T y_k} \\ &= \frac{1}{\tau_k} \left(I - \frac{s_k y_k^T + y_k s_k^T}{s_k^T y_k} \right) + \left(1 + \frac{1}{\tau_k} \frac{\|y_k\|^2}{s_k^T y_k} \right) \frac{s_k s_k^T}{s_k^T y_k}. \end{aligned} \quad (3.2)$$

Substituindo (3.2) em (3.1) e lembrando que $\alpha_k d_k = s_k$, vemos que

$$\begin{aligned} d_{k+1}^{PS} &= - \left[\frac{1}{\tau_k} \left(I - \frac{s_k y_k^T + y_k s_k^T}{s_k^T y_k} \right) + \left(1 + \frac{1}{\tau_k} \frac{\|y_k\|^2}{s_k^T y_k} \right) \frac{s_k s_k^T}{s_k^T y_k} \right] g_{k+1} \\ &= -g_{k+1} + \left[\frac{g_{k+1}^T y_k}{s_k^T y_k} - \left(\tau_k + \frac{\|y_k\|^2}{s_k^T y_k} \right) \frac{g_{k+1}^T s_k}{s_k^T y_k} \right] s_k + \frac{g_{k+1}^T s_k}{s_k^T y_k} y_k \\ &= -g_{k+1} + \left[\frac{g_{k+1}^T y_k}{d_k^T y_k} - \left(\tau_k + \frac{\|y_k\|^2}{s_k^T y_k} \right) \frac{g_{k+1}^T s_k}{d_k^T y_k} \right] d_k + \frac{g_{k+1}^T s_k}{s_k^T y_k} y_k. \end{aligned} \quad (3.3)$$

Para que a expressão acima se converta na atualização clássica para algoritmos de Gradientes Conjugados, ou seja, $d_{k+1} = -g_{k+1} + \beta_k d_k$, alguns autores simplesmente desprezam o último termo em (3.3). No entanto, em [5] os autores fornecem uma maneira mais razoável de lidar com este termo, através de uma aproximação por quadrados mínimos. Para isso, consideram a variedade $S_{k+1} = \{-g_{k+1} + \beta d_k, \beta \in \mathbb{R}\}$. A ideia é escolher como direção o vetor em S_{k+1} que esteja o mais próximo possível de d_{k+1}^{PS} , ou seja

$$d_{k+1}^{PS} = \arg \min \{ \|d - d_{k+1}^{PS}\|_2 : d \in S_{k+1} \}.$$

Se escrevermos $d_{k+1}^{PS} = -g_{k+1} + Ad_k + By_k$, com $A = \frac{g_{k+1}^T y_k}{d_k^T y_k} - \left(\tau_k + \frac{\|y_k\|^2}{s_k^T y_k} \right) \frac{g_{k+1}^T s_k}{d_k^T y_k}$ e $B = \frac{g_{k+1}^T s_k}{s_k^T y_k}$, então,

$$\min_{d \in S_{k+1}} \{ \|d - d_{k+1}^{PS}\|_2 \} = \min_{\beta \in \mathbb{R}} \{ \|\beta d_k - Ad_k - By_k\|_2 \} = \min_{\beta \in \mathbb{R}} \{ \|By_k - (\beta - A)d_k\|_2 \}.$$

Uma vez que a solução do problema $\min_{\zeta \in \mathbb{R}} \|y_k - \zeta d_k\|$ é $\zeta = d_k^T y_k / \|d_k\|^2$, vemos que,

$$\beta_k = A + B \frac{d_k^T y_k}{\|d_k\|^2} = A + \frac{g_{k+1}^T s_k}{d_k^T y_k} \frac{s_k^T y_k}{s_k^T s_k},$$

de onde

$$\beta_k(\tau_k) = \frac{g_{k+1}^T y_k}{d_k^T y_k} - \left(\tau_k + \frac{\|y_k\|^2}{s_k^T y_k} - \frac{s_k^T y_k}{s_k^T s_k} \right) \frac{g_{k+1}^T s_k}{d_k^T y_k}, \quad (3.4)$$

e com a expressão para β_k fornecida acima, calculamos a direção de busca d_{k+1}^{PS} através da relação padrão em métodos de Gradientes Conjugados

$$d_{k+1}^{PS} = -g_{k+1} + \beta_k(\tau_k) d_k. \quad (3.5)$$

É possível encontrar na literatura diferentes sugestões para o valor do parâmetro τ_k . Dentre elas, as mais relevantes segundo os autores são τ_k^B , devida a Oren e Luenberger [26, 24], τ_k^H sugerida por Oren e Spedicato [25], e os truncamentos $\bar{\tau}_k^B$ e $\bar{\tau}_k^H$ sugeridos por Al-Baali [1], de expressões

$$\tau_k^B = \frac{s_k^T y_k}{\|s_k\|^2}, \quad (3.6)$$

$$\tau_k^H = \frac{\|y_k\|^2}{s_k^T y_k}, \quad (3.7)$$

$$\bar{\tau}_k^B = \min \left\{ 1, \frac{s_k^T y_k}{\|s_k\|^2} \right\}, \quad (3.8)$$

$$\bar{\tau}_k^H = \min \left\{ 1, \frac{\|y_k\|^2}{s_k^T y_k} \right\}. \quad (3.9)$$

Em [5] são apresentados diversos resultados sobre a condição suficiente de descida (2.4) para a nova proposta.

Lema 3.1. *Para a família de métodos de Gradientes Conjugados (2.1), (3.5) e (3.4), se $d_k^T y_k \neq 0$, sempre vamos ter que*

$$-d_{k+1}^T g_{k+1} \geq \min \left(\gamma_k, \frac{3}{4} \right) \|g_{k+1}\|^2.$$

Demonstração. Ver [5], Lema 2.1. □

Lema 3.2. *Assumamos que a função f seja limitada inferiormente e que satisfaça a Hipótese de Lipschitz. Consideremos a família de métodos de Gradientes Conjugados (2.1), (3.5) e (3.4). Se τ_k é escolhido como um dos τ_k^H , τ_k^B , $\bar{\tau}_k^H$ e $\bar{\tau}_k^B$, e se $d_k^T y_k \neq 0$, temos que*

$$-g_{k+1}^T d_{k+1} \geq c \|g_{k+1}\|^2,$$

para alguma constante positiva $c > 0$.

Demonstração. Ver [5], Lema 2.2. □

Para deixar o Lema mais geral, os autores consideram uma combinação convexa entre τ_k^H e τ_k^B , assim

$$\tau_k = \nu \frac{\|y_k\|^2}{s_k^T y_k} + (1 - \nu) \frac{s_k^T y_k}{\|s_k\|^2}, \quad (3.10)$$

com $\nu \in [0, 1]$.

Lema 3.3. *Assumamos que a função f seja limitada inferiormente e que satisfaça a Hipótese de Lipschitz. Consideremos a família de métodos de Gradientes Conjugados (2.1), (3.5) e (3.4), onde τ_k é da forma (3.10) com $\nu \in [0, 1]$. Se $d_k^T y_k \neq 0$, temos que*

$$-g_{k+1}^T d_{k+1} \geq \frac{3}{4} \|g_{k+1}\|^2.$$

Demonstração. Ver [5], Lema 2.3. □

Powell em [32] apresentou um contraexemplo mostrando que o método de Polak-Ribière com busca linear exata pode não convergir para funções não quadráticas. Como $\beta_k(\tau_k) = \beta_k^{PR}$ se $g_{k+1}^T d_k = 0$, este mesmo contraexemplo também pode ser usado para mostrar que o método (2.1) e (3.5) com β_k dado por (3.4) não necessariamente converge para funções gerais. De modo semelhante à proposta de Gilbert e Nocedal [14], na qual provaram a convergência global do método de Polak-Ribière para funções gerais impondo a restrição $\beta_k \geq 0$, Dai e Kou propuseram em [5] a substituição de (3.4) por

$$\beta_k^+(\tau_k) = \max \left\{ \beta_k(\tau_k), \eta \frac{g_{k+1}^T d_k}{\|d_k\|^2} \right\}, \quad (3.11)$$

onde $\eta \in [0, 1]$. Segundo os autores, este truncamento é mais interessante que simplesmente tomar o máximo entre 0 e β_k pois, dentre outros motivos, aproveita nos reinícios alguma informação de curvatura obtida em iterações anteriores ao invés de tomar apenas $d_{k+1} = -g_{k+1}$. Com essa nova proposta para β_k , também é possível provar a condição suficiente de descida, conforme o lema a seguir.

Lema 3.4. *Assumamos que a função f seja limitada inferiormente e que satisfaça a Hipótese de Lipschitz. Consideremos a família de métodos de Gradientes Conjugados (2.1), (3.5) e (3.4), onde $\beta_k(\tau_k)$ é substituído com $\beta_k^+(\tau_k)$ em (3.11) e τ_k é escolhido para ser qualquer entre τ_k^H , τ_k^B , $\bar{\tau}_k^H$ e $\bar{\tau}_k^B$. Se $d_k^T y_k \neq 0$ temos que*

$$-g_{k+1}^T d_{k+1} \geq \bar{c} \|g_{k+1}\|^2$$

para alguma constante $\bar{c} > 0$.

Demonstração. Ver [5], Lema 2.4. □

3.2 Busca de Wolfe Melhorada

Em otimização irrestrita, sabemos que a escolha da direção de busca e da busca linear são decisões importantes para o bom desempenho de um algoritmo. Nesta seção, apresentaremos uma proposta de Dai e Kou [5] para melhorar a busca linear de Wolfe tradicional, que

ao mesmo tempo permite evitar as desvantagens numéricas do critério de Wolfe tradicional e conserva as propriedades teóricas.

A primeira etapa do estudo se concentra na escolha do passo inicial, ou seja, o passo no qual o sub-algoritmo que faz a busca linear será iniciado a cada iteração. Denotaremos por ϕ_k a função de uma dimensão dada por

$$\phi_k(\alpha) = f(x_k + \alpha d_k), \quad \alpha \geq 0$$

e por $q(\phi_k(a), \phi'_k(a), \phi_k(b))$ a quadrática unidimensional que interpola ϕ_k nos pontos a e b , no sentido que $q(\phi_k(a), \phi'_k(a), \phi_k(b))(a) = \phi_k(a)$, $q(\phi_k(a), \phi'_k(a), \phi_k(b))'(a) = \phi'_k(a)$ e $q(\phi_k(a), \phi'_k(a), \phi_k(b))(b) = \phi_k(b)$. Note que há um certo abuso de notação na definição da função q , porém decidimos manter a notação apresentada em [5].

Estudos sobre a melhor escolha para o tamanho de passo inicial $\alpha_k^{(0)}$ tentam normalmente evitar tamanhos de passos pequenos e passos que estejam muito longe do minimizador de ϕ_k . Nesse sentido, em [5] os autores propõem o seguinte algoritmo para escolher o tamanho do passo inicial.

Algoritmo 3.1 (Estratégia para escolher o tamanho do passo inicial).

Passo 0 Dados os parâmetros positivos ϵ_1 , ϵ_2 e ψ .

Passo 1 Faça $\alpha_k^{(0)} = \max\{\psi\alpha_{k-1}, -2|f_k - f_{k-1}|/d_k^T g_k\}$ calcule $\phi_k(\alpha_k^{(0)})$.

Passo 2 Se $\frac{|\phi_k(\alpha_k^{(0)}) - \phi_k(0)|}{\epsilon_1 + |\phi_k(0)|} \leq \epsilon_2$, faça $\alpha_k^{(0)} := \arg \min q(\phi_k(0), \phi'_k(0), \phi_k(\alpha_k^{(0)}))$.

No Algoritmo 3.1 acima, a condição

$$\frac{|\phi_k(\alpha_k^{(0)}) - \phi_k(0)|}{\epsilon_1 + |\phi_k(0)|} \leq \epsilon_2$$

é usada para garantir que os pontos $x_k + \alpha_k^{(0)} d_k$ e x_k não fiquem muito longe um do outro. Com isso a interpolação possivelmente trará uma boa aproximação para ϕ_k , logo podemos tomar seu minimizador como tamanho do passo inicial.

Uma vez escolhido o passo inicial, vamos discutir o novo critério de busca linear. Inicialmente, considere as condições de Wolfe

$$\phi_k(\alpha) \leq \phi_k(0) + c_1 \alpha \phi'_k(0), \tag{3.12}$$

$$\phi'_k(\alpha) \geq c_2 \phi'_k(0) \tag{3.13}$$

onde $0 < c_1 < c_2 < 1$.

Teoricamente, se f satisfaz as hipóteses de Lipschitz e de Limitação e d_k é uma direção de descida, temos que existe um tamanho de passo $\alpha_k > 0$ que satisfaz (3.12) e (3.13). Mas em experimentos computacionais pode ocorrer de (3.12) nunca ser satisfeita, devido a erros numéricos. Se $\alpha_k^* > 0$ é um minimizador exato de $\phi_k(\alpha)$ e se $\alpha_k^* d_k$ é pequeno, temos que $\phi_k(0) - \phi_k(\alpha_k^*) = O(\|\alpha_k^* d_k\|^2)$. Consequentemente, $\phi_k(\alpha^*)$ é muito próximo de $\phi_k(0)$ se $\|\alpha_k^* d_k\|$ for da ordem da raiz quadrada da precisão da máquina. Nesta situação, pode ocorrer de numericamente termos $\phi_k(\alpha) \geq \phi_k(0)$ para todo $\alpha_k > 0$, logo (3.12) nunca seria satisfeita nos cálculos computacionais. Esta falha da busca linear de Wolfe pode acontecer em uma vizinhança de qualquer minimizador local se a tolerância for muito pequena.

Para evitar o inconveniente numérico da busca linear de Wolfe exposta acima, Hager e Zhang [16] propõem uma variação das condições de Wolfe tradicionais na forma $c_2 \phi'_k(0) \leq \phi'_k(\alpha) \leq (2c_1 - 1) \phi'_k(0)$. Estas mudanças proporcionaram um bom desempenho prático ao método, mas não garantiram resultados teóricos de convergência. Baseando-se nessa ideia, em [5] é apresentado um novo critério com bons resultados teóricos e práticos. Considerando um parâmetro constante $\epsilon > 0$, uma sequência positiva $\{\eta_k\}$ que satisfaz $\sum_{k \geq 1} \eta_k < +\infty$, e tomando novamente os parâmetros σ e δ que satisfazem $0 < c_1 < c_2 < 1$, a condição (3.12) é substituída por

$$\phi_k(\alpha) \leq \phi_k(0) + \min\{\epsilon |\phi'_k(0)|, c_1 \alpha \phi'_k(0) + \eta_k\}. \quad (3.14)$$

Chamaremos de busca linear de Wolfe melhorada a que satisfaz (3.13) e (3.14). A condição (3.14) pode ser vista como uma extensão da condição (3.12), pois um tamanho de passo que satisfaz a segunda desigualdade satisfaz também a primeira. E como sabemos que, sob certas hipóteses, sempre há um tamanho de passo que satisfaz (3.12), o mesmo ocorre para a condição mais fraca (3.14). Podemos notar que o termo η_k permite que os valores da função tenham em algumas iterações um ligeiro aumento, o que é útil para evitar a desvantagem numérica do critério de Wolfe clássico.

Na sequência é descrito um procedimento detalhado para implementar a busca linear de Wolfe melhorada, sendo que o ponto x_k e a direção de descida d_k são dados.

Algoritmo 3.2 (Busca linear de Wolfe Melhorada).

Passo 0. Determine $\alpha_k^{(0)}$ pelo Algoritmo 3.1.

Faça $a_0 = 0$, $\phi_a = f(x_k)$, $\phi'_a = g_k^T d_k$ e $b_0 = M$, onde M é algum número grande.

Faça $t_1 = 1.0$, $t_2 = 0.1$, $\rho > 1$, e $i = 0$.

Passo 1. Avalie $\phi_k(\alpha_k^{(i)})$ e teste a condição (3.14).

Se (3.14) é satisfeito vá ao Passo 2.

Faça $b_i = \alpha_k^{(i)}$, $\phi_b = \phi_k(\alpha_k^{(i)})$, $\alpha^ = \arg \min q(\phi_a, \phi'_a, \phi_b)$ e $t_1 = 0.1t_1$.*

Escolha $\alpha_k^{(i+1)} := \min\{\max[\alpha^, a_i + t_1(b_i - a_i)], b_i - t_2(b_i - a_i)\}$.*

Faça $i = i + 1$ e vá ao Passo 1.

Passo 2. Avalie $\phi'(\alpha_k^{(i)})$ e teste a condição (3.13).

Se (3.13) é satisfeita, retorne $\alpha_k = \alpha_k^{(i)}$ e pare.

Faça $t_1 = 0.1$, $t_2 = 0.1t_2$. Se $b_i = M$, vá ao Passo 3.

Faça $a_i = \alpha_k^{(i)}$, $\phi_a = \phi(\alpha_k^{(i)})$, $\phi'_a = \phi'(\alpha_k^{(i)})$, e $\alpha^ = \arg \min q(\phi_a, \phi'_a, \phi_b)$.*

Escolha $\alpha_k^{(i+1)} := \min\{\max[\alpha^, a_i + t_1(b_i - a_i)], b_i - t_2(b_i - a_i)\}$.*

Faça $i = i + 1$, vá ao Passo 1.

Passo 3. Faça $a_i = \alpha_k^{(i)}$, $\phi_a = \phi(\alpha_k^{(i)})$, $\phi'_a = \phi'(\alpha_k^{(i)})$ e $\alpha_k^{(i+1)} = \rho\alpha_k^{(i)}$.

Faça $i = i + 1$, vá ao Passo 1.

Podemos notar que o algoritmo acima é similar mas não idêntico ao da busca linear de Wolfe tradicional. O tamanho do passo $\alpha_k^{(0)}$ é calculado pelo Algoritmo 3.1 e pertence ao intervalo $[a_0, b_0] = [0, M]$ onde M é um número grande. Então em cada intervalo $[a_i, b_i]$ é escolhido um novo tamanho de passo $\alpha_k^{(i+1)}$ como o minimizador da função interpolação quadrática $q(\phi_a, \phi'_a, \phi_b)$, mas evitando que esteja perto de algum dos extremos dos intervalos através dos fatores t_1 e t_2 . Se $\alpha_k^{(i+1)}$ satisfaz as condições (3.13) e (3.14), a busca linear é terminada com $\alpha_k = \alpha_k^{(i+1)}$. Se só satisfaz (3.14), atualizamos o intervalo $[a_i, b_i]$ por $[\alpha_k^{(i+1)}, b_i]$, caso contrário por $[a_i, \alpha_k^{(i+1)}]$. O procedimento vai produzir uma sequência de intervalos $[a_i, b_i]$ com a propriedade $[a_{i+1}, b_{i+1}] \subset [a_i, b_i]$ para todo i , tal que $|b_i - a_i| \rightarrow 0$, e

$$\phi_k(a_i) \leq \phi_k(0) + \min\{\epsilon|\phi_k(0)|, c_1 a_i \phi'(0) + \eta_k\} \text{ mas } \phi'_k(a_i) < c_2 \phi'_k(0),$$

$$\phi_k(b_i) > \phi_k(0) + \min\{\epsilon|\phi_k(0)|, c_1 b_i \phi'(0) + \eta_k\},$$

até que um tamanho de passo satisfatório seja encontrado com sucesso.

Um resultado importante desta busca linear é que esta conserva a condição de Zoutendijk, conforme enuncia o lema a seguir.

Lema 3.5. *Assumamos que a função f seja limitada inferiormente e que satisfaça a Hipótese de Lipschitz. Consideremos um método iterativo da forma (1.5), onde a direção d_k satisfaz $g_k^T d_k < 0$ e o tamanho de passo α_k satisfaz (3.13) e (3.14). Então temos que*

$$\sum_{k \geq 1} \frac{(g_k^T d_k)^2}{\|d_k\|^2} < \infty.$$

Demonstração. Ver ([5]), Lema 3.3. □

3.3 Algoritmo e Análise de Convergência

Nesta seção, apresentamos por fim o algoritmo proposto por Dai e Kou [5] e seus principais resultados teóricos. Quando o algoritmo alcança a k -ésima iteração, temos disponíveis pelo menos os quatro valores: $\phi_{k-1}(0) = f_{k-1}$, $\phi'_{k-1}(0) = g_{k-1}^T d_{k-1}$, $\phi_{k-1}(\alpha_{k-1}) = f_k$ e $\phi'(\alpha_{k-1}) = g_k^T d_{k-1}$. Fazemos então uma interpolação quadrática, calculando a função q_{k-1} que satisfaz às condições

$$q_{k-1}(0) = \phi_{k-1}(0), \quad q'_{k-1}(0) = \phi'_{k-1}(0) \quad \text{e} \quad q'_{k-1}(\alpha_{k-1}) = \phi'_{k-1}(\alpha_{k-1}).$$

Então, de modo semelhante à teoria de região de confiança (Ver página 97 de [33]), podemos usar a razão

$$r_{k-1} = \frac{\phi_{k-1}(0) - \phi_{k-1}(\alpha_{k-1})}{q_{k-1}(0) - q_{k-1}(\alpha_{k-1})} \quad (3.15)$$

para tentar medir o quão próximas localmente são as funções q_{k-1} e ϕ_{k-1} , o que ocorrerá se $r_{k-1} \approx 1$. Além disso, note que $q_{k-1}(0) = \phi_{k-1}(0) = f_{k-1}$, $\phi_{k-1}(\alpha_{k-1}) = f_k$ e, por cálculos diretos, $q_{k-1}(\alpha_{k-1}) = f_{k-1} + \frac{1}{2}\alpha_{k-1}(g_{k-1}^T d_{k-1} + g_k^T d_{k-1})$, logo a razão (3.15) pode ser simplificada como

$$r_{k-1} = \frac{2(f_k - f_{k-1})}{\alpha_{k-1}(g_{k-1}^T d_{k-1} + g_k^T d_{k-1})}. \quad (3.16)$$

Ao minimizar uma função qualquer, é comum que a função seja bastante não linear nas primeiras iterações, porém tenha um comportamento semelhante a uma quadrática próximo do minimizador. Por essa razão, os autores propõem utilizar o parâmetro r_k para decidir quando reiniciar a direção de busca, o que deve ocorrer se $r_k \approx 1$ consecutivas vezes. O algoritmo proposto é explicitado a seguir.

Algoritmo 3.3 (Novo Algoritmo de Gradiente Conjugados).

Passo 0. Dados $x_1 \in \mathbb{R}^n$, $\epsilon > 0$, $\epsilon_4 > 0$ e inteiros positivos *MaxRestart* e *MinQuad*.

Passo 1. Fixe $k := 1$, se $\|g_1\| \leq \epsilon$ pare.

Faça $d_1 = -g_1$, *IterRestart* := 0 e *InterQuad* := 0.

Passo 2. Calcule o tamanho do passo $\alpha_k > 0$ pelo Algoritmo 3.2.

Passo 3. Faça $x_{k+1} = x_k + \alpha_k d_k$. Se $\|g_{k+1}\| \leq \epsilon$ pare.

$IterRestart := IterRestart + 1$.

Calcule r_k como em (3.16). Se $|r_k - 1| \leq \epsilon_4$,

$IterQuad := IterQuad + 1$; senão $IterQuad := 0$.

Passo 4. Se $IterRestart = MaxRestart$ ou ($IterQuad = MinQuad$ e $IterQuad \neq IterRestart$), faça $d_{k+1} = -g_{k+1}$ e faça $IterRestart := 0$, $IterQuad := 0$, $k = k + 1$, e vá ao Passo 2.

Passo 5 Calcule β_k por (3.11) e $d_{k+1} = -g_{k+1} + \beta_k d_k$

$k := k + 1$ vá ao Passo 2.

Na sequência, apresentamos os principais resultados teóricos para o Algoritmo 3.3.

Teorema 3.6. *Assumamos que a função f seja limitada inferiormente e que satisfaça a Hipótese de Lipschitz, a direção de busca definida por (2.2), (3.5) e (3.4), com τ_k escolhido para ser qualquer entre (3.6), (3.7), (3.8) e (3.9), onde o tamanho do passo α_k é calculado pela busca linear satisfazendo (3.13) e (3.14).*

Se adicionalmente, f é uniformemente convexa, ou seja, existe uma constante $\mu > 0$ tal que

$$(\nabla f(x) - \nabla f(y))^T(x - y) \geq \mu \|x - y\|^2 \text{ para todo } x, y \in \mathbb{R}^n,$$

temos que

$$\lim_{k \rightarrow \infty} g_k = 0.$$

Demonstração. Ver ([5]), Teorema 4.2. □

Teorema 3.7. *Assumamos que a função f seja limitada inferiormente e que satisfaça a Hipótese de Lipschitz. Consideremos a família de métodos da forma (2.1), onde d_{k+1} é dado por (3.5) e (3.11) e o tamanho do passo é calculado pela busca linear de Wolfe melhorada, satisfazendo (3.14) e (3.13). Se a sequência gerada $\{x_k\}$ é limitada, e se τ_k é escolhido entre qualquer um de (3.6), (3.7), (3.8) e (3.9). O método converge no sentido que $\liminf_{k \rightarrow \infty} \|g_k\| = 0$.*

Demonstração. Ver ([5]), Teorema 4.4. □

Capítulo 4

Experimentos Numéricos

Neste capítulo, apresentamos os resultados numéricos obtidos com as implementações feitas para o algoritmo de Gradientes Conjugados para diferentes buscas lineares e escolhas do parâmetro β_k . Para deixar mais claros os detalhes da implementação, na Seção 4.1 apresentamos o pseudocódigo do algoritmo de Gradientes Conjugados que foi utilizado nos testes e listamos tanto os parâmetros β_k escolhidos como os valores de diversos parâmetros necessários ao algoritmo. Na Seção 4.2, descrevemos o banco de funções utilizado nos testes e discutimos a técnica de comparação usada. Finalmente na Seção 4.3 apresentamos, através de gráficos de perfil de desempenho, os resultados numéricos obtidos.

4.1 Algoritmo Implementado

Os experimentos consistiram, conforme iremos detalhar no decorrer do texto, em testar o desempenho do algoritmo de Gradientes Conjugados na resolução dos problemas da coleção proposta por Moré, Garbow e Hillstrom [22]. Foram testadas várias versões do parâmetro β_k que define a atualização da direção de busca, bem como três critérios de buscas lineares diferentes: os clássicos de Armijo e Wolfe Forte e também a busca de Wolfe Melhorada estudada no Capítulo 3. O algoritmo utilizado na implementação computacional foi o que se segue.

Algoritmo 4.1 (Algoritmo de Gradientes Conjugados).

Passo 0. Escolha $x_0 \in \mathbb{R}^n$, $\text{To1} > 0$ e inteiros positivos ItMax , FevalMax , GevalMax e AlphaMax . Fixe $\mathbf{k} = 0$. Calcule $f(x_0)$ e $d_0 = -g_0$, $\text{Geval} = 1$, $\text{Feval} = 1$.

Passo 1. Enquanto $k < \text{ItMax}$, $\text{Feval} < \text{FevalMax}$, $\text{Geval} < \text{GevalMax}$ e $\|g_k\| > \text{Tol}$,

Faça $k = k + 1$.

Calcule o passo α_k através de alguma busca linear inexata. Se o número de iterações internas nesta chamada do subalgoritmo de busca linear ultrapassar o valor AlphaMax , interrompa o algoritmo de Gradientes Conjugados.

Defina o novo ponto $x_{k+1} = x_k + \alpha_k d_k$.

Defina FevalBusca como o número de avaliações de funções realizadas dentro do subalgoritmo de busca linear para encontrar x_{k+1} e GevalBusca o número de avaliações de gradiente.

Faça $\text{Feval} = \text{Feval} + \text{FevalBusca}$ e $\text{Geval} = \text{Geval} + \text{GevalBusca}$.

Passo 2. Calcule o parâmetro β_k e defina a nova direção $d_{k+1} = -g_{k+1} + \beta_k d_k$.

Passo 3. Se $d_{k+1}^T g_{k+1} > 0$, faça $d_{k+1} = -g_{k+1}$. Volte ao Passo 1.

Alguns comentários sobre o algoritmo são importantes. As avaliações de funções são feitas dentro do subalgoritmo de busca linear uma vez que, a grosso modo, o algoritmo principal apenas calcula d_{k+1} e testa os critérios de parada. O teste $d_{k+1}^T g_{k+1} > 0$ se mostrou necessário durante os experimentos, pois em alguns casos a direção obtida não estava sendo numericamente de descida, o que foi corrigido reiniciando a direção de busca definindo-a novamente como a de máxima descida. O critério de parada mais desejado é que seja atingido $\|g_k\| \leq \text{Tol}$. Porém, o algoritmo também é interrompido quando pelo menos um dos quatro limitantes é atingido: o número máximo de iterações ItMax , o de avaliações de função FevalMax , o de avaliações de gradiente GevalMax ou o número máximo de iterações internas em cada chamada do subalgoritmo de busca linear, AlphaMax .

Os parâmetros β_k considerados nos testes estão listados na Tabela 4.1. Para cada escolha definimos uma sigla, para que fique mais simples referenciá-la nos experimentos. Lembremos que $\beta_k(\tau_k) = \frac{g_{k+1}^T y_k}{d_k^T y_k} - \left(\tau_k + \frac{\|y_k\|^2}{s_k^T y_k} - \frac{s_k^T y_k}{\|s_k\|^2} \right) \frac{g_{k+1}^T s_k}{d_k^T y_k}$.

Vários parâmetros são necessários para a execução do algoritmo. Listamos a seguir os valores utilizados, separados de acordo com a natureza do parâmetro. Quando o valor em questão vem de alguma referência, nós a citamos. Nos casos em que não conseguimos uma referência, escolhemos um valor arbitrário dentro do intervalo de definição do parâmetro.

BETA	NOME	SIGLA
1 $\beta_k^{FR} = \frac{\ g_{k+1}\ ^2}{\ g_k\ ^2}$	Fletcher e Reeves [12].	fr
2 $\beta_k^{HS} = \frac{g_{k+1}^T y_k}{d_k^T y_k}$	Hestenes e Stiefel [19].	hs
3 $\beta_k^{PR} = \frac{g_{k+1}^T y_k}{\ g_k\ ^2}$	Polak, Ribière e Polyak [30][29].	pr
4 $\beta_k^{PR+} = \max\{\frac{g_{k+1}^T y_k}{\ g_k\ ^2}, 0\}$	Truncamento Polak, Ribière e Polyak [32].	maxpr
5 $\beta_k^{LS} = \frac{g_{k+1}^T y_k}{-d_k^T g_k}$	Liu e Storey [20].	ls
6 $\beta_k^{CD} = \frac{\ g_{k+1}\ ^2}{-d_k^T g_k}$	Descida Conjugado [13].	dc
7 $\beta_k(\tau_k^H)$	Dai e Kou com $\tau_k^H = \frac{\ y_k\ ^2}{s_k^T y_k}$ [5].	dT1
8 $\beta_k(\tau_k^B)$	Dai e Kou com $\tau_k^B = \frac{s_k^T y_k}{\ s_k\ ^2}$ [5].	dT2
9 $\beta_k(\bar{\tau}_k^H)$	Dai e Kou com $\bar{\tau}_k^H = \min\{1, \frac{\ y_k\ ^2}{s_k^T y_k}\}$ [5].	mindT1
10 $\beta_k(\bar{\tau}_k^B)$	Dai e Kou com $\bar{\tau}_k^B = \min\{1, \frac{s_k^T y_k}{\ s_k\ ^2}\}$ [5].	mindT2
11 $\beta_k^N = \left(y_k - 2d_k \frac{\ y_k\ ^2}{d_k^T(y_k)}\right)^T \frac{g_{k+1}}{d_k^T y_k}$	Hager e Zhag [16].	hz
12 $\beta_k^{DY} = \frac{\ g_{k+1}\ ^2}{d_k^T y_k}$	Dai e Yuan [9].	dy
13 $\beta_k^+(\tau_k^H) = \max\{\beta_k(\tau_k^H), \eta \frac{g_{k+1}^T d_k}{\ d_k\ ^2}\}$	Dai e Kou com $\tau_k^H = \frac{\ y_k\ ^2}{s_k^T y_k}$ [5].	MdT1
14 $\beta_k^+(\tau_k^B) = \max\{\beta_k(\tau_k^B), \eta \frac{g_{k+1}^T d_k}{\ d_k\ ^2}\}$	Dai e Kou com $\tau_k^B = \frac{s_k^T y_k}{\ s_k\ ^2}$ [5].	MdT2
15 $\beta_k^+(\bar{\tau}_k^H) = \max\{\beta_k(\bar{\tau}_k^H), \eta \frac{g_{k+1}^T d_k}{\ d_k\ ^2}\}$	Dai e Kou com $\bar{\tau}_k^H = \min\{1, \frac{\ y_k\ ^2}{s_k^T y_k}\}$ [5].	MmindT1
16 $\beta_k^+(\bar{\tau}_k^B) = \max\{\beta_k(\bar{\tau}_k^B), \eta \frac{g_{k+1}^T d_k}{\ d_k\ ^2}\}$	Dai e Kou com $\bar{\tau}_k^B = \min\{1, \frac{s_k^T y_k}{\ s_k\ ^2}\}$ [5].	MmindT2

Tabela 4.1: Resumo das escolhas do parâmetro β_k testadas.

Parâmetros do algoritmo:

- $\text{To1} = 10^{-6}$, $\text{ItMax} = 5000$, $\text{FevalMax} = 10^6$, $\text{GevalMax} = 10^5$.

Parâmetros das Buscas Lineares:

- Armijo: $\delta = 0.5$, $\gamma = 0.38$. [27].
- Wolfe Forte: $c_1 = 0.38$, $c_2 = 0.9$ [27].
- Wolfe Melhorada: $\epsilon_1 = 10^{-3}$, $\epsilon_2 = 100$, $\psi = 5$, $\sigma = 0.9$, $\delta = 0.2$, $\rho = 5$, $\epsilon = 10^{-6}$, $M = 10^{10}$ e $\eta_k = \frac{1}{k^2}$ [5].

Parâmetros para β_k de [5]

- $\eta = 0.5$.

4.2 Metodologia de teste

Os problemas resolvidos em nossos experimentos foram retirados da coletânea organizada por Moré, Garbow e Hillstorn [22]. As funções objetivo deste trabalho têm a particularidade de serem somas de quadrados, isto é, são funções da forma

$$f : \mathbb{R}^n \rightarrow \mathbb{R}$$
$$x \rightarrow f(x) = \sum_{i=1}^m f_i(x)^2$$

onde m , n e $f_i(x)$ para $i = 1, 2, \dots, m$ são definidos para cada função. Há algumas funções para as quais os parâmetros n e m são arbitrários. Para escolher estes parâmetros nestes casos, utilizamos o seguinte critério. Se para alguma combinação de n e m a solução é dada explicitamente no artigo [22], essa combinação foi usada. Quando isso não ocorre ou quando a solução dada no artigo vale para quaisquer n e m , escolhemos $n = 20$ e tomamos m de modo a satisfazer a regra do problema. Se m puder tomar um valor arbitrário diferente de n , escolhemos $m = n$, tal como é feito em [27]. Ao todo, os 35 problemas de [22] foram testados, cada um com uma única combinação de n e m .

A técnica de comparação usada foi apresentada em [27], e a explicamos a seguir. Considere um problema fixo e seja $x^{(i)}$, $i = 1, 2, \dots, 16$, a solução encontrada para este problema pelo

i -ésimo algoritmo, ou seja, o Algoritmo 4.1 com a i -ésima escolha para β_k de acordo com a Tabela 4.1. Definimos

$$f_{\min} = \min_{i=1,\dots,16} \{f(x^{(i)})\},$$

e diremos que o j -ésimo algoritmo resolveu o problema se

$$f(x^{(j)}) \leq f_{\min} + 10^{-3}|f_{\min}| + 10^{-6}. \quad (4.1)$$

Assim sendo, para cada problema, consideramos que um algoritmo obteve sucesso quando encontrou, a menos de uma tolerância, o melhor valor de função entre todos os valores de função encontrados para os diferentes algoritmos testados. É importante observar que se o Algoritmo 4.1, para uma escolha específica de β_k e de problema, foi encerrado não com $\|g_k\| \leq \text{To1}$ mas, digamos, com $\text{Feval} \geq \text{FevalMax}$, ainda assim poderemos considerar que o problema foi resolvido se (4.1) for satisfeita. Esse certamente não será o caso se o algoritmo for interrompido com mensagem de erro (por exemplo, se ocorrer *overflow*), pois nessa situação definimos $f(x^{(i)}) = \infty$ para que o problema nunca seja considerado resolvido.

4.3 Resultados Numéricos

Nesta Seção, apresentamos os resultados numéricos para o algoritmo de Gradientes Conjugados, com os três critérios de busca linear (Armijo, Wolfe Forte e Wolfe Melhorada) e as 16 diferentes expressões de β_k (Tabela 4.1). Os testes computacionais foram realizados em Matlab versão 7.10.0.499(R2010a), em um computador Intel(R) Core(TM) i3-4005U CPU @ 1.70GHz, com frequência de 1.70GHz e memória de 6GB, e com sistema operacional Windows 10 Home Single Language.

A maior parte dos resultados teóricos apresentados no Capítulo 2 dependem da utilização das condições de Wolfe. No entanto, decidimos testar também a busca de Armijo por sua simplicidade e pra avaliar a importância da segunda condição de Wolfe (2.5). Por outro lado, para a busca de Wolfe Melhorada, até onde sabemos só há demonstrações de convergências se esta for utilizada dentro do Algoritmo (3.3) e com as escolhas de β_k apresentadas em [5]. Ainda assim, decidimos testar o Algoritmo 3.2, que encontra um passo satisfazendo a condição de Wolfe Melhorada, dentro do Algoritmo 4.1 para ver como esse critério se comporta na prática. Pelo mesmo motivo, testamos as escolhas para β_k

dentro do Algoritmo 4.1, mesmo perdendo a princípio as garantias de convergência. Como veremos, a ideia se mostrou acertada.

Comparamos as versões do algoritmo em termos de robustez e eficiência. Lembramos que um algoritmo é dito robusto quando resolve uma porcentagem grande dos problemas, enquanto é considerado o mais eficiente quando é mais rápido que os demais algoritmos testados. Essa rapidez pode ser medida em termos de número de iterações, número de avaliações de função, tempo computacional, etc. É importante observar que a robustez diz respeito apenas a cada algoritmo individualmente, enquanto a eficiência leva em consideração o desempenho de cada um em relação aos demais. A metodologia de comparação está baseada principalmente em gráficos de perfil de desempenho. Para um detalhamento sobre esses gráficos, bem como discussões sobre comparações de algoritmos, robustez, eficiência, etc, ver [33, 27].

Nas tabelas seguintes vamos apresentar os resultados obtidos em termos de eficiência e robustez, ou seja, a porcentagem dos problemas para os quais o algoritmo em questão foi o mais rápido e a porcentagem total de problemas resolvidos. A eficiência foi medida em termos de número de iterações e também de avaliações de funções. É importante mencionar que o número de avaliações de função é muito sensível à busca usada, já que temos buscas que avaliam a função várias vezes. A Tabela 4.2 e as Figuras 4.1 e 4.2 trazem os resultados obtidos pela busca de Armijo em termos de número de iterações e de avaliações de funções.

Notamos que, se a busca de Armijo é utilizada, em termos de robustez os betas **fr**, **dT1**, **dT2**, **mindT1**, **mindT2** e **hz** foram os melhores, resolvendo 82,86% dos problemas, seguidos por **pr**, **maxpr**, **ls**, **MdT2**, **MmindT1** e **MmindT2** que resolveram 80%. Os 4 betas restantes tiveram um desempenho inferior. Em termos de eficiência, destacamos o beta **mindT1**, que foi o mais rápido em termos de iterações para 28,57% dos problemas, e também **dT2** e **mindT2**, que foram os melhores em 25,71% dos casos. Pelos gráficos, é possível perceber um desempenho particularmente inferior para os betas **dc** e **hs**.

Procedendo da mesma forma para a busca de Wolfe Forte, obtemos a Tabela 4.3 e os gráficos de perfil de desempenho das Figuras 4.3 e 4.4.

É interessante compararmos os resultados para as buscas de Armijo e Wolfe Forte. Es-

	Beta	Eficiência (Iterações)	Eficiência (Avaliações)	Robustez
1	fr	17,14%	14,29%	82,86%
2	hs	11,43%	11,43%	71,43%
3	pr	8,57%	5,71%	80%
4	maxpr	8,57%	5,71%	80%
5	ls	14,29%	17,14%	80%
6	dc	11,43%	11,43%	68,57%
7	dT1	11,43%	5,71%	82,86%
8	dT2	25,71%	20%	82,86%
9	mindT1	28,57%	25,71%	82,86%
10	mindT2	25,71%	17,14%	82,86%
11	hz	11,43%	5,71%	82,86%
12	dy	14,29%	17,14%	77,15%
13	MdT1	5,71%	5,71%	71,43%
14	MdT2	8,57%	5,71%	80%
15	MmindT1	11,43%	8,57%	80%
16	MmindT2	14,29%	8,57%	80%

Tabela 4.2: Tabela de resultados para a busca de Armijo.

perávamos que a busca de Wolfe Forte tivesse um desempenho melhor em termos de robustez para quase todos os valores de β_k , uma vez que é com essa busca que a maior parte dos resultados teóricos se estabelece. No entanto, houve empate entre a quantidade de opções de β_k que tiveram um desempenho melhor com Wolfe Forte e as que se saíram melhor com Armijo apenas. Ainda assim, achamos que houve ganho, se observarmos por exemplo que o beta **mindT2**, o mais robusto para as duas buscas, saltou de 82,86% resolvidos com Armijo para 91,43% com Wolfe Forte. O beta com melhor eficiência foi o **MmindT1**, que foi o mais rápido em termos de iterações para 34,29% dos problemas. Os gráficos de perfil de desempenho sugerem mais uma vez um desempenho insuficiente do beta **hs**.

Encerrando essa etapa dos testes, apresentamos os resultados para a busca de Wolfe melhorada na Tabela 4.4 e Figuras 4.5 e 4.6.

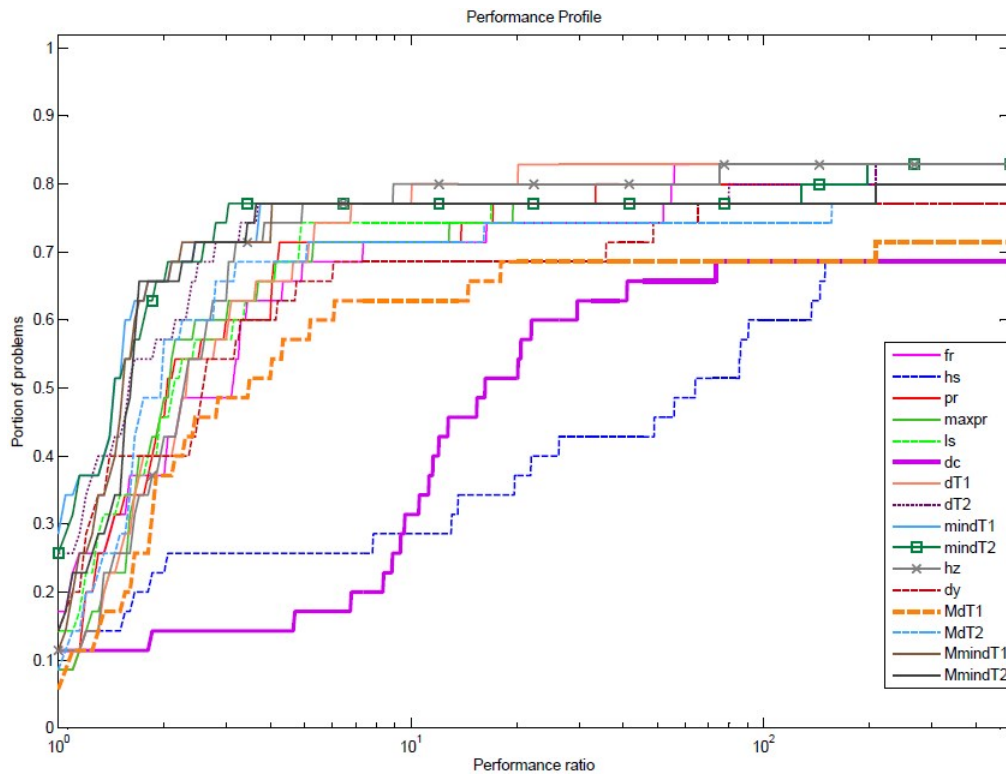


Figura 4.1: Perfil de desempenho em relação ao número de iterações para a busca de Armijo.

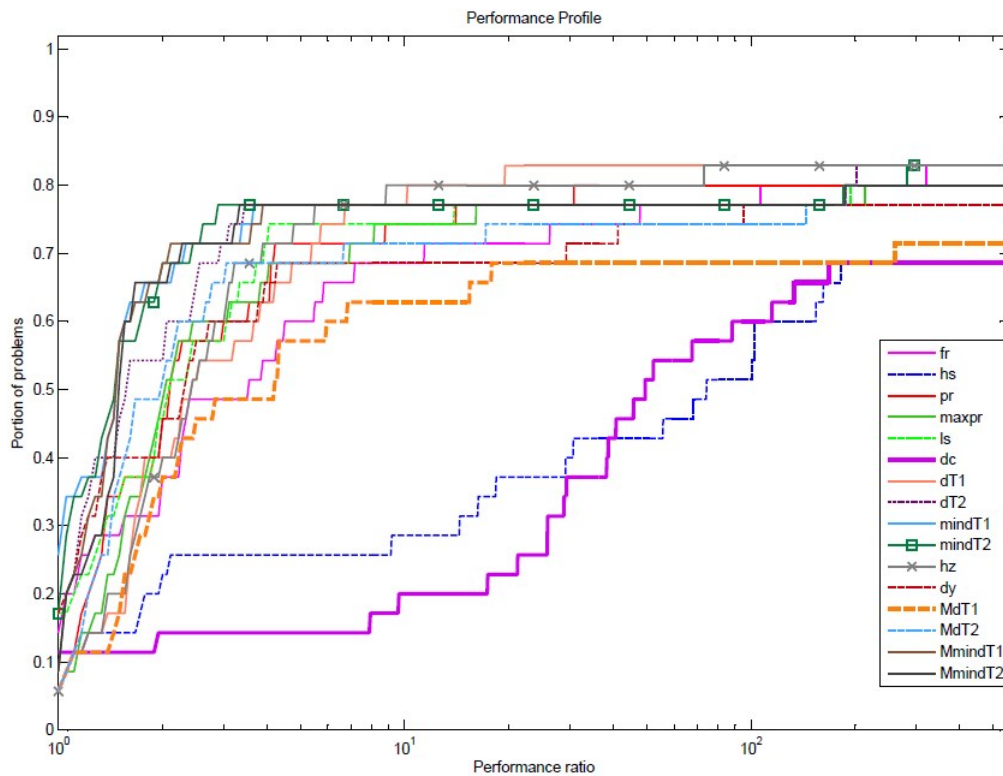


Figura 4.2: Perfil de desempenho em relação ao número de avaliações de funções para a busca de Armijo.

	Beta	Eficiência (Iterações)	Eficiência (Avaliações)	Robustez
1	fr	14,29%	8,57%	74,29%
2	hs	5,71%	5,71%	60%
3	pr	8,57%	8,57%	74,29%
4	maxpr	8,57%	2,86%	74,29%
5	ls	8,57%	8,57%	80%
6	dc	11,43%	5,71%	80%
7	dT1	8,57%	2,86%	80%
8	dT2	11,43%	5,71%	82,86%
9	mindT1	17,14%	11,43%	88,57%
10	mindT2	25,71%	22,86%	91,43%
11	hz	11,43%	2,86%	80%
12	dy	8,57%	14,29%	71,43%
13	MdT1	11,43%	5,71%	80%
14	MdT2	14,29%	11,43%	82,86%
15	MmindT1	34,29%	28,57%	85,71%
16	MmindT2	20%	14,29%	88,57%

Tabela 4.3: Tabela de resultados para a busca de Wolfe Forte.

Avaliando os resultados para a busca de Wolfe melhorada, observamos um aumento na robustez em relação às outras buscas. Esse é um bom sinal para a técnica, uma vez que até o momento não há resultados de convergência que relacionem essa busca com o algoritmo implementado, em particular para várias das escolhas de β_k da Tabela 4.1, e ainda assim os resultados práticos são interessantes. A versão mais robusta do algoritmo foi a com o beta **mindT1**, que resolveu 97,14% dos problemas, ou seja, deixou de resolver apenas 1 dos 35 problemas. Em termos de eficiência, destacamos os betas **mindT2** e **MmindT2**, que foram os mais eficientes em termos de número de iterações para 37,14% dos problemas. Antes de nos precipitarmos em já afirmar que a busca de Wolfe melhorada é superior à de Wolfe Forte, vale lembrar além do fato da melhora na robustez, que com o critério de sucesso (4.1) não se comparou a solução obtida por buscas diferentes. Nosso palpite, porém, baseando-nos na experiência que adquirimos durante a produção dos experimentos computacionais, é de que a busca melhorada é no mínimo promissora. Uma análise que compare o desempenho de uma busca contra a outra é um interessante

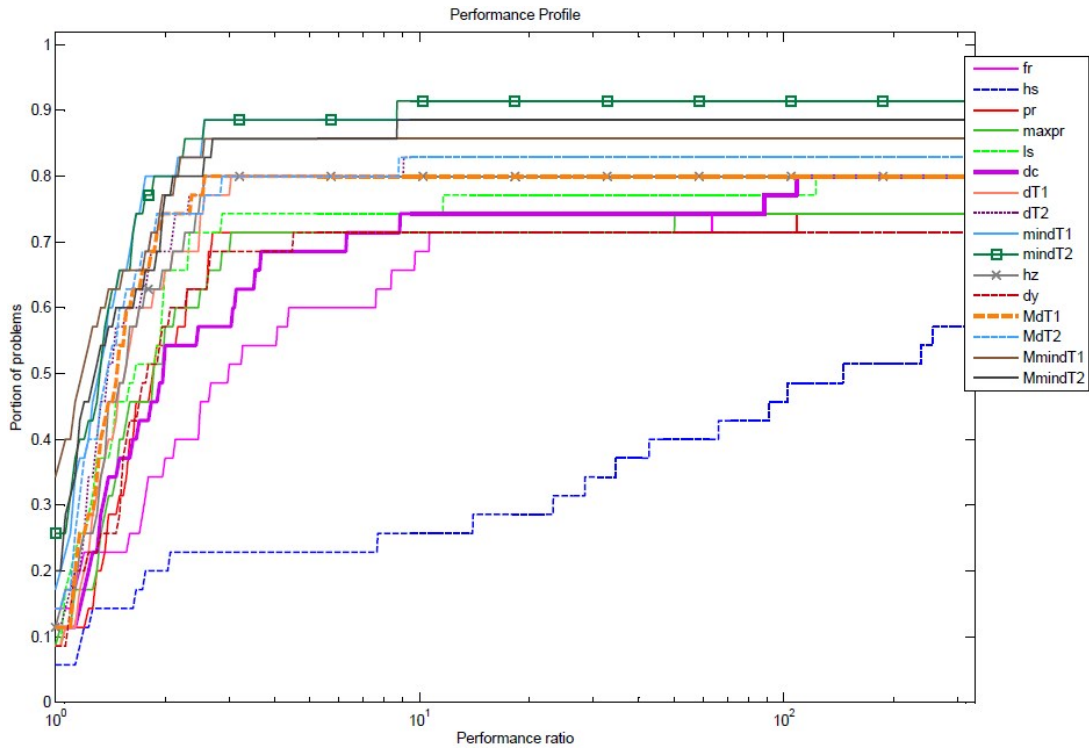


Figura 4.3: Perfil de desempenho em relação ao número de iterações para a busca de Wolfe Forte.

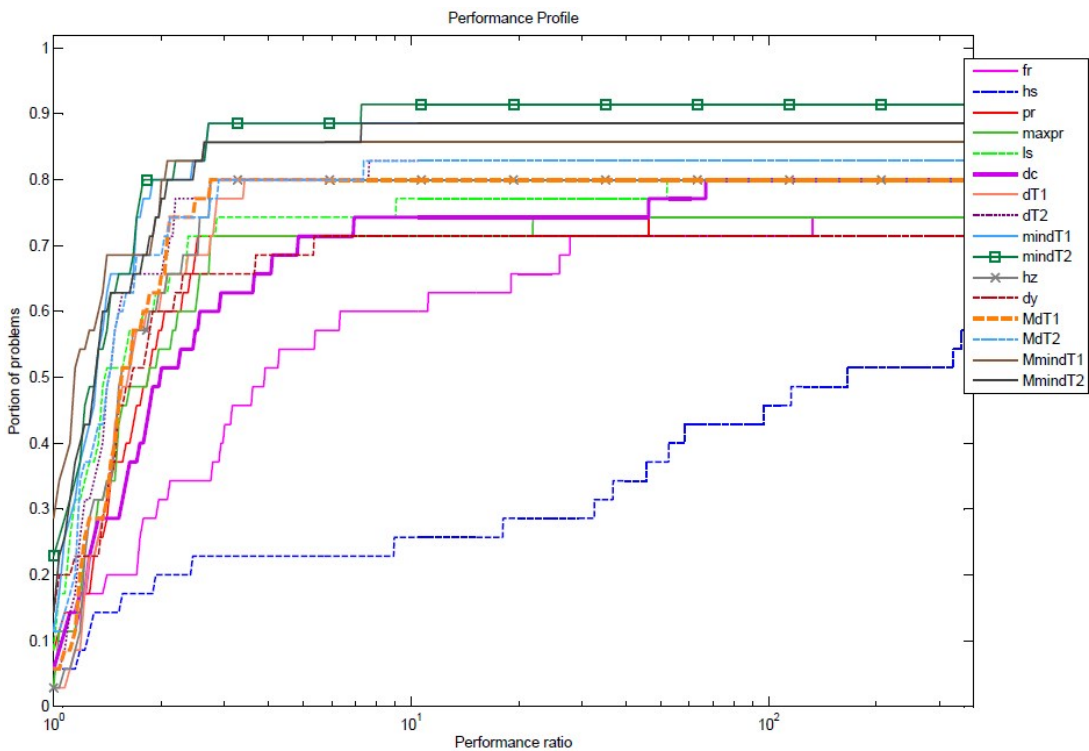


Figura 4.4: Perfil de desempenho em relação ao número de avaliações de função para a busca de Wolfe Forte.

	Beta	Eficiência (Iterações)	Eficiência (Avaliações)	Robustez
1	fr	22,86%	22,86%	85,71%
2	hs	20%	20%	80%
3	pr	22,86%	25,71%	82,86%
4	maxpr	28,57%	25,71%	80%
5	ls	28,57%	25,71%	82,86%
6	dc	22,86%	20%	88,57%
7	dT1	31,43%	28,57%	91,43%
8	dT2	34,29%	34,29%	94,29%
9	mindT1	28,57%	37,14%	97,14%
10	mindT2	37,14%	45,71%	91,43%
11	hz	31,43%	28,57%	91,43%
12	dy	34,29%	28,57%	85,71%
13	MdT1	28,57%	28,57%	85,71%
14	MdT2	28,57%	25,71%	88,57%
15	MmindT1	25,71%	34,29%	91,43%
16	MmindT2	37,14%	34,29%	88,57%

Tabela 4.4: Tabela de resultados para a busca de Wolfe melhorada.

assunto para pesquisa futura.

Por fim, para enriquecer nosso estudo, olharemos com um pouco mais de cuidado para o desempenho do Algoritmo 4.1 com busca de Wolfe Forte. Escolhemos essa busca para essa última análise por ela ser padrão em algoritmos de Gradientes Conjugados. Avaliando os resultados obtidos, observamos que os betas **mindT1**, **mindT2**, **MmindT1** e **MmindT2** se destacam em termos de robustez e eficiência. Por essa razão, decidimos analisá-los separadamente. A vantagem de fazer uma comparação com menos versões do algoritmo é poder observar melhor a eficiência de cada uma. Optamos por apresentar, na Figura 4.7, apenas o perfil de desempenho em relação ao número de iterações, uma vez que os resultados para os quatro betas em questão com respeito ao número de avaliações de funções foram semelhantes.

Apesar dos resultados serem semelhantes para os quatro betas em análise na Figura 4.7,

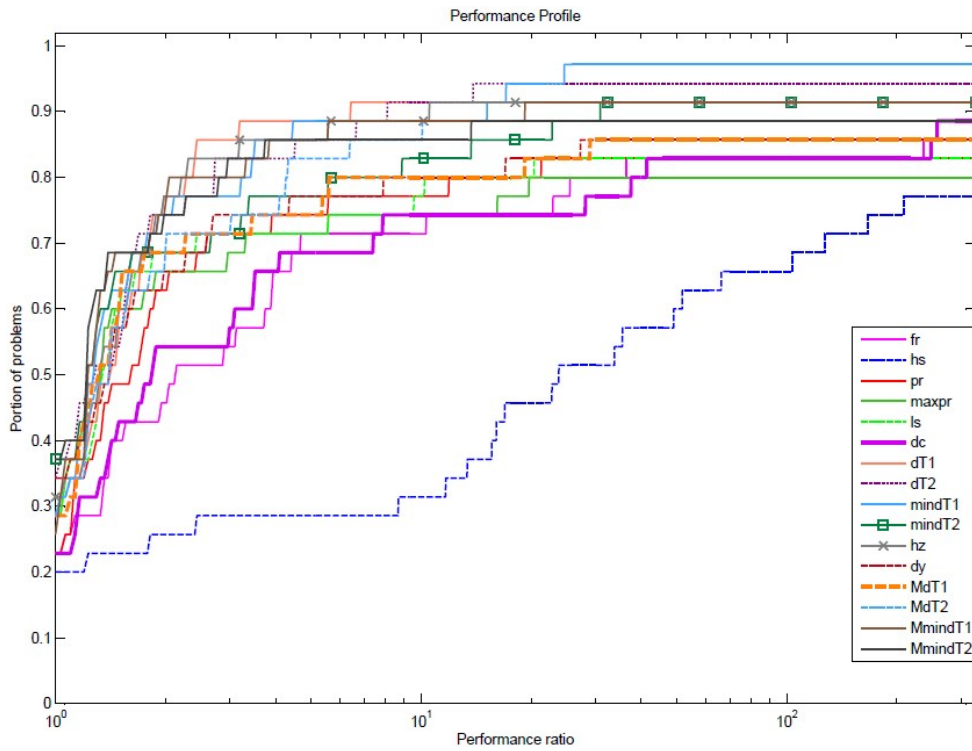


Figura 4.5: Perfil de desempenho em relação ao número de iterações para a busca de Wolfe melhorada.

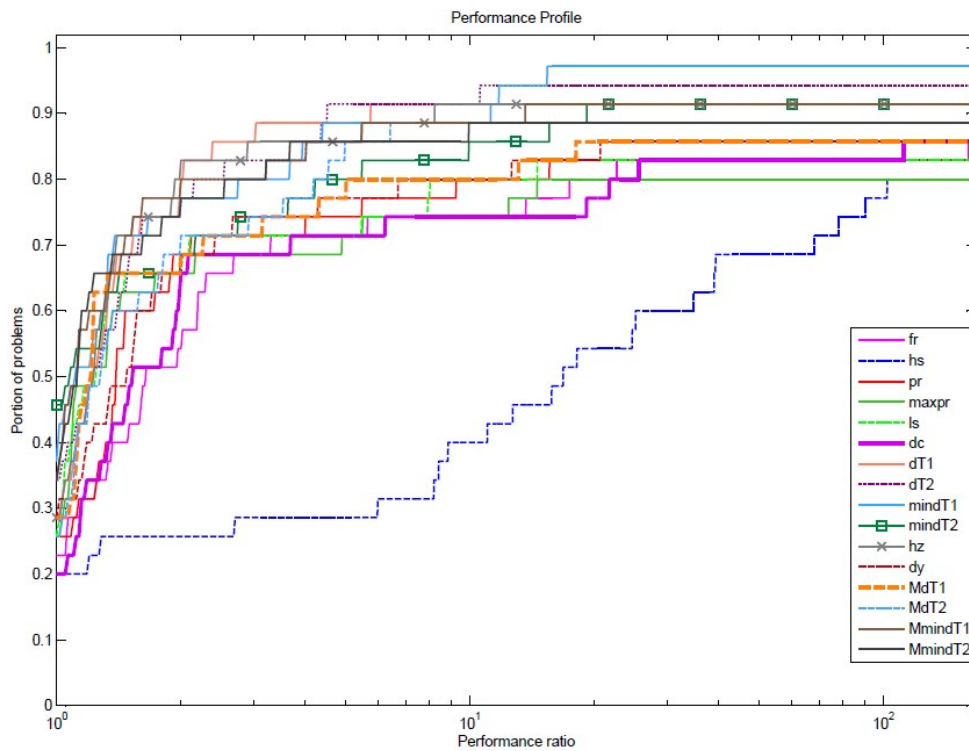


Figura 4.6: Perfil de desempenho em relação ao número de avaliações de função para a busca de Wolfe melhorada.

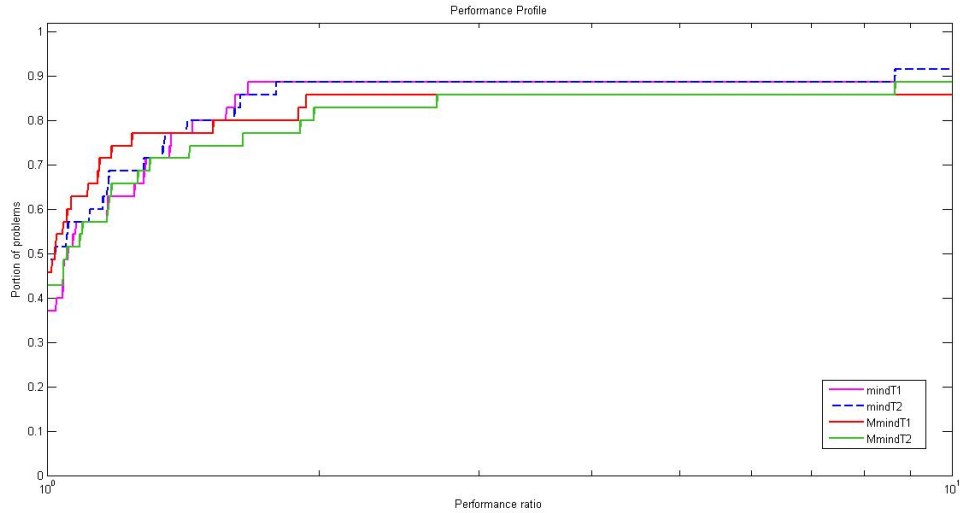


Figura 4.7: Os melhores betas em robustez e eficiência com a busca de Wolfe Forte.

destacamos o desempenho de **mindT2**, por ter sido o mais robusto, resolvendo 91,43% dos problemas, e também o mais eficiente, sendo o mais rápido em 48,57% dos casos. Podemos observar que, apesar de nossos testes não respeitarem toda a estrutura sugerida em [5], uma vez que o Algoritmo 3.2 não é igual ao Algoritmo 4.1, os resultados para os betas propostos no artigo foram superiores tanto em termos de robustez como de eficiência. Um estudo mais detalhado dos resultados apresentados em [5] pode ser bastante produtivo e motivar outros algoritmos de Gradientes Conjugados ainda mais eficientes.

Capítulo 5

Conclusões

Neste trabalho, discutimos os métodos de Gradientes Conjugados para problemas gerais de minimização irrestrita. Estudamos seus aspectos teóricos, primeiramente para problemas quadráticos e depois também para os não-quadráticos, dando especial importância ao que diz respeito às diferentes escolhas do parâmetro beta. Dentre os algoritmos clássicos para problemas não-quadráticos, demos maior atenção aos propostos por Fletcher e Reeves [19] e Polak e Ribière [30], dos quais estudamos mais a fundo os resultados de convergência. Observamos, através da teoria para esses algoritmos, que a busca de Wolfe é muito importante no contexto de Gradientes Conjugados. Em seguida, estudamos o artigo de Dai e Kou [5], no qual foram introduzidas novas estratégias, das quais destacamos uma nova classe de parâmetros β_k que se inspira na atualização BFGS e uma proposta de melhoria para busca de Wolfe. Ao longo de nosso trabalho, vimos 16 versões do parâmetro β_k , que testamos em experimentos numéricos em três versões diferentes do algoritmo de Gradientes Conjugados: uma com busca de Wolfe Forte, por esta embasar muitos dos resultados teóricos nos métodos clássicos, outra com busca de Armijo, por entender que é a busca mais comum em Otimização, e outra com a busca proposta em [5], pela curiosidade em observar seu desempenho prático.

Os resultados numéricos indicam uma superioridade dos betas propostos em [5], bem como da busca de Wolfe melhorada em relação às demais testadas. Na melhor combinação de β_k e busca linear em termos de robustez, apenas um dos 35 problemas testados deixou de ser resolvido, o que ocorreu para o beta **mindT1** e busca de Wolfe melhorada. Mesmo nos demais casos, o algoritmo se mostrou satisfatório em quase todas suas versões, raramente resolvendo menos de 80% dos problemas mesmo para os piores betas. Os bons resultados

para as ideias de Dai e Kou nos motivam a propor, como trabalho futuro, um estudo mais aprofundado dos aspectos teóricos do algoritmo que os autores apresentam, bem como sua implementação completa, que contempla por exemplo interessantes estratégias de reinício da direção de busca.

Referências Bibliográficas

- [1] M. Al-Baali, *Numerical experience with a class of self-scaling quasi-Newton algorithms*, J. Optim. Theory Appl., 96 (1998), pp. 533–553.
- [2] Y. H. Dai, *Analyses of conjugate gradient methods*, Ph.D. thesis, Institute of Computational Mathematics and Scientific/Engineering Computing, Chinese Academy of Sciences, 1997.
- [3] Y. H. Dai, *New properties of a nonlinear conjugate gradient method*, Numer. Math., 89 (2001), pp. 83-98.
- [4] Y. Dai, J. Han, G. Liu, D. Sun, H. Yin, e Y. X. Yuan, *Convergence properties of nonlinear conjugate gradient methods*, SIAM J. Optim., 10 (1999), pp. 345–358.
- [5] Y. H. Dai e C. X. Kou, *A nonlinear conjugate gradient algorithm with an optimal property and an improved Wolfe line search*, SIAM J. Optim., 23 (2013), pp. 296–320.
- [6] Y. H. Dai e Y. Yuan, *Further studies on the Polak-Ribière-Polyak method*, Research report ICM-95-040, Institute of Computational Mathematics and Scientific/Engineering Computing, Chinese Academy of Sciences, 1995.
- [7] Y. H. Dai e Y. Yuan, *Convergence of the Fletcher-Reeves method under a generalized Wolfe wearch*, J. Comput. Math., 2 (1996), pp. 142-148.
- [8] Y. H. Dai e Y. Yuan, *Convergence properties of the conjugate descent method*, Adv. Math. (China), 26 (1996), pp. 552-562.
- [9] Y. H. Dai e Y. Yuan, *A nonlinear conjugate gradient method with a strong global convergence property*, SIAM J. Optim., 10 (1999), pp. 177-182.
- [10] Y. H. Dai e Y. Yuan, *Nonlinear Conjugate Gradient Methods*, Shanghai Science and Technology Publisher, Shanghai, 2000.

- [11] J. W. Daniel, *The conjugate gradient method for linear and nonlinear operator equations*, SIAM J. Numer. Anal., 4 (1967), pp. 10-26.
- [12] R. Fletcher e C. Reeves, *Function minimization by conjugate gradients*, Comput. J., 7 (1964), pp. 149-154.
- [13] R. Fletcher, *Practical Methods of Optimization vol.1: Unconstrained Optimization*, John Wiley and Sons, New York, 1987.
- [14] J. C. Gilbert e J. Nocedal, *Global convergence properties of conjugate gradient methods for optimization*, SIAM J. Optim., 2 (1992), pp. 21-42.
- [15] L. Grippo e S. Lucidi, *Convergence conditions, line search algorithms and trust region implementations for the Polak-Ribière conjugate gradient method*, Optimization Methods Software, 20 (2005), pp. 71-98.
- [16] W. W. Hager e H. Zhang, *A new conjugate gradient method with guaranteed descent and an efficient line search*, SIAM J. Optim., 16 (2005), pp. 170-192.
- [17] W. W. Hager e H. Zhang, *Algorithm 851: CG DESCENT, A conjugate gradient method with guaranteed descent*, ACM Transactions on Mathematical Software, 32 (2006), pp. 113-137.
- [18] W. W. Hager e H. Zhang, *A Survey of Nonlinear Conjugate Gradient Methods*. Pacific Journal of Optimization, Volume 2 (2006), pp. 35-58.
- [19] M. R. Hestenes e E. L. Stiefel, *Methods of conjugate gradients for solving linear systems*. J. Research Nat. Bur. Standards, 49 (1952), pp. 409-436.
- [20] Y. Liu e C. Storey, *Efficient generalized conjugate gradient algorithms, Part 1: Theory*, J. Optim. Theory Appl., 69 (1991), pp. 129-137.
- [21] D. G. Luenberger, *Linear and Nonlinear Programming*. Addison-Wesley, 1989.
- [22] J. J. Moré, B. S. Garbow e K. E. Hillstom. *Testing unconstrained optimization software*. ACM Transactions on Mathematical Software. Vol. 7, pp. 17-41, 1981.
- [23] M. Nocedal, e S. J. Wright, *Numerical Optimization*. Springer, 2006.

- [24] S. S. Oren e D. G. Luenberger, *Self scaling variable metric (SSVM) algorithms*, Part I: Criteria and sufficient conditions for scaling a class of algorithms, *Management Sci.*, 20 (1974), pp. 845–862.
- [25] S. S. Oren e E. Spedicato, *Optimal conditioning of self scaling variable metric algorithms*, *Math. Program.*, 10 (1976), pp. 70–90.
- [26] S. S. Oren, *Self scaling variable metric (SSVM) algorithms*, Part II: Implementation and experiments, *Management Sci.*, 20 (1974), pp. 863–874.
- [27] D. M. Panonceli, *Um Estudo de Buscas Unidirecionais Aplicadas ao Método BFGS*, Dissertação para Mestre, UFPR, (2015).
- [28] J. M. Perry, *A Class of Conjugate Gradient Algorithms with a Two-step Variable-Metric Memory*, Discussion Paper 269, Center for Mathematical Studies in Economics and Management Sciences, Northwestern University, Evanston, IL, 1977.
- [29] E. Polak e G. Ribière, *Note sur la convergence de directions conjuguées*, *Rev. Francaise Informat Recherche Opertionelle*, 3e Année 16 (1969), pp. 35-43.
- [30] B. T. Polyak, *The conjugate gradient method in extreme problems*, *USSR Comp. Math. Math. Phys.*, 9 (1969), pp. 94-112.
- [31] M. J. D. Powell, *Restart procedures of the conjugate gradient method*, *Math. Prog.*, 2 (1977), pp. 241-254.
- [32] M. J. D. Powell, *Nonconvex minimization calculations and the conjugate gradient method*, *Numerical Analysis* (Dundee, 1983), *Lecture Notes in Mathematics*, Vol. 1066, Springer- Verlag, Berlin, 1984, pp. 122-141.
- [33] A. A. Ribeiro e E. W. Karas, *Otimização contínua: Aspectos teóricos e computacionais*, Cengage Learning. São Paulo, (2013).
- [34] D. F. Shanno, *On the convergence of a new conjugate gradient algorithm*, *SIAM J. Numer. Anal.*, 15 (1978), pp. 1247–1257.
- [35] J. Sun e J. Zhang, *Global convergence of conjugate gradient methods without line search*, *Ann. Oper. Res.*, 163 (2001), pp. 161-173.

- [36] Y. Yuan, *Analysis on the conjugate gradient method*, Optim. Methods Softw., 2 (1993), pp. 19-29.
- [37] G. Zoutendijk, *Nonlinear Programming, Computational Methods, in Integer and Nonlinear Programming*, J. Abadie, ed., North-Holland, Amsterdam, 1970, pp. 37-86.