



Universidade Estadual de Campinas
Instituto de Matemática Estatística e Computação Científica
Departamento de Matemática Aplicada



Implementação Eficiente dos Métodos de Pontos Interiores Especializados para o Problema de Regressão pela Norma L_p

Eliana Contharteze Grigoletto

Mestrado em Matemática Aplicada

Orientador: Prof. Dr. Aurelio Ribeiro Leite de Oliveira

Co-orientadora: Dra. Daniela Renata Cantane

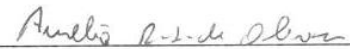
Este trabalho contou com suporte financeiro da FAPESP.

Campinas-SP
Junho de 2011

Implementação Eficiente dos Métodos de Pontos Interiores Especializados para o Problema de Regressão pela Norma L_p

Este exemplar corresponde à redação final da tese devidamente corrigida e defendida por **Eliana Conharteze Grigoletto** e aprovada pela comissão julgadora.

Campinas, 13 de junho de 2011.



Prof. Dr. Aurelio R. L. de Oliveira
Orientador



Dr.ª. Daniela Renata Cantane
Co-Orientadora

Banca examinadora:

Prof. Dr. Aurelio Ribeiro Leite de Oliveira (IMECC - UNICAMP)

Profa. Dra. Maria Aparecida Diniz Ehrhardt (IMECC - UNICAMP)

Prof. Dr. Antonio Roberto Balbo (Depto. de Matemática - UNESP)

Dissertação apresentada ao Instituto de Matemática Estatística e Computação Científica, UNICAMP, como requisito parcial para a obtenção do título de **Mestre em Matemática Aplicada**.

FICHA CATALOGRÁFICA ELABORADA PELA BIBLIOTECA DO IMECC DA UNICAMP

Bibliotecária: Maria Fabiana Bezerra Müller – CRB8 / 6162

Grigoletto
Contharteze, Eliana *Grigoletto*

G877i Implementação eficiente dos métodos de pontos interiores especializados para o problema de regressão pela norma L_p / Eliana Contharteze Grigoletto – Campinas, [S.P.: s.n.], 2011.
135f. : il.

Orientador: Aurelio Ribeiro Leite de Oliveira.
Co-orientadora: Daniela Renata Cantane.
Dissertação (Mestrado) – Universidade Estadual de Campinas, Instituto de Matemática, Estatística e Computação Científica.

1.Métodos de pontos interiores. 2.Programação não-linear. 3.Modelos de regressão. I. Oliveira, Aurelio Ribciro Leite de II. Universidade Estadual de Campinas. Instituto de Matemática, Estatística e Computação Científica. III. Título.

Título em Inglês: Efficient implementation of interior point methods specialized to the L_p -norm fitting problem

Palavras-chave em inglês (Key-words): 1.Interior point methods. 2.Nonlinear programming. 3.Fitting models.

Área de Concentração: Matemática Aplicada.

Titulação: Mestre em Matemática Aplicada.

Banca Examinadora: Prof. Dr. Aurclio Ribeiro Leite de Oliveira (IMECC - UNICAMP)

Profa. Dra. Maria Aparecida Diniz Ehrhardt (IMECC - UNICAMP)

Prof. Dr. Antonio Roberto Balbo (Depto. de Matemática - UNESP)

Data da defesa: 02/05/2011

Programa de Pós-graduação: Mestrado em Matemática Aplicada

Dissertação de Mestrado defendida em 02 de maio de 2011 e aprovada

Pela Banca Examinadora composta pelos Profs. Drs.

Aurelio R L de Oliveira

Prof.(a). Dr(a). AURELIO RIBEIRO LEITE DE OLIVEIRA

Maria Aparecida Diniz Ehrhardt

Prof.(a). Dr(a). MARIA APARECIDA DINIZ EHRHARDT

Antonio Roberto Balbo

Prof.(a). Dr(a). ANTONIO ROBERTO BALBO

Agradecimentos

À Deus, por me dar forças e guiar em mais essa etapa da minha vida.

À minha mãe Inês e meu irmão Eliel, cujo apoio e incentivo aos estudos foram imprescindíveis para que eu pudesse concluir mais essa etapa da minha vida acadêmica, e por estarem sempre comigo nessa caminhada.

Aos meus amigos e colegas Liz, Luis, Luciana, Marta, Juliano, Rafael, Sofia, Fábio, Cícero, entre outros, pelo ânimo para seguir em frente e pelo apoio nas dúvidas técnicas.

Ao meu orientador Aurelio, por estar sempre presente para me ajudar no projeto e diante dos momentos de cansaço e desânimo soube ter paciência e me dar o apoio necessário para eu vencer mais esse desafio.

Ao meu orientador da graduação Balbo, pela força, por me incentivar nos estudos e por ter me direcionado ao mestrado.

À FAPESP - Fundação de Amparo e Apoio à Pesquisa do Estado de São Paulo, pelo apoio financeiro.

Resumo

Os métodos de pontos interiores primais-duais para o problema de regressão pela norma- p desenvolvidos terão sua estrutura matricial resultante explorada objetivando uma implementação eficiente. O problema de regressão tem inúmeras aplicações em diversas áreas. A norma-2 é muito popular, entre outros motivos, por permitir uma solução direta. Por sua vez a norma-1 permite reduzir o efeito de pontos discrepantes enquanto que a norma infinito garante proteção contra o pior caso. A norma- p permite pensar estas características de diferentes formas, adaptando o método ao problema a ser resolvido. A implementação dos métodos de pontos interiores a ser desenvolvida será comparada com implementações eficientes de outros métodos já existentes.

Abstract

The specific primal-dual interior point methods for the p -norm fitting problem previously developed will have their resulting matrix structure exploited aiming at an efficient implementation. The fitting problem has numerous applications in various areas. The 2-norm is very popular, among other reasons, for allowing a direct solution. The 1-norm allows the reduction of the effect of outliers while the infinite norm provides protection against the worst case. The p -norm allows to think these characteristics in different ways adapting the method to the problem to be solved. The interior point method implementation to be developed will be compared with an efficient implementation of other existing methods.

Sumário

1	Introdução	1
2	Conceitos de Programação Não-Linear	9
2.1	Análise Convexa	9
2.1.1	Conjuntos Convexos	9
2.1.2	Função Convexa	10
2.1.3	Função Côncava	10
2.2	Otimização Não-Linear	10
2.3	Método de Newton	12
2.4	Método Barreira Logarítmica	14
2.5	Método Barreira Logarítmica Preditor-Corretor	19
3	Métodos Existentes para o Problema de Regressão L_p	25
3.1	O problema de Regressão pela Norma L_p	25
3.2	Métodos Similares	27
3.3	Método IRLS	28
3.4	Método IRLSL	29
3.5	Método GNCS	32
3.6	Métodos de Pontos Interiores Aplicados ao Problema de Regressão pela Norma L_p	34
3.6.1	Método Barreira Logarítmica	34
3.6.2	Método Barreira Logarítmica Preditor-Corretor	39
3.6.3	Método Primal-Dual Barreira Logarítmica	46
3.6.4	Método Primal-Dual Barreira Logarítmica Preditor-Corretor	51
4	Modificação nos MPI Aplicados ao Problema de Regressão pela Norma L_p	61
4.1	Método Barreira Logarítmica	61
4.2	Método Barreira Logarítmica Preditor-Corretor	66

4.3	Método Primal-Dual Barreira Logarítmica	69
4.4	Método Primal-Dual Barreira Logarítmica Preditor- Corretor	74
5	Estudo da Estrutura Matricial	79
5.1	Cálculo de Ax	81
5.2	Cálculo de A^tA	82
5.3	Inversa de uma Matriz de Hankel	85
6	Resultados Computacionais	89
6.1	Problema de Grande Porte	90
6.1.1	Modificação no Ponto Inicial	90
6.1.2	Modificação no Cálculo do Resíduo	93
6.1.3	Modificação no Cálculo das Direções	94
6.1.4	Resultados Finais	97
6.2	Função Cosseno	102
6.3	Função Logaritmo	107
6.4	Função Seno Hiperbólico	111
6.5	Implementação Eficiente em C	115
6.5.1	Problema de Grande Porte	115
6.5.2	Função Seno	121
7	Conclusões	125
	Bibliografia	128
	Referências Bibliográficas	129

Capítulo 1

Introdução

O objetivo desse trabalho é aperfeiçoar e implementar métodos de pontos interiores para resolver de maneira eficiente problemas de otimização não-linear da forma:

$$\min_{x \in \mathfrak{R}^n} \|Ax - b\|_p^p, \quad (1.1)$$

onde $A \in \mathfrak{R}^{m \times n}$ é uma matriz de posto completo, $b \in \mathfrak{R}^m$, $m > n$ e $1 < p < \infty$. O sistema linear $Ax = b$ pode ser inconsistente e a matriz A é uma matriz de Vandermonde no caso particular do problema de regressão polinomial.

Podemos definir a norma-p de um vetor $v \in \mathfrak{R}^n$ da seguinte forma:

$$\bullet \|v\|_p = \left(\sum_{i=1}^n |v_i|^p \right)^{\frac{1}{p}} \Rightarrow \|v\|_p^p = \sum_{i=1}^n |v_i|^p.$$

Em particular:

$$\bullet \text{ Norma-1: } \|v\|_1 = \sum_{i=1}^n |v_i|.$$

$$\bullet \text{ Norma-2: } \|v\|_2 = \left(\sum_{i=1}^n (v_i)^2 \right)^{\frac{1}{2}}.$$

- Norma- ∞ : $\|v\|_\infty = \max_{i=1,2,\dots,n} \{|v_i|\}$.

Vejamos o que é um problema de regressão polinomial com um exemplo didático.

Exemplo

Vamos considerar os seguintes pontos no \mathfrak{R}^2 : $(-4,1)$, $(-3,-2)$, $(-2,2)$, $(-1,4)$, $(1,1)$, $(2,3)$, $(3,-1)$, $(4,2)$. No problema de regressão polinomial a matriz A é uma matriz de Vandermonde. Vamos encontrar os coeficientes de um polinômio, dados pela resolução do problema de $\min \|Ax - b\|_p^p$.

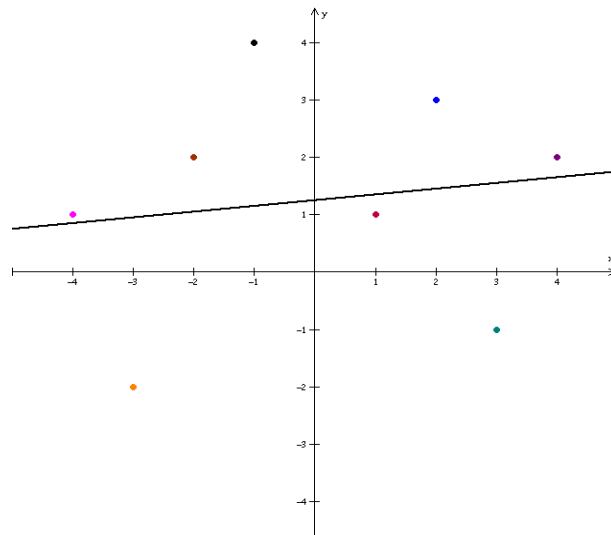
Polinômio de grau 1: Inicialmente o polinômio buscado é da forma $p(x) = a_0 + a_1x$. O ideal é que esse polinômio passe por todos os pontos para minimizar o problema. Sendo assim, devemos ter: $a_0 + a_1(-4) = 1$ para o primeiro ponto, \dots e $a_0 + a_14 = 2$ para o último ponto. Temos então que resolver o problema $\min \|Ax - b\|_p^p$, onde:

$$A = \begin{bmatrix} 1 & -4 \\ 1 & -3 \\ 1 & -2 \\ 1 & -1 \\ 1 & 1 \\ 1 & 2 \\ 1 & 3 \\ 1 & 4 \end{bmatrix}, \quad b = \begin{bmatrix} 1 \\ -2 \\ 2 \\ 4 \\ 1 \\ 3 \\ -1 \\ 2 \end{bmatrix} \quad \text{e} \quad x = \begin{bmatrix} a_0 \\ a_1 \end{bmatrix}.$$

A solução desse problema utilizando o Método Barreira Logarítmica (será descrito no Capítulo 4) para $p = 1,5$ é dada por:

$$x^* = \begin{bmatrix} 1,25 \\ 0,10 \end{bmatrix} \quad \text{e} \quad \|Ax^* - b\|_{1,5}^{1,5} = 17,2781.$$

Graficamente:



$$p(x) = 1,25 + 0,1x.$$

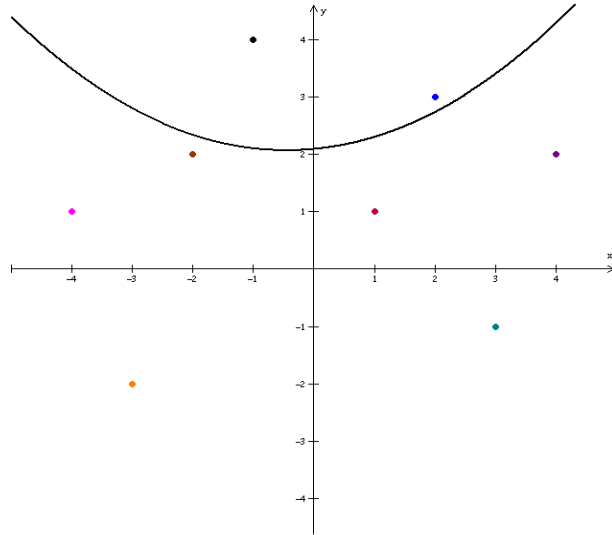
Polinômio de grau 2: O polinômio agora é da forma $p(x) = a_0 + a_1x + a_2x^2$. Para esse problema teremos:

$$A = \begin{bmatrix} 1 & -4 & 16 \\ 1 & -3 & 9 \\ 1 & -2 & 4 \\ 1 & -1 & 1 \\ 1 & 1 & 1 \\ 1 & 2 & 4 \\ 1 & 3 & 9 \\ 1 & 4 & 16 \end{bmatrix}, \quad b = \begin{bmatrix} 1 \\ -2 \\ 2 \\ 4 \\ 1 \\ 3 \\ -1 \\ 2 \end{bmatrix} \quad \text{e} \quad x = \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix}.$$

A solução desse problema é:

$$x^* = \begin{bmatrix} 2,0930 \\ 0,1000 \\ -0,1124 \end{bmatrix} \quad \text{e} \quad \|Ax^* - b\|_{1,5}^{1,5} = 16,8352.$$

Graficamente:



$$p(x) = 2,093 + 0,1x - 0,1124x^2.$$

Polinômio de grau 6: O polinômio é da forma $p(x) = a_0 + a_1x + a_2x^2 + a_3x^3 + a_4x^4 + a_5x^5 + a_6x^6$. Para esse problema teremos:

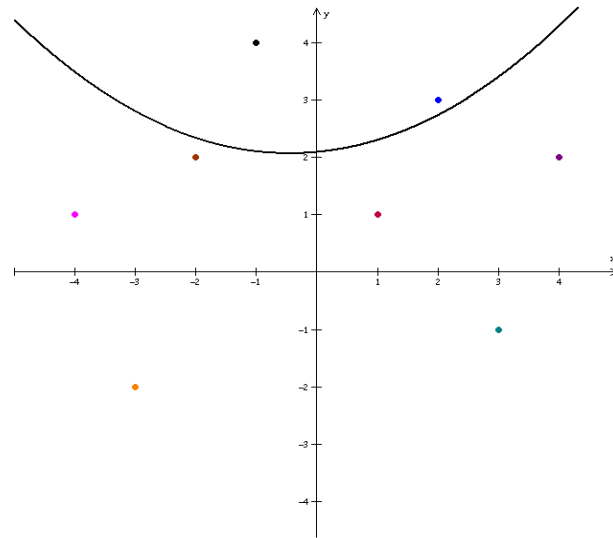
$$A = \begin{bmatrix} 1 & -4 & 16 & -64 & 256 & -1024 & 4096 \\ 1 & -3 & 9 & -27 & 81 & -243 & 729 \\ 1 & -2 & 4 & -8 & 16 & -32 & 64 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 4 & 8 & 16 & 32 & 64 \\ 1 & 3 & 9 & 27 & 81 & 243 & 729 \\ 1 & 4 & 16 & 64 & 256 & 1024 & 4096 \end{bmatrix}, \quad b = \begin{bmatrix} 1 \\ -2 \\ 2 \\ 4 \\ 1 \\ 3 \\ -1 \\ 2 \end{bmatrix} \quad \text{e} \quad x = \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \end{bmatrix}.$$

A solução desse problema é:

$$x^* = \begin{bmatrix} 1,6143 & -0,8735 & 1,1611 & 0,2153 & -0,2889 & -0,0096 & 0,0135 \end{bmatrix}^t \text{ e}$$

$$\|Ax^* - b\|_{1,5}^{1,5} = 3,4915.$$

Graficamente:



$$p(x) = 1,6143 - 0,8735x + 1,1611x^2 + 0,2153x^3 - 0,2889x^4 - 0,0096x^5 + 0,0135x^6.$$

Generalizando, se tivermos m pontos no \mathfrak{R}^2 dados por $(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)$ e quisermos fazer regressão por polinômio de grau n , $p(x) = a_0 + a_1x + a_2x^2 + \dots + a_nx^n$, então teremos que resolver o problema de $\min \|Ax - b\|_p^p$, onde:

$$A = \begin{bmatrix} 1 & x_1 & x_1^2 & \cdots & x_1^n \\ 1 & x_2 & x_2^2 & \cdots & x_2^n \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_m & x_m^2 & \cdots & x_m^n \end{bmatrix}, \quad b = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix} \quad \text{e} \quad x = \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{bmatrix}.$$

No Capítulo 2 apresentaremos conceitos fundamentais e alguns métodos para resolver

um problema de otimização não-linear, métodos esses que serão utilizados na resolução do problema (1.1).

No Capítulo 3 citamos vários métodos existentes que resolvem o problema (1.1), para os casos em que $1 < p < \infty$, $p = 1$ e $p = \infty$. Em [16] são descritos os métodos Iteratively Reweighted Least-Squares (IRLS), Método IRLS com procedimento de busca linear (IRLSL) e o Método de Newton Globalizado que usa as condições de folgas complementares (GNCS). O método IRLS funciona melhor quando p está suficientemente longe de 1. Dentre esses métodos, o que apresenta melhor desempenho é o GNCS. Entretanto, esse método utiliza um procedimento de busca linear que é computacionalmente muito caro.

Em [7] e [12] são descritos os métodos de pontos interiores barreira logarítmica, barreira logarítmica preditor-corretor, primal-dual barreira logarítmica e primal-dual barreira logarítmica preditor-corretor para resolver problemas de regressão pela norma L_p , com $1 < p < 2$. Esses métodos foram implementados em MATLAB e os resultados computacionais comprovam que são mais eficientes que o GNCS.

No Capítulo 3, Seção 3.6.2 e Seção 3.6.4, apresentaremos os métodos barreira logarítmica preditor-corretor e primal-dual barreira logarítmica preditor-corretor já existentes, mas com as modificações no cálculo dos resíduos. No Capítulo 4 será apresentada uma estratégia para aperfeiçoar os métodos propostos em [7], por meio de modificações no cálculo das direções, e dos pontos iniciais, com a intenção de reduzir o número de operações e fazer com que estes métodos sejam computacionalmente mais eficientes, especialmente quando utilizados para resolver problemas de grande porte. Em seguida será feita uma implementação em MATLAB e em C. Os novos métodos implementados serão comparados com os métodos

descritos em [7].

O enfoque deste trabalho é a resolução de problemas de regressão polinomial, com isso ganhamos algumas propriedades importantes que farão com que a implementação em C seja mais eficiente. No Capítulo 5 faremos uma análise acerca das matrizes envolvidas nos métodos do Capítulo 4, essas observações nos levam a uma implementação eficiente em C.

No Capítulo 6 podemos verificar os resultados computacionais da implementação em MATLAB e em C dos métodos descritos no Capítulo 4 e também podemos compará-los com os métodos já existentes.

As conclusões deste trabalho serão discutidas no Capítulo 7.

Capítulo 2

Conceitos de Programação Não-Linear

Apresentaremos neste capítulo alguns conceitos fundamentais para a resolução de um problema de programação não-linear utilizando métodos de pontos interiores, esses métodos serão usados para resolver o problema de regressão pela norma L_p .

2.1 Análise Convexa

2.1.1 Conjuntos Convexos

Definição: Um conjunto S em \mathfrak{R}^n é dito convexo se o segmento de reta que une dois pontos quaisquer de S também pertence ao conjunto S . Em outras palavras, se x_1 e x_2 pertencem a S , então $\lambda x_1 + (1 - \lambda)x_2$ também deve pertencer a S para todo $\lambda \in [0, 1]$.

O elemento $x = \lambda x_1 + (1 - \lambda)x_2$, com $\lambda \in [0, 1]$, é uma combinação convexa de x_1 e x_2 . Por indução, $y = \sum_{j=1}^k \lambda_j x_j$, onde $\sum_{j=1}^k \lambda_j = 1$, $\lambda_j \geq 0$, $j = 1, \dots, k$ também é chamado de combinação convexa de x_1, \dots, x_k . Se a condição de não-negatividade dos multiplicadores λ_j , $j = 1, \dots, k$ for retirada, então a combinação é dita combinação afim.

Para maiores detalhes sobre conjuntos convexos, ver [23] e [10].

2.1.2 Função Convexa

Definição: Seja $f : S \rightarrow E$, onde S é um conjunto convexo não vazio em \mathfrak{R}^n . A função f é dita convexa em S se

$$f(\lambda x_1 + (1 - \lambda)x_2) \leq \lambda f(x_1) + (1 - \lambda)f(x_2)$$

para cada $x_1, x_2 \in S$ e para cada $\lambda \in [0, 1]$.

A função f é dita estritamente convexa em S se

$$f(\lambda x_1 + (1 - \lambda)x_2) < \lambda f(x_1) + (1 - \lambda)f(x_2)$$

para cada $x_1, x_2 \in S$, com $x_1 \neq x_2$ e $\lambda \in (0, 1)$.

2.1.3 Função Côncava

Definição: A função $f : S \rightarrow E$ é dita côncava (estritamente côncava) em S se $-f$ é convexa (estritamente convexa) em S .

Para mais detalhes sobre convexidade ver [5].

2.2 Otimização Não-Linear

Seja X um conjunto não vazio aberto em \mathfrak{R}^n e sejam $f : \mathfrak{R}^n \rightarrow \mathfrak{R}$ e $h : \mathfrak{R}^n \rightarrow \mathfrak{R}^m$. Considere o problema (P):

$$\begin{aligned} \min f(x) \\ \text{s.a. } h(x) = 0 \end{aligned}$$

Seja S o conjunto de soluções factíveis de (P), isto é, $S := \{x \in \mathfrak{R}^n, h(x) = 0\}$.

A função Lagrangiana $L : X \times \mathfrak{R}^m \rightarrow \mathfrak{R}$ associada ao problema de minimizar f sujeita a $x \in S$ é definida por

$$L(x, \lambda) = f(x) + \lambda^t h(x),$$

onde $\lambda \in \mathfrak{R}^m$ é o vetor multiplicador de Lagrange.

Ponto Regular (Qualificação de restrições): Um ponto $x_1 \in S$ é um ponto regular com respeito às restrições $h(x) = 0$ se $\{\nabla h_1(x_1), \nabla h_2(x_1), \dots, \nabla h_m(x_1)\}$ é um conjunto linearmente independente.

Teorema: Num ponto regular $x_1 \in S$, o subespaço tangente é representado por

$$M(x_1) = \{\lambda \in \mathfrak{R}^m : J(h(x_1))^t \lambda = 0\},$$

onde $J(h(x_1)) = \begin{bmatrix} \nabla h_1(x_1)^t \\ \nabla h_2(x_1)^t \\ \vdots \\ \nabla h_m(x_1)^t \end{bmatrix}$ representa a Jacobiana de h em x_1 .

Condições Necessárias de Primeira Ordem: Suponhamos $f, h \in C^1$. Seja x^* um minimizador local de f sujeito a $x \in S$ e um ponto regular de S . Então existe um vetor $\lambda \in \mathfrak{R}^m$ tal que:

$$\nabla f(x^*) + \sum_{i=1}^m \lambda_i \nabla h_i(x^*) = \nabla f(x^*) + J(h(x^*))^t \lambda = 0.$$

Condições Necessárias de Segunda Ordem: Suponhamos $f, h \in C^2$. Seja x^* um mínimo local de f em X e um ponto regular de X . Então existe $\lambda \in \mathfrak{R}^m$ tal que:

$$\nabla f(x^*) + \sum_{i=1}^m \lambda_i \nabla h_i(x^*) = 0,$$

e a Hessiana da função Lagrangiana $\nabla^2 L(x^*) = \nabla^2 f(x^*) + \sum_{i=1}^m \lambda_i \nabla^2 h_i(x^*)$ é semi-definida positiva em $M(x^*)$, isto é, $y^t \nabla^2 L(x^*) y \geq 0, \forall y \in M(x^*)$.

Função Coerciva: Diz-se que f é coerciva em X se $\lim_{k \rightarrow \infty} f(x^k) = \infty$ para toda sequência $\{x^k\}$ em X tal que $\|x^k\| \rightarrow \infty$.

- Exemplo: Considere $f : \mathfrak{R} \rightarrow \mathfrak{R}$ definida por $f(x) = x^4 - x^2$. Essa função é coerciva em \mathfrak{R} , pois $\lim_{|x| \rightarrow \infty} x^4 - x^2 = +\infty$.

Teorema de Weierstrass: Seja $X \subset \mathfrak{R}^n$ não-vazio e $f : D_f \supset X \rightarrow \mathfrak{R}$ contínua em X . Assuma que uma das seguintes afirmações é verdadeira:

- X é compacto.
- X é fechado e f é coerciva em X .
- $\exists \theta \in \mathfrak{R}$ tal que $L_\theta := \{x \in X : f(x) \leq \theta\}$ é compacto e não-vazio.

Então $\exists x^* \in X$ tal que $f(x^*) = \inf \{f(x) : x \in X\}$. Em outras palavras, este teorema fornece condições suficientes para a existência de mínimos globais de f em $X \subset \mathfrak{R}^n$.

Para mais detalhes sobre otimização não-linear ver [17] e [6].

Proposição - Função Convexa: Seja $f : X \rightarrow \mathfrak{R}$ uma função convexa sobre o conjunto convexo X .

(i) Um mínimo local de f sobre X é também um mínimo global sobre X . Se além disso f é estritamente convexa, então existe um único mínimo global de f .

(ii) Se f é convexa e o conjunto X é aberto, então $\nabla f(x^*) = 0$ é uma condição necessária e suficiente para que o vetor $x^* \in X$ seja um minimizador global de f sobre X .

A demonstração dessa proposição está disponível em [6].

2.3 Método de Newton

Considere o problema de

$$\begin{aligned} \min \quad & f(x) \\ \text{s.a.} \quad & x \in \mathfrak{R}^n. \end{aligned}$$

Se x^k é tal que $\nabla f(x^k) \neq 0$, os passos para determinar x^{k+1} são:

Passo 1: Determinar d_k tal que

$$\nabla^2 f(x^k) d_k = -\nabla f(x^k),$$

ou seja, resolver este sistema linear. Notemos que este passo pode não estar bem definido se $\nabla^2 f(x^k)$ for singular.

Passo 2: Fazer $x^{k+1} = x^k + d_k$.

A ideia do método de Newton é que a função f a ser minimizada é aproximada localmente por uma função quadrática que é exatamente minimizada. Assim, na vizinhança de x^k podemos aproximar f pela série de Taylor truncada:

$$f(x) \approx f(x^k) + \nabla f(x^k)^t (x - x^k) + \frac{1}{2} (x - x^k)^t \nabla^2 f(x^k) (x - x^k).$$

Dessa equação temos:

$$f(x) \approx f(x^k) + \nabla f(x^k)^t x - \nabla f(x^k)^t x^k + \frac{1}{2} x^t \nabla^2 f(x^k) x - x^t \nabla^2 f(x^k) x^k + \frac{1}{2} x^{k^t} \nabla^2 f(x^k) x^k.$$

Derivando ambos os membros dessa equação com relação a x , e considerando que devemos ter $\nabla f(x) = 0$, pois estamos buscando um minimizador para f , obtemos:

$$0 \approx 0 + \nabla f(x^k) + \nabla^2 f(x^k) x - \nabla^2 f(x^k) x^k.$$

Fazendo $x = x^{k+1}$, temos:

$$x^{k+1} = x^k - \nabla^2 f(x^k)^{-1} \nabla f(x^k).$$

Observe que o método de Newton pode ser visto como uma técnica iterativa para resolver um sistema de equações da forma $F(x) = 0$, onde $F : \mathfrak{R}^n \rightarrow \mathfrak{R}^n$. Basta fazer $F(x) = \nabla f(x)$. Nesta notação, o método de Newton toma a forma:

$$x^{k+1} = x^k - \nabla F(x^k)^{-1} F(x^k), \text{ onde } \nabla F(x^k) \text{ é a matriz Jacobiana de } f.$$

Para esse método temos o seguinte resultado:

Teorema: Seja $f \in C^3$ em \mathfrak{R}^n , e assumamos que no minimizador x^* a Hessiana $\nabla^2 f(x^*)$ é definida positiva. Então, se o ponto inicial for suficientemente próxima de x^* , os pontos gerados pelo método de Newton convergem para x^* . A ordem de convergência é pelo menos quadrática.

Demonstração: Ver [17].

2.4 Método Barreira Logarítmica

Considere o problema de otimização não-linear:

$$\begin{aligned} \min \quad & f(x) \\ \text{s.a} \quad & h(x) = 0, \\ & x \geq 0 \end{aligned}$$

onde $f : \mathfrak{R}^n \rightarrow \mathfrak{R}$ e $h : \mathfrak{R}^n \rightarrow \mathfrak{R}^m$.

No método de ponto interior barreira logarítmica, relaxa-se a restrição de não negatividade $x \geq 0$, inserindo-a na função objetivo f através do parâmetro barreira μ , com $\mu \rightarrow 0$. Obtemos dessa forma um novo problema cuja solução é a mesma do problema anterior:

$$\begin{aligned} \min \quad & f(x) - \mu \sum_{i=1}^n \ln(x_i) \\ \text{s.a} \quad & h(x) = 0. \end{aligned}$$

Esse novo problema é equivalente ao seguinte problema Lagrangiano:

$$\min L(x, \lambda) = f(x) - \mu \sum_{i=1}^n \ln(x_i) + \lambda^t h(x), \quad (2.1)$$

onde λ é o multiplicador de Lagrange.

Utilizando o método de Newton para resolver esse problema, teremos:

$$\nabla^2 L(x, \lambda) d = -\nabla L(x, \lambda).$$

Definindo X^{-1} da forma:

$$X^{-1} = \text{diag} \left(\frac{1}{x_i} \right) = \begin{cases} \frac{1}{x_i}, & \text{se } i = j, \\ 0, & \text{se } i \neq j, \end{cases}$$

e $J(h(x)) = \begin{bmatrix} \nabla h_1(x)^t \\ \nabla h_2(x)^t \\ \vdots \\ \nabla h_m(x)^t \end{bmatrix}$ a Jacobiana de h em x , temos então:

$$\nabla L(x, \lambda) = \begin{bmatrix} \nabla f(x) - \mu X^{-1} e + \sum_{i=1}^m \lambda_i \nabla h_i(x) \\ h(x) \end{bmatrix},$$

onde $e = [1, 1, \dots, 1]^t$, $e \in \Re^n$.

Observe que:

$$\nabla^2 L(x, \lambda) = \begin{bmatrix} \nabla^2 f(x) + \mu X^{-2} + \sum_{i=1}^m \lambda_i \nabla^2 h_i(x) & J(h(x))^t \\ J(h(x)) & 0 \end{bmatrix}.$$

Para encontrar a direção de busca desse método devemos resolver o seguinte sistema linear:

$$\begin{aligned} & \begin{bmatrix} \nabla^2 f(x) + \mu X^{-2} + \sum_{i=1}^m \lambda_i \nabla^2 h_i(x) & J(h(x))^t \\ J(h(x)) & 0 \end{bmatrix} \begin{bmatrix} dx \\ d\lambda \end{bmatrix} = \\ & = - \begin{bmatrix} \nabla f(x) - \mu X^{-1} e + \sum_{i=1}^m \lambda_i \nabla h_i(x) \\ h(x) \end{bmatrix}. \end{aligned} \quad (2.2)$$

Resolvendo esse sistema, encontramos a direção de Newton. Para encontrar o novo ponto devemos fazer $x^{k+1} = x^k + \alpha d_k$, $\lambda^{k+1} = \lambda^k + \alpha d_k$, onde

$$\alpha = \min \{1, \tau \rho\}, \text{ com } \tau = 0,99995 \text{ e } \rho = \min_{dx_i < 0} \left\{ -\frac{x_i}{dx_i} \right\}. \quad (2.3)$$

O valor sugerido para τ pode ser encontrado em [26].

A atualização do parâmetro barreira μ pode ser calculada como em [28], da forma:

$$\mu_{k+1} = \beta \mu_k, \text{ com } \beta \in (0, 1).$$

O critério de convergência é baseado nas condições de otimalidade:

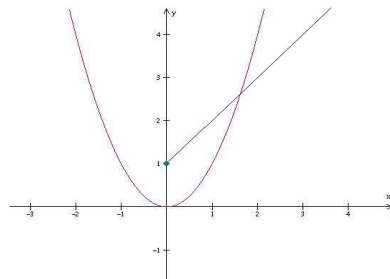
$$\|\nabla L\| < \epsilon, \quad (2.4)$$

onde ϵ é uma tolerância pré-determinada.

EXEMPLO: Vejamos um exemplo simples de um problema de otimização não-linear em \mathfrak{R}^2 .

$$\begin{aligned} \min \quad & x^2 \\ \text{s.a.} \quad & y = 1 + x \\ & x \geq 0 \end{aligned} \quad (2.5)$$

Graficamente:

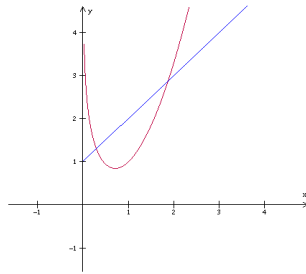


Como a função objetivo $f(x) = x^2$ é contínua e coerciva no conjunto viável, e o conjunto viável é fechado, pelo Teorema de Weierstrass, o problema possui mínimo global.

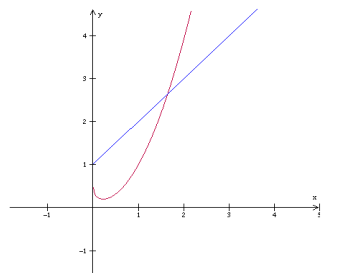
Observe que a solução ótima desse problema é o ponto $(x^*, y^*) = (0, 1)$ pertencente à reta $y = 1 + x$, para esse ponto o valor da função objetivo será 0. O problema de barreira logarítmica associado é:

$$\begin{aligned} \min \quad & x^2 - \mu \ln(x) \\ \text{s.a.} \quad & y = 1 + x. \end{aligned} \tag{2.6}$$

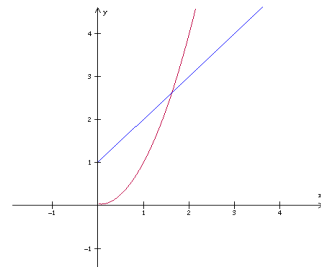
Se fizermos $\mu \rightarrow 0$, quando aplicamos o Método de Newton às condições de otimalidade do problema de barreira, a solução tende à solução do problema original. Observe graficamente que quando fazemos $\mu \rightarrow 0$, a função objetivo do problema de barreira tende à função objetivo do problema original, mantendo sempre $x > 0$:



$$\mu = 1.$$



$$\mu = 0,1.$$



$$\mu = 0,01.$$

Vamos dar um passo do Método Barreira Logarítmica. A função Lagrangiana associada ao problema (2.6) é:

$$L(x, y, \lambda) = x^2 - \mu \ln(x) + \lambda(y - 1 - x),$$

onde λ é o vetor multiplicador de Lagrange.

Obtemos assim as condições de otimalidade:

$$\nabla L(x, y, \lambda) = \begin{bmatrix} 2x - \frac{\mu}{x} - \lambda \\ \lambda \\ y - 1 - x \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}.$$

Reescrevendo esse sistema obtemos:

$$\nabla L(x, y, \lambda) = \begin{bmatrix} 2x^2 - \lambda x \\ \lambda \\ y - 1 - x \end{bmatrix} = \begin{bmatrix} \mu \\ 0 \\ 0 \end{bmatrix}. \quad (2.7)$$

Aplicando o Método de Newton ao sistema (2.7):

$$\begin{bmatrix} 4x - \lambda & 0 & -x \\ 0 & 0 & 1 \\ -1 & 1 & 0 \end{bmatrix} \begin{bmatrix} dx \\ dy \\ d\lambda \end{bmatrix} = \begin{bmatrix} -2x^2 + \lambda x + \mu \\ -\lambda \\ -y + 1 + x \end{bmatrix}. \quad (2.8)$$

Devemos partir de um ponto interior qualquer, seja então $(x^0, y^0, \lambda^0) = (2, 2, 2)$, e seja $\mu^0 = 1$, temos:

$$\begin{bmatrix} 6 & 0 & -2 \\ 0 & 0 & 1 \\ -1 & 1 & 0 \end{bmatrix} \begin{bmatrix} dx \\ dy \\ d\lambda \end{bmatrix} = \begin{bmatrix} -3 \\ -2 \\ 1 \end{bmatrix} \Rightarrow$$

$$\begin{bmatrix} dx^0 \\ dy^0 \\ d\lambda^0 \end{bmatrix} = \begin{bmatrix} -1,1667 \\ -0,1667 \\ -2,0000 \end{bmatrix}.$$

Cálculo do tamanho do passo α :

$$\rho = \min \left\{ -\frac{2}{(-1, 1667)} \right\} = 1,7142 \Rightarrow \alpha = \min \{1; (0,99995)(1,7142)\} = 1.$$

Obtemos o próximo ponto:

$$\begin{bmatrix} x^1 \\ y^1 \\ \lambda^1 \end{bmatrix} = \begin{bmatrix} x^0 \\ y^0 \\ \lambda^0 \end{bmatrix} + 1 \begin{bmatrix} dx^0 \\ dy^0 \\ d\lambda^0 \end{bmatrix} = \begin{bmatrix} 0,8333 \\ 1,8333 \\ 0 \end{bmatrix}.$$

Para o próximo passo podemos usar $\beta = \frac{1}{10}$ para a atualização do parâmetro barreira, assim:

$$\mu^1 = \frac{\mu^0}{10} = \frac{1}{10} = 0,1.$$

Tendo (x^1, y^1, λ^1) e μ^1 deve-se ir para o próximo passo e continuar até que satisfaça o critério de convergência dado por (2.4).

2.5 Método Barreira Logarítmica Preditor-Corretor

O passo preditor consiste inicialmente em encontrar uma direção para o problema de otimização não-linear barreira logarítmica (2.1) sem perturbação, ou seja, devemos fazer $\mu = 0$ no sistema de equações:

$$\nabla L(x, \lambda) = \begin{bmatrix} \nabla f(x) - \mu X^{-1}e + \sum_{i=1}^m \lambda_i \nabla h_i(x) \\ h(x) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \quad (2.9)$$

Podemos reescrever esse sistema da forma equivalente:

$$\begin{bmatrix} \nabla f(x) + \sum_{i=1}^m \lambda_i \nabla h_i(x) \\ h(x) \end{bmatrix} = \begin{bmatrix} \mu X^{-1}e \\ 0 \end{bmatrix}$$

$$\Rightarrow \begin{bmatrix} X\nabla f(x) + X \sum_{i=1}^m \lambda_i \nabla h_i(x) \\ h(x) \end{bmatrix} = \begin{bmatrix} \mu e \\ 0 \end{bmatrix}.$$

Fazendo $\mu = 0$, teremos:

$$\begin{bmatrix} X\nabla f(x) + X \sum_{i=1}^m \lambda_i \nabla h_i(x) \\ h(x) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \quad (2.10)$$

Aplicamos então o Método de Newton ao problema (2.10) para encontrar a direção do passo preditor.

No passo corretor temos que calcular a correção não-linear e perturbarmos. Para encontrarmos a correção não-linear substituímos x por $x + dx$ e λ por $\lambda + d\lambda$ no sistema de equações (2.9) e comparamos com a direção encontrada no passo preditor quando resolvemos o sistema (2.2).

O tamanho do passo é calculado como em (2.3).

O critério de convergência é como em (2.4).

Para saber mais sobre método preditor-corretor, ver [28].

EXEMPLO: Considere o mesmo exemplo dado em (2.5).

Passo preditor: Devemos fazer $\mu = 0$ no sistema (2.7) e aplicar o Método de Newton, obtendo o sistema:

$$\begin{bmatrix} 4x - \lambda & 0 & -x \\ 0 & 0 & 1 \\ -1 & 1 & 0 \end{bmatrix} \begin{bmatrix} \bar{d}x \\ \bar{d}y \\ \bar{d}\lambda \end{bmatrix} = \begin{bmatrix} -2x^2 + \lambda x \\ -\lambda \\ -y + 1 + x \end{bmatrix}.$$

Partindo de um ponto interior qualquer, $(x^0, y^0, \lambda^0) = (2, 2, 2)$, temos:

$$\begin{bmatrix} 6 & 0 & -2 \\ 0 & 0 & 1 \\ -1 & 1 & 0 \end{bmatrix} \begin{bmatrix} \bar{d}x \\ \bar{d}y \\ \bar{d}\lambda \end{bmatrix} = \begin{bmatrix} -4 \\ -2 \\ 1 \end{bmatrix} \Rightarrow$$

$$\begin{bmatrix} \bar{d}x \\ \bar{d}y \\ \bar{d}\lambda \end{bmatrix} = \begin{bmatrix} -1,3333 \\ -0,3333 \\ -2,0000 \end{bmatrix}.$$

Vamos usar a direção \bar{d} encontrada para calcular o resíduo:

Substituindo x por $x + dx$ e λ por $\lambda + d\lambda$ na primeira equação de (2.7):

$$2(x + dx)^2 - \lambda(x + dx) = \mu.$$

Na primeira equação de (2.8) temos:

$$(4x - \lambda)dx - xd\lambda = \mu - 2x^2 + \lambda x.$$

Dessa forma o resíduo nessa equação será:

$$R_1 = (2(x + dx)^2 - \lambda(x + dx) - \mu) - ((4x - \lambda)dx - xd\lambda - \mu + 2x^2 - \lambda x) \Rightarrow$$

$$R_1 = (2x^2 + 4xdx + 2dx^2 - \lambda x - \lambda dx - xd\lambda - dx d\lambda - \mu) - (4xdx - \lambda dx - xd\lambda - \mu + 2x^2 - \lambda x) \Rightarrow$$

$$R_1 = 2dx^2 - dx d\lambda.$$

Substituindo λ por $\lambda + d\lambda$ na segunda equação de (2.7) e comparando com a segunda equação de (2.8):

$$R_2 = (\lambda + d\lambda) - (d\lambda + \lambda) \Rightarrow R_2 = 0.$$

Substituindo x por $x + dx$ e y por $y + dy$ na terceira equação de (2.7) e comparando com a terceira equação de (2.8):

$$R_3 = (y + dy - 1 - x - dx) - (-dx + dy - x - 1 + y) \Rightarrow R_3 = 0.$$

Observe que R_2 e R_3 valem 0 porque a segunda e a terceira equações de (2.7) são lineares.

Passo corretor: Agora temos que resolver o seguinte sistema para encontrarmos a direção inicial:

$$\begin{bmatrix} 4x - \lambda & 0 & -x \\ 0 & 0 & 1 \\ -1 & 1 & 0 \end{bmatrix} \begin{bmatrix} dx \\ dy \\ d\lambda \end{bmatrix} = \begin{bmatrix} -2x^2 + \lambda x + \mu - R_1 \\ -\lambda \\ -y + 1 + x \end{bmatrix}.$$

Para encontrar R_1 usamos \bar{d} calculado no passo preditor:

$$R_1 = 2dx^2 - dx d\lambda = 2(-1, 3333)^2 - (-1, 3333)(-2, 0000) = 0, 8888.$$

Partimos de $(x^0, y^0, \lambda^0) = (2, 2, 2)$ e seja $\mu^0 = 1$:

$$\begin{bmatrix} 6 & 0 & -2 \\ 0 & 0 & 1 \\ -1 & 1 & 0 \end{bmatrix} \begin{bmatrix} dx \\ dy \\ d\lambda \end{bmatrix} = \begin{bmatrix} -3, 8888 \\ -2 \\ 1 \end{bmatrix} \Rightarrow$$

$$\begin{bmatrix} dx^0 \\ dy^0 \\ d\lambda^0 \end{bmatrix} = \begin{bmatrix} -1, 3148 \\ -0, 3148 \\ -2, 0000 \end{bmatrix}.$$

O cálculo de α é análogo ao do Método Barreira Logarítmica:

$$\rho = \min \left\{ -\frac{2}{(-1, 3148)} \right\} = 1, 5211 \Rightarrow \alpha = \min \{1; (0, 99995)(1, 5211)\} = 1.$$

Obtemos o próximo ponto:

$$\begin{bmatrix} x^1 \\ y^1 \\ \lambda^1 \end{bmatrix} = \begin{bmatrix} x^0 \\ y^0 \\ \lambda^0 \end{bmatrix} + 1 \begin{bmatrix} dx^0 \\ dy^0 \\ d\lambda^0 \end{bmatrix} = \begin{bmatrix} 0,6852 \\ 1,6852 \\ 0 \end{bmatrix}.$$

Para o próximo passo podemos usar $\beta = \frac{1}{10}$ para a atualização do parâmetro barreira, assim:

$$\mu^1 = \frac{\mu^0}{10} = \frac{1}{10} = 0,1.$$

De posse do novo ponto, vamos para o próximo passo e paramos quando o critério de convergência (2.4) for satisfeito.

Observe que o ponto $(x^1, y^1, \lambda^1) = (0,6852; 1,6852; 0)$ obtido por esse método está mais próximo da solução ótima $(0,1,0)$ do que o ponto $(x^1, y^1, \lambda^1) = (0,8333; 1,8333; 0)$ obtido pelo Método Barreira Logarítmica.

Para maiores informações sobre métodos de pontos interiores, ver [24].

Capítulo 3

Métodos Existentes para o Problema de Regressão L_p

Neste capítulo vamos apresentar o problema de regressão pela norma L_p e os métodos existentes que resolvem o problema.

3.1 O problema de Regressão pela Norma L_p

O problema de regressão pode ser escrito como:

$$\min_{x \in \mathfrak{R}^n} \|Ax - b\|_p^p, \quad (3.1)$$

onde $A \in \mathfrak{R}^{m \times n}$ é uma matriz de posto completo, $1 < p < \infty$, $b \in \mathfrak{R}^m$ e $m > n$.

Quando $p = 1$ e $p = \infty$, o problema pode ser formulado por programação linear e os métodos de pontos interiores aplicados a esses problemas permitem a exploração da estrutura matricial do problema de forma muito eficiente. A norma-2 permite solução direta.

A norma L_p , com $1 < p < \infty$ é estritamente convexa. Considere a função $f(x) = \|x\|_p^p$, onde $f : \mathfrak{R}^n \rightarrow \mathfrak{R}$ e $1 < p < \infty$. Dados $x, y \in \mathfrak{R}^n$ e dado um α qualquer no intervalo

$[0, 1]$, temos que:

$$\begin{aligned} f(\alpha x + (1 - \alpha)y) &= \|\alpha x + (1 - \alpha)y\|_p \leq |\alpha| \|x\|_p + |(1 - \alpha)| \|y\|_p = \\ &= \alpha \|x\|_p + (1 - \alpha) \|y\|_p = \alpha f(x) + (1 - \alpha)f(y) \Rightarrow \\ f(\alpha x + (1 - \alpha)y) &\leq \alpha f(x) + (1 - \alpha)f(y). \end{aligned}$$

Entretanto, se tomarmos x e y quaisquer em \mathfrak{R}^n , com $x \neq y$, e $\alpha \in (0, 1)$, a desigualdade acima será estrita, ou seja, $f(\alpha x + (1 - \alpha)y) < \alpha f(x) + (1 - \alpha)f(y)$. Podemos dizer então que a norma L_p é estritamente convexa.

Considere agora as funções $f(x) = \|x\|_p$ e $g(x) = x^p$ onde $h : \mathfrak{R}^n \rightarrow \mathfrak{R}$ e $g : \Omega \subset \mathfrak{R}_+ \rightarrow \mathfrak{R}$. A função g é convexa para $p > 1$, pois $g''(x) = p(p - 1)x^{p-2} \geq 0$ (veja que $x \geq 0$, pois o domínio Ω de g está contido em \mathfrak{R}_+). Além disso, g é não-decrescente para $p > 1$, portanto a função composta $g(f(x)) = \|x\|_p^p = h(x)$ é convexa. Podemos afirmar isso com base no teorema a seguir:

Teorema: Se $f : D_f \subset \mathfrak{R}^n \rightarrow \mathfrak{R}$ é convexa e $g : D_g \subset \mathfrak{R} \rightarrow \mathfrak{R}$ é convexa e não-decrescente em um conjunto convexo que contém a imagem de f , então $h : D_h \subset \mathfrak{R}^n \rightarrow \mathfrak{R}$, $h(x) = g(f(x))$ é convexa.

Demonstração:

Sejam $x, y \in D_h$, $\alpha \in [0, 1]$, $x_1 = f(\alpha x + (1 - \alpha)y)$ e $x_2 = \alpha f(x) + (1 - \alpha)f(y)$, como f é convexa, $x_1 \leq x_2$ e como g é não-decrescente $g(x_1) \leq g(x_2)$, então:

$$\begin{aligned} h(\alpha x + (1 - \alpha)y) &= g(f(\alpha x + (1 - \alpha)y)) = g(x_1) \leq g(x_2) = \\ &= g(\alpha f(x) + (1 - \alpha)f(y)) \leq \alpha g(f(x)) + (1 - \alpha)g(f(y)) = \alpha h(x) + (1 - \alpha)h(y). \end{aligned}$$

Temos então: $h(\alpha x + (1 - \alpha)y) \leq \alpha h(x) + (1 - \alpha)h(y)$, assim h é convexa.

Com isso, podemos dizer que o problema (3.1) possui formulação convexa e portanto é garantida a existência de mínimo global.

O problema de regressão pode ser reescrito da seguinte forma:

$$\begin{aligned} \min \quad & \|r\|_p^p \\ \text{s.a} \quad & Ax + r = b. \end{aligned} \tag{3.2}$$

Definindo $r = u - v$, com $u \geq 0$ e $v \geq 0$:

$$\begin{aligned} \min \phi(u, v) = \quad & \sum_{i=1}^m (u_i + v_i)^p \\ \text{s.a} \quad & Ax + u - v - b = 0 \\ & (u, v) \geq 0. \end{aligned} \tag{3.3}$$

Observe que minimizar $\|r\|_p^p = \sum_{i=1}^m |u_i - v_i|^p$ é equivalente a minimizar $\phi(u, v) = \sum_{i=1}^m (u_i + v_i)^p$, desde que se verifiquem as restrições $u_i v_i = 0$, $i = 1, \dots, m$. Mas, dado r_i , sempre podemos escrevê-lo como $u_i - v_i$ de forma que $u_i v_i = 0$. Por exemplo, se $r_i < 0$, basta fazer $u_i = 0$ e $v_i = -r_i$; por outro lado, se $r_i \geq 0$, podemos fazer $u_i = r_i$ e $v_i = 0$, assim teremos $\|r\|_p^p = \sum_{i=1}^m |u_i - v_i|^p = \sum_{i=1}^m (u_i + v_i)^p$.

3.2 Métodos Similares

A primeira especialização de métodos de programação linear para resolver problemas de regressão pela norma L_1 vem do algoritmo simplex (ver [27], [4], [2] e [1]).

Em [27] e [3] estão propostas formulações para resolver o problema de regressão pela norma L_∞ utilizando programação linear. Em [4] está proposto um método simplex especializado para o problema de regressão pela norma L_∞ .

Em [8] e [9] são descritos métodos que utilizam procedimento de busca linear para os problemas de regressão pelas normas L_1 e L_∞ , respectivamente.

Em [20] está descrita uma implementação eficiente para o problema de regressão pela

norma L_∞ e, em [21], uma implementação eficiente para o problema de regressão pela norma L_1 . Em ambos utilizam-se métodos de ponto interior.

Em [11] encontramos o Método de Relaxamento da Coluna para Problemas de Norma Mínima. Porém, esse método é pouco eficiente, uma vez que para alterar o valor de uma única variável ele precisa de uma iteração, sendo assim em um problema onde temos um número muito grande de variáveis, o número de iterações necessárias utilizadas por esse método para chegar ao valor ótimo será muito elevado.

3.3 Método IRLS

Segue a breve descrição do método IRLS (Iteratively Reweighted Least-Squares) desenvolvido em [19] e [22].

Considere o problema (3.2). A função objetivo é denotada em termos de r por $\phi(r) = \|r\|_p^p (= \psi(x))$. A norma L_p é estritamente convexa para $1 < p < \infty$, pois estamos supondo que $\text{posto}(A) = n$. Vamos assumir que não existe x tal que $Ax - b = 0$.

Defina $r = Ax - b$ e σ_i como o sinal de $r_i = (Ax - b)_i$, i -ésima coordenada de r . Seja $\sigma = (\sigma_i)_{i=1, \dots, m}$. A solução ocorre no ponto onde o gradiente $\nabla\psi(x) = pA^t(|Ax - b|)^{p-1}\sigma$ vale zero, onde $(|Ax - b|)^{p-1}$ é um vetor cujas entradas são $|a_i^t x - b_i|^{p-1}$, $i = 1, 2, \dots, m$, a_i representa a i -ésima linha da matriz A . Dizer que $\nabla\psi(x) = pA^t(|Ax - b|)^{p-1}\sigma$ vale zero significa dizer que todas as coordenadas desse vetor valem zero. Supomos que existe um ponto com $r_i \neq 0$, $1 \leq i \leq m$. Assim,

$$\nabla\psi(x) = pA^t(|Ax - b|)^{p-1}\sigma = 0$$

$$\Rightarrow A^t(|Ax - b|)^{p-1} = 0$$

é equivalente a

$$A^t \text{diag}((|Ax - b|)^{p-1})r = 0.$$

Defina $D = \text{diag}(|Ax - b|)^{(2-p)/2}$, ou seja, $d_{ii} = (|Ax - b|_i)^{(2-p)/2}$ e $d_{ij} = 0$ para $i \neq j$, com $i, j = 1, 2, \dots, m$. Então a solução desse problema ocorre onde $A^t(D)^{-2}r = 0$. Essa equação vem do seguinte sistema de quadrados mínimos:

$$(D)^{-1}Ax = (D)^{-1}b.$$

Método IRLS

Dado um ponto inicial x^0

Passo 1: Calcule $r^k = Ax^k - b$;

Passo 2: Defina $D^k = \text{diag}(|Ax^k - b|)^{(2-p)/2}$, encontre x^{k+1} de:

$$(D^k)^{-1}Ax^{k+1} = (D^k)^{-1}b; \quad k \leftarrow k + 1;$$

Vá para o passo 1.

Critério de convergência: O critério de convergência usado em [16] é:

$$\frac{|\phi(r^{k+1}) - \phi(r^k)|}{\phi(r^{k+1})} < \tau_s, \text{ ou } \eta^k < \tau_s, \text{ ou } k > 50, \quad (3.4)$$

onde τ_s foi definido como $\frac{1}{2}10^{-11}$ e k denota o número de iterações.

Em [22] está demonstrado que se assumirmos $1 < p < 2$ e $r_i^k \neq 0$, $1 \leq i \leq m$, o limite da sequência de pontos $\{x^k\}$ gerada pelo algoritmo IRLS é a solução para o problema e a convergência é linear com taxa de convergência constante $2 - p$.

3.4 Método IRLSL

Segue a breve descrição do método IRLSL desenvolvido em [16].

O método IRLSL é o método IRLS modificado com uma busca linear. Considere o problema (3.2).

Método IRLSL

Dado um ponto inicial $r^0 = Ax^0 - b$, com $|r^0| > 0$, ou seja $r_i \neq 0, \forall i = 1, 2, \dots, m$.

Calcule:

- $g^k = p(|r^k|)^{p-1}\sigma^k$, onde $g_i^k = p(|r_i^k|)^{p-1}\sigma_i^k, i = 1, 2, \dots, m$.
- $D_r^k = \text{diag}(|r^k|)$, onde $(D_r^k)_{ij} = \begin{cases} |r_i^k|, & \text{se } i = j, \\ 0, & \text{se } i \neq j. \end{cases}$
- $D^k = (D_r^k \text{diag}((p-1)|g^k|)^{-1})^{1/2}$, onde $(D^k)_{ij} = \begin{cases} \left(\frac{|r_i^k|}{(p-1)|g_i^k|} \right)^{1/2}, & \text{se } i = j, \\ 0, & \text{se } i \neq j. \end{cases}$

Calcule a direção d^k resolvendo o sistema:

$$\begin{cases} (D^k)^{-1} A d^k = -D^k g^k \\ d^k = A d_x^k; \end{cases}$$

Realizar a busca linear para encontrar α^k .

Procedimento de busca linear:

Dados $\tau^k, \beta_f \in (0, 1)$, $d^k, r^k, \tilde{\alpha}^k, \rho_b > 0$ (por ex. 10^6) e α_i^k , onde:

$$\Gamma = \left\{ \alpha_i^k : \alpha_i^k = -\frac{r_i^k}{d_i^k}, \quad r_i^k d_i^k < 0 \right\}.$$

Faça $\tau = 0,975$ (ver [16]) e calcule:

$$\tau^k = \max \left\{ \tau, 1 - \frac{\|A^t g^k\|_2}{1 + \|A^t g^k\|_2} \right\}$$

$$\tilde{\alpha}^k = -\frac{(g^k)^t d^k}{(d^k)^t \text{diag}(p(|r^k|)^{p-2}) d^k}$$

Passo 1: Seja α_*^k o menor ponto positivo em $[\tilde{\alpha}^k, \rho_b]$ com $g(r^k + \tilde{\alpha}^k d^k)^t d^k \geq 0$.

Se α_*^k existe e

$$\phi(r^{k+1}) \leq \phi(r^k) + \beta_f \alpha_*^k \nabla \phi(r^k)^t d^k,$$

onde $r^{k+1} = r^k + \alpha_*^k d^k$ é satisfeita, defina

$$\alpha_{\#}^k \leftarrow \max \{ \alpha_i^k : 0 \leq \alpha_i^k \leq \alpha_*^k \} \text{ e}$$

$$\alpha^k \leftarrow \alpha_{\#}^k + \tau^k (\alpha_*^k - \alpha_{\#}^k).$$

Caso contrário, retorne.

Passo 2: Se $\phi(r^{k+1}) > \phi(r^k) + \beta_f \nabla \phi(r^k)^t d^k$, onde $r^{k+1} = r^k + d^k$, vá para o próximo passo.

Caso contrário, defina

$$\alpha^k \leftarrow \begin{cases} 1, & \text{se } \min(|r^k + d^k|) > 0, \\ \alpha_{\#}^k + \tau^k (1 - \alpha_{\#}^k), & \text{caso contrário.} \end{cases}$$

onde $\alpha_{\#}^k \leftarrow \max \{ \alpha_i^k : 0 \leq \alpha_i^k \leq 1 \}$, retorna.

Passo 3: Defina

$$\alpha^k \leftarrow \begin{cases} \tilde{\alpha}^k, & \text{se } \min(|r^k + \tilde{\alpha}^k d^k|) > 0, \\ \alpha_{\#}^k + \tau^k (\tilde{\alpha}^k - \alpha_{\#}^k), & \text{caso contrário.} \end{cases}$$

onde $\alpha_{\#}^k \leftarrow \max \{ \alpha_i^k : 0 \leq \alpha_i^k \leq \tilde{\alpha}^k \}$, retorna.

Fim

Com α^k calculado pelo procedimento de busca linear, faça

- $r^{k+1} \leftarrow r^k + \alpha^k d^k, \quad k \leftarrow k + 1;$

O procedimento de busca linear faz com que o método tenha convergência global e se não existir resíduo nulo na solução, esse método tem convergência quadrática (ver [16]).

Critério de convergência: O critério de convergência é o mesmo que (3.4), descrito no Método IRLS.

3.5 Método GNCS

O método conhecido como GNCS é um método de Newton globalizado que usa as condições de folgas complementares para o problema da norma L_p .

O método IRLSL funciona bem quando p está suficientemente longe de 1. O método GNCS, descrito em [16], é mais rápido que o método IRLSL quando p está próximo de 1.

Método GNCS

Dado um ponto inicial $r^0 = Ax^0 - b$, com $|r^0| > 0$ e $\lambda^0 = \tau \frac{g^0}{\max(|r^0|)}$, com $\tau = 0,975$.

Passo 1: Calcule:

- $D_r^k = \text{diag}(|r^k|)$, onde $r^k = Ax^k - b$
- $g^k = p(|r^k|^{p-1})\sigma$

(No método IRLS temos $A^t(D)^{-2}r = 0$, essa equação é equivalente a escrever $g - Z^t w = 0$, onde as linhas da matriz Z formam uma base para o espaço nulo de A^t , isto é, $A^t Z^t = 0$, note que $x \in \Re^n$ e $w \in \Re^{m-n}$ e $\lambda^k = Z^t w^k$)

- $\phi(r^0) = \|r^0\|_p^p$.

Dado $0 < \gamma < 1$ (na implementação usa-se $\gamma = 0.99$), calcule:

- $\eta^k = \max \left\{ \max \left\{ \frac{|D_r^k(g^k - \lambda^k)|}{\phi(r^0)} \right\}, \max \{ \max \{ |\lambda^k| - |g^k|, 0 \} \} \right\}$

Em outras palavras, η^k é o máximo da violação da condição de folga complementar ($D_r(g - \lambda) = 0$) e da viabilidade dual ($|\lambda| \leq |g|$). Note que $|g| = |p(|r|)^{p-1}| = [1, 1, \dots, 1]$ quando $p = 1$. Neste caso, $\theta = 0$ (ou $\eta = 0$) é uma condição necessária e suficiente de otimalidade (Para mais detalhes ver [9]).

- $\theta^k = \text{diag} \left(\frac{1}{(\gamma |g^k| + \eta^k e)} \right) \eta^k e$, onde $e = [1, 1, \dots, 1]^t \in \Re^m$.
- $D_\theta^k = \text{diag}(|pg^k - \text{diag}(\lambda^k)(e - \theta^k)|)$
- $D^k = (D_r^k(D_\theta^k)^{-1})^{1/2}$

Calcule a direção d^k por:

$$\begin{cases} (D^k)^{-1} A d^k = -D^k g^k \\ d^k = A d_x^k; \end{cases}$$

Atualize λ^{k+1} :

$$\lambda^{k+1} \leftarrow (D_r^k)^{-1} D_\theta^k d^k + g^k$$

Faça $\tau = 0,975$ e calcule:

$$\tau^k = \max \left\{ \tau, 1 - \frac{\eta^k}{\gamma + \eta^k} \right\}$$

$$\tilde{\alpha}^k = - \frac{(g^k)^t d^k}{(d^k)^t \text{diag}(p(|r^k|)^{p-2}) d^k}$$

Procedimento de busca linear:

O procedimento de busca linear utilizado para encontrar α^k é o mesmo procedimento descrito no Método IRLS na Seção 3.5.

Com α^k calculado pelo procedimento de busca linear, faça:

- $r^{k+1} \leftarrow r^k + \alpha^k d^k, \quad k \leftarrow k + 1.$

Critério de convergência: O critério de convergência é o mesmo que (3.4), descrito no Método IRLS. Esse método tem convergência superlinear (ver [16]).

3.6 Métodos de Pontos Interiores Aplicados ao Problema de Regressão pela Norma L_p

Todos os métodos que serão apresentados nessa seção foram desenvolvidos em [7]. Entretanto apresentaremos os métodos barreira logarítmica preditor-corretor e primal-dual barreira logarítmica preditor-corretor com a modificação no cálculo dos resíduos.

Para todos esses métodos vamos considerar o problema da Seção 3.1.

3.6.1 Método Barreira Logarítmica

O problema de barreira associado ao problema (3.3) é:

$$\begin{aligned} \min \quad & \sum_{i=1}^m (u_i + v_i)^p - \mu \sum_{i=1}^m \ln(u_i) - \mu \sum_{i=1}^m \ln(v_i) , \\ \text{s.a.} \quad & Ax + u - v - b = 0 \end{aligned} \tag{3.5}$$

onde $\mu > 0$ é o parâmetro barreira, $A \in \mathbb{R}^{m \times n}$, $u, v, b \in \mathbb{R}^m$, $m > n$ e $1 < p < \infty$.

A função Lagrangiana associada ao problema de barreira é:

$$L(x, y, u, v) = \sum_{i=1}^m (u_i + v_i)^p - \mu \sum_{i=1}^m \ln(u_i) - \mu \sum_{i=1}^m \ln(v_i) + y^t (Ax + u - v - b),$$

onde y é vetor multiplicador de Lagrange.

Temos que

$$\nabla \phi(u, v) = \begin{bmatrix} \partial \phi(u, v) / \partial u \\ \partial \phi(u, v) / \partial v \end{bmatrix} = \begin{bmatrix} p(u + v)^{p-1} \\ p(u + v)^{p-1} \end{bmatrix} = \begin{bmatrix} g \\ g \end{bmatrix}$$

é o gradiente da função objetivo do problema original dada por $\phi(u, v) = \sum_{i=1}^m (u_i + v_i)^p$, $g = p(u + v)^{p-1}$, onde $g_i = p(u_i + v_i)^{p-1}$, $i = 1, \dots, m$.

$$\nabla^2 \phi(u, v) = G = \begin{cases} \frac{p(p-1)}{(u_i + v_i)^{2-p}}, & \text{se } i = j, \\ 0, & \text{se } i \neq j. \end{cases}$$

Observe que G é uma matriz diagonal.

Aplicando as condições de otimalidade do problema de barreira, devemos ter:

$$\underbrace{\nabla L}_{J(x,y,u,v)} = \begin{bmatrix} A^t y \\ Ax + u - v - b \\ g - \mu U^{-1} e + y \\ g - \mu V^{-1} e - y \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix},$$

onde:

$$e = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}, \quad U = \text{diag}(u) = \begin{cases} u_i, & \text{se } i = j, \\ 0, & \text{se } i \neq j. \end{cases} \quad \text{e} \quad V = \text{diag}(v) = \begin{cases} v_i, & \text{se } i = j, \\ 0, & \text{se } i \neq j. \end{cases}$$

Esse sistema de equações pode ser reescrito da seguinte forma:

$$\underbrace{\nabla L}_{J(x,y,u,v)} = \begin{bmatrix} A^t y \\ Ax + u - v - b \\ U(g + y) \\ V(g - y) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \mu e \\ \mu e \end{bmatrix}$$

Aplicando o método de Newton a esse problema temos:

$$\begin{bmatrix} 0 & A^t & 0 & 0 \\ A & 0 & I & -I \\ 0 & U & \text{diag}(g + y) + UG & UG \\ 0 & -V & VG & \text{diag}(g - y) + VG \end{bmatrix} \begin{bmatrix} dx \\ dy \\ du \\ dv \end{bmatrix} = \begin{bmatrix} r_1 \\ r_2 \\ r_3 \\ r_4 \end{bmatrix},$$

onde:

$$\begin{aligned} r_1 &= -A^t y, \\ r_2 &= -Ax - u + v + b, \\ r_3 &= -U(g + y) + \mu e, \\ r_4 &= -V(g - y) + \mu e, \\ \text{diag}(g + y) &= \begin{cases} g_i + y_i, & \text{se } i = j, \\ 0, & \text{se } i \neq j. \end{cases} \\ \text{diag}(g - y) &= \begin{cases} g_i - y_i, & \text{se } i = j, \\ 0, & \text{se } i \neq j. \end{cases} \end{aligned}$$

Desse sistema calculamos as direções e atualizamos as variáveis seguindo o método.

Algoritmo Barreira Logarítmica

Ponto inicial:

$$\begin{aligned}
 x^0 &= (A^t A)^{-1} A^t b, \\
 r^0 &= b - Ax^0, \\
 u_i^0 &= \begin{cases} \frac{\zeta + 1}{2} r_i^0, & \text{se } r_i^0 > 0, \\ \frac{1 - \zeta}{2} r_i^0, & \text{caso contrário.} \end{cases} \\
 v_i^0 &= \begin{cases} \frac{\zeta - 1}{2} r_i^0, & \text{se } r_i^0 > 0, \\ \frac{-1 - \zeta}{2} r_i^0, & \text{caso contrário.} \end{cases} \\
 y^0 &= \frac{\delta r^0}{\|r^0\|_\infty}, \text{ com } \delta = 0.975.
 \end{aligned} \tag{3.6}$$

O ponto inicial é calculado como em [7]. Devemos usar $\zeta > 1$ para que u e v sejam pontos interiores factíveis.

Assim, dados $x^0, y^0, (u^0, v^0) > 0, \mu^0, \beta > 1$ e $\tau = 0,99995$.

O valor sugerido para τ pode ser encontrado em [26].

Para $k = 0, 1, 2, \dots$ faça:

$$\begin{aligned}
 g^k &= p(u^k + v^k)^{p-1} \\
 G^k &= \begin{cases} \frac{p(p-1)}{(u_i^k + v_i^k)^{2-p}}, & \text{se } i = j, \\ 0, & \text{se } i \neq j. \end{cases} \\
 U^k &= \text{diag}(u^k) \\
 V^k &= \text{diag}(v^k) \\
 r_1^k &= -A^t y^k \\
 r_2^k &= -Ax^k - u^k + v^k + b \\
 r_3^k &= -U^k(g^k + y^k) + \mu^k e \\
 r_4^k &= -V^k(g^k - y^k) + \mu^k e
 \end{aligned}$$

$$\begin{aligned}
D_u^k &= [\text{diag}(g^k + y^k) + U^k G^k] \\
D_v^k &= [\text{diag}(g^k - y^k) + V^k G^k] \\
D_\phi^k &= [D_v^k - V^k U^k (G^k)^2 (D_u^k)^{-1}] \\
D^k &= [-(D_u^k)^{-1} U^k - (D_\phi^k)^{-1} V^k (I + (D_u^k)^{-1} U^k G^k)^2]^{-1} \\
r^k &= -\{r_1^k - A^t D^k [r_2^k - (D_u^k)^{-1} r_3^k + (D_\phi^k)^{-1} (I + (D_u^k)^{-1} U^k G^k)(r_4^k - (D_u^k)^{-1} V^k G^k r_3^k)]\} \\
dx^k &= (A^t D^k A)^{-1} r^k \\
dy^k &= D^k [r_2^k - A dx^k - (D_u^k)^{-1} r_3^k + (D_\phi^k)^{-1} (I + (D_u^k)^{-1} U^k G^k)(r_4^k - (D_u^k)^{-1} V^k G^k r_3^k)] \\
dv^k &= (D_\phi^k)^{-1} [r_4^k + V^k dy^k + (D_u^k)^{-1} (-V^k G^k r_3^k + V^k U^k G^k dy^k)] \\
du^k &= (D_u^k)^{-1} (r_3^k - U^k dy^k - U^k G^k dv^k) \\
\alpha^k &= \min \left\{ \tau \left(\min_{du_i^k < 0} -\frac{u_i^k}{du_i^k} \right), \tau \left(\min_{dv_i^k < 0} -\frac{v_i^k}{dv_i^k} \right), 1 \right\} \\
x^{k+1} &= x^k + \alpha^k dx^k \\
y^{k+1} &= y^k + \alpha^k dy^k \\
u^{k+1} &= u^k + \alpha^k du^k \\
v^{k+1} &= v^k + \alpha^k dv^k \\
\mu^{k+1} &= \frac{\mu^k}{\beta}
\end{aligned}$$

Até convergir.

Critério de Convergência

Utilizamos dois critérios de convergência, um baseado nas condições de otimalidade e o outro baseado na diferença dos valores de N atual e da iteração anterior.

$$N^k = \frac{\|\nabla L^k\|}{(1 + \|x^k\| + \|u^k\| + \|v^k\| + \|y^k\|)(2m)} \leq \epsilon, \tag{3.7}$$

$$|N^{k+1} - N^k| \leq \epsilon_1.$$

Em N^k , m representa o número de linhas da matriz A , ϵ e ϵ_1 são tolerâncias pré-determinadas.

3.6.2 Método Barreira Logarítmica Predictor-Corretor

Esse método foi desenvolvido em [7], e será apresentado aqui com algumas modificações no cálculo dos resíduos.

Considere o problema de barreira (3.5).

Passo predictor: Devemos fixar o parâmetro barreira $\mu = 0$ e calcularmos a direção resolvendo o sistema:

$$\begin{bmatrix} 0 & A^t & 0 & 0 \\ A & 0 & I & -I \\ 0 & U & \text{diag}(g + y) + UG & UG \\ 0 & -V & VG & \text{diag}(g - y) + VG \end{bmatrix} \begin{bmatrix} \bar{d}x \\ \bar{d}y \\ \bar{d}u \\ \bar{d}v \end{bmatrix} = \begin{bmatrix} \bar{r}_1 \\ \bar{r}_2 \\ \bar{r}_3 \\ \bar{r}_4 \end{bmatrix},$$

onde:

$$\begin{aligned} \bar{r}_1 &= -A^t y, \\ \bar{r}_2 &= -Ax - u + v + b, \\ \bar{r}_3 &= -U(g + y), \\ \bar{r}_4 &= -V(g - y). \end{aligned}$$

Assim temos o sistema:

$$A^t \bar{d}y = \bar{r}_1 \quad (1)$$

$$A \bar{d}x + \bar{d}u - \bar{d}v = \bar{r}_2 \quad (2)$$

$$U \bar{d}y + [\text{diag}(g + y) + UG] \bar{d}u + UG \bar{d}v = \bar{r}_3 \quad (3)$$

$$-V \bar{d}y + VG \bar{d}u + [\text{diag}(g - y) + VG] \bar{d}v = \bar{r}_4 \quad (4)$$

Isolando $\bar{d}u$ da equação (3), substituindo $\bar{d}u$ na equação (4) e isolando $\bar{d}v$, substi-

tuindo $\bar{d}v$ na equação (2) e isolando $\bar{d}y$, substituindo $\bar{d}y$ na equação (1), obtemos $\bar{d}x$. Temos portanto o passo predictor:

$$\begin{aligned}\bar{d}x &= (A^t DA)^{-1} \bar{r}, \\ \bar{d}y &= D[\bar{r}_2 - A\bar{d}x - D_u^{-1} \bar{r}_3 + (D_v - VUG^2 D_u^{-1})^{-1} (I + D_u^{-1} UG)(\bar{r}_4 - D_u^{-1} VG\bar{r}_3)], \\ \bar{d}v &= (D_v - VUG^2 D_u^{-1})^{-1} [\bar{r}_4 + V\bar{d}y + D_u^{-1} (-VG\bar{r}_3 + VUG\bar{d}y)], \\ \bar{d}u &= D_u^{-1} (\bar{r}_3 - U\bar{d}y - UG\bar{d}v),\end{aligned}$$

onde:

$$\begin{aligned}D_u &= [\text{diag}(g + y) + UG], \\ D_v &= [\text{diag}(g - y) + VG], \\ D &= [-D_u^{-1} U - (D_v - VUG^2 D_u^{-1})^{-1} V(I + D_u^{-1} UG)^2]^{-1}, \\ \bar{r} &= -\{\bar{r}_1 - A^t D[\bar{r}_2 - D_u^{-1} \bar{r}_3 + (D_v - VUG^2 D_u^{-1})^{-1} (I + D_u^{-1} UG)(\bar{r}_4 - D_u^{-1} VG\bar{r}_3)]\},\end{aligned}$$

Considere agora os seguintes sistemas de equações:

$$\underbrace{\nabla L}_{J(x,y,u,v)} = \begin{bmatrix} A^t y \\ Ax + u - v - b \\ U(g + y) \\ V(g - y) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \mu e \\ \mu e \end{bmatrix} \quad \text{e} \quad (3.8)$$

$$\begin{bmatrix} 0 & A^t & 0 & 0 \\ A & 0 & I & -I \\ 0 & U & \text{diag}(g + y) + UG & UG \\ 0 & -V & VG & \text{diag}(g - y) + VG \end{bmatrix} \begin{bmatrix} dx \\ dy \\ du \\ dv \end{bmatrix} = \begin{bmatrix} r_1 \\ r_2 \\ r_3 \\ r_4 \end{bmatrix}, \quad (3.9)$$

onde:

$$\begin{aligned} r_1 &= -A^t y, \\ r_2 &= -Ax - u + v + b, \\ r_3 &= -U(g + y) + \mu e, \\ r_4 &= -V(g - y) + \mu e. \end{aligned}$$

O sistema (3.9) é o método de Newton aplicado ao problema (3.8).

Agora vamos encontrar novos resíduos para calcular o passo corretor. Para calcular os resíduos devemos comparar os sistemas de equações (3.8), substituindo x por $x + dx$, y por $y + dy$, u por $u + du$ e v por $v + dv$ com o sistema de equações (3.9). As diferenças encontradas serão nossos resíduos.

Primeiro sistema de equações do sistema (3.8):

$$\begin{aligned} A^t(y + dy) &= 0 \\ \Rightarrow A^t dy &= -A^t y. \end{aligned}$$

Comparando com o sistema (3.9), vemos que não existe resíduo.

Segundo sistema de equações do sistema (3.8):

$$\begin{aligned} A(x + dx) + u + du - v - dv - b &= 0 \\ \Rightarrow Adx + du - dv &= b - Ax - u + v. \end{aligned}$$

Comparando com o sistema (3.9), vemos que nesse sistema também não existe resíduo.

Terceiro sistema de equações do sistema (3.8):

$$(U + dU)(\tilde{g} + y + dy) = \mu e ,$$

onde $\tilde{g}_i = p(u_i + du_i + v_i + dv_i)^{p-1}$, assim $\tilde{g} = \tilde{g}_i$, $i = 1, 2, \dots, m$. Entretanto, $p(u_i + du_i + v_i + dv_i)^{p-1}$ pode resultar em um número imaginário, para corrigir esse problema devemos fazer $\tilde{g}_i = p(u_i + \beta du_i + v_i + \beta dv_i)^{p-1}$, onde β é calculado da seguinte forma:

$$\beta = \min \left\{ \tau \left(\min_{(d_{u_i} + d_{v_i}) < 0} -\frac{(u_i + v_i)}{(d_{u_i} + d_{v_i})} \right), 1 \right\}, \text{ onde } \tau = 0,99995. \quad (3.10)$$

Com base nos testes computacionais, acreditamos que \tilde{g}_i não afeta a convergência deste método.

Nesse sistema de equações temos o seguinte resíduo:

$$R_1 = ((U + dU)(\tilde{g} + y + dy) - \mu e) - (Udy + \text{diag}(g)du + \text{diag}(y)du + UGdu + UGdv + Ug + Uy - \mu e)$$

$$\Rightarrow R_1 = (U + dU)\tilde{g} + dUdy - dUg - UGdu - UGdv - Ug.$$

Quarto sistema de equações do sistema (3.8):

$$(V + dV)(\tilde{g} - y - dy) = \mu e,$$

onde $\tilde{g} = p(u + \beta du + v + \beta dv)^{p-1}$, e β é calculada como em (3.10).

Nesse sistema de equações temos o seguinte resíduo:

$$R_2 = ((V + dV)(\tilde{g} - y - dy) - \mu e) - (-Vdy + VGdu + \text{diag}(g - y)dv + VGdv + Vg - Vy - \mu e)$$

$$\Rightarrow R_2 = (V + dV)\tilde{g} - dVdy - VGdu - dVg - VGdv - Vg.$$

Passo corretor: Agora podemos impor um valor para o parâmetro barreira μ e resolver o sistema:

$$\begin{bmatrix} 0 & A^t & 0 & 0 \\ A & 0 & I & -I \\ 0 & U & \text{diag}(g + y) + UG & UG \\ 0 & -V & VG & \text{diag}(g - y) + VG \end{bmatrix} \begin{bmatrix} \hat{d}x \\ \hat{d}y \\ \hat{d}u \\ \hat{d}v \end{bmatrix} = \begin{bmatrix} \hat{r}_1 \\ \hat{r}_2 \\ \hat{r}_3 \\ \hat{r}_4 \end{bmatrix},$$

onde:

$$\begin{aligned}
 \hat{r}_1 &= -A^t y, \\
 \hat{r}_2 &= -Ax - u + v + b, \\
 \hat{r}_3 &= -U(g + y) + \mu e - R_1, \\
 \hat{r}_4 &= -V(g - y) + \mu e - R_2, \\
 R_1 &= (U + dU)\tilde{g} + dUdy - dUg - UGdu - UGdv - Ug, \\
 R_2 &= (V + dV)\tilde{g} - dVdy - VGdu - dVg - VGdv - Vg, \\
 \tilde{g} &= p(u + \beta du + v + \beta dv)^{p-1}, \\
 \beta &= \min \left\{ \tau \left(\min_{(\bar{d}_{u_i} + \bar{d}_{v_i}) < 0} -\frac{(u_i + v_i)}{(\bar{d}_{u_i} + \bar{d}_{v_i})} \right), 1 \right\} \text{ onde } \tau = 0.9995,
 \end{aligned}$$

Obtemos dessa forma as direções:

$$\begin{aligned}
 \hat{d}x &= (A^t DA)^{-1} \hat{r}, \\
 \hat{d}y &= D[\hat{r}_2 - A\hat{d}x - D_u^{-1} \hat{r}_3 + (D_v - VUG^2 D_u^{-1})^{-1} (I + D_u^{-1} UG)(\hat{r}_4 - D_u^{-1} VG\hat{r}_3)], \\
 \hat{d}v &= (D_v - VUG^2 D_u^{-1})^{-1} [\hat{r}_4 + V\hat{d}y + D_u^{-1} (-VG\hat{r}_3 + VUG\hat{d}y)], \\
 \hat{d}u &= D_u^{-1} (\hat{r}_3 - U\hat{d}y - UG\hat{d}v),
 \end{aligned}$$

onde:

$$\begin{aligned}
 D_u &= [\text{diag}(g + y) + UG], \\
 D_v &= [\text{diag}(g - y) + VG], \\
 D &= [-D_u^{-1} U - (D_v - VUG^2 D_u^{-1})^{-1} V (I + D_u^{-1} UG)^2]^{-1}, \\
 \hat{r} &= -\{\hat{r}_1 - A^t D[\hat{r}_2 - D_u^{-1} \hat{r}_3 + (D_v - VUG^2 D_u^{-1})^{-1} (I + D_u^{-1} UG)(\hat{r}_4 - D_u^{-1} VG\hat{r}_3)]\}.
 \end{aligned}$$

Para garantir que o novos vetores u e v sejam estritamente positivos, devemos calcular o tamanho do passo α dado por:

$$\alpha = \min \left\{ \tau \left(\min_{\hat{d}_{u_i} < 0} -\frac{u_i}{\hat{d}_{u_i}} \right), \tau \left(\min_{\hat{d}_{v_i} < 0} -\frac{v_i}{\hat{d}_{v_i}} \right), 1 \right\}, \text{ onde } \tau = 0,99995.$$

Tanto no passo preditor quanto no passo corretor temos a mesma matriz para calcular as direções, o que se altera é apenas o vetor do lado direito. Desse sistema calculamos as direções e atualizamos as variáveis seguindo o método a seguir.

Algoritmo Barreira Logarítmica Preditor-Corretor

O ponto inicial é calculado como em (3.6).

Assim, dados $x^0, y^0, (u^0, v^0) > 0, \mu^0, \beta > 1$ e $\tau = 0,99995$.

Para $k = 0, 1, 2, \dots$ faça:

Passo Preditor:

$$g^k = p(u^k + v^k)^{p-1}$$

$$U^k = \text{diag}(u^k)$$

$$V^k = \text{diag}(v^k)$$

$$G^k = \begin{cases} \frac{p(p-1)}{(u_i^k + v_i^k)^{2-p}}, & \text{se } i = j, \\ 0, & \text{se } i \neq j. \end{cases}$$

$$r_1^{\bar{k}} = -A^t y^k$$

$$r_2^{\bar{k}} = -Ax^k - u^k + v^k + b$$

$$r_3^{\bar{k}} = -U^k(g^k + y^k)$$

$$r_4^{\bar{k}} = -V^k(g^k - y^k)$$

$$D_u^k = [\text{diag}(g^k + y^k) + U^k G^k]$$

$$D_v^k = [\text{diag}(g^k - y^k) + V^k G^k]$$

$$D_\phi^k = [D_v^k - V^k U^k (G^k)^2 (D_u^k)^{-1}]$$

$$D^k = [-(D_u^k)^{-1} U^k - (D_\phi^k)^{-1} V^k (I + (D_u^k)^{-1} U^k G^k)^2]^{-1}$$

$$r^{\bar{k}} = -\{r_1^{\bar{k}} - A^t D^k [r_2^{\bar{k}} - (D_u^k)^{-1} r_3^{\bar{k}} + (D_\phi^k)^{-1} (I + (D_u^k)^{-1} U^k G^k) (r_4^{\bar{k}} - (D_u^k)^{-1} V^k G^k r_3^{\bar{k}})]\}$$

$$\begin{aligned}\bar{d}x^k &= (A^t D^k A)^{-1} \bar{r}^k \\ \bar{d}y^k &= D^k [r_2^k - A \bar{d}x^k - (D_u^k)^{-1} r_3^k + (D_\phi^k)^{-1} (I + (D_u^k)^{-1} U^k G^k) (r_4^k - (D_u^k)^{-1} V^k G^k r_3^k)] \\ \bar{d}v^k &= (D_\phi^k)^{-1} [r_4^k + V \bar{d}y^k + (D_u^k)^{-1} (-V^k G^k r_3^k + V^k U^k G \bar{d}y^k)] \\ \bar{d}u^k &= (D_u^k)^{-1} (r_3^k - U^k \bar{d}y^k - U^k G^k \bar{d}v^k)\end{aligned}$$

Passo Corretor:

$$\begin{aligned}\beta^k &= \min \left\{ \tau \left(\min_{(d_{u_i}^k + d_{v_i}^k) < 0} -\frac{(u_i^k + v_i^k)}{(d_{u_i}^k + d_{v_i}^k)} \right), 1 \right\} \\ \tilde{g}^k &= p(u^k + \beta^k \bar{d}u^k + v^k + \beta^k \bar{d}v^k)^{p-1} \\ d\bar{U}^k &= \text{diag}(\bar{d}u^k) \\ d\bar{V}^k &= \text{diag}(\bar{d}v^k) \\ R_1^k &= (U^k + d\bar{U}^k) \tilde{g}^k + d\bar{U}^k \bar{d}y^k - d\bar{U}^k g^k - U^k G^k \bar{d}u^k - U^k G^k \bar{d}v^k - U^k g^k \\ R_2^k &= (V^k + d\bar{V}^k) \tilde{g}^k - d\bar{V}^k \bar{d}y^k - V^k G^k \bar{d}u^k - d\bar{V}^k g^k - V^k G^k \bar{d}v^k - V^k g^k \\ \hat{r}_1^k &= -A^t y^k \\ \hat{r}_2^k &= -Ax^k - u^k + v^k + b \\ \hat{r}_3^k &= -U^k (g^k + y^k) + \mu^k e - R_1^k \\ \hat{r}_4^k &= -V^k (g^k - y^k) + \mu^k e - R_2^k \\ \hat{r}^k &= -\left\{ \hat{r}_1^k - A^t D^k \left[\hat{r}_2^k - (D_u^k)^{-1} \hat{r}_3^k + (D_\phi^k)^{-1} (I + (D_u^k)^{-1} U^k G^k) (\hat{r}_4^k - (D_u^k)^{-1} V^k G^k \hat{r}_3^k) \right] \right\} \\ d\hat{x}^k &= (A^t D^k A)^{-1} \hat{r}^k \\ d\hat{y}^k &= D^k [\hat{r}_2^k - A d\hat{x}^k - (D_u^k)^{-1} \hat{r}_3^k + (D_\phi^k)^{-1} (I + (D_u^k)^{-1} U^k G^k) (\hat{r}_4^k - (D_u^k)^{-1} V^k G^k \hat{r}_3^k)] \\ d\hat{v}^k &= (D_\phi^k)^{-1} [\hat{r}_4^k + V d\hat{y}^k + (D_u^k)^{-1} (-V^k G^k \hat{r}_3^k + V^k U^k G d\hat{y}^k)] \\ d\hat{u}^k &= (D_u^k)^{-1} (\hat{r}_3^k - U^k d\hat{y}^k - U^k G^k d\hat{v}^k) \\ \alpha^k &= \min \left\{ \tau \left(\min_{d\hat{u}_i^k < 0} -\frac{u_i^k}{d\hat{u}_i^k} \right), \tau \left(\min_{d\hat{v}_i^k < 0} -\frac{v_i^k}{d\hat{v}_i^k} \right), 1 \right\}\end{aligned}$$

$$\begin{aligned}
x^{k+1} &= x^k + \alpha^k \hat{d}x^k \\
y^{k+1} &= y^k + \alpha^k \hat{d}y^k \\
u^{k+1} &= u^k + \alpha^k \hat{d}u^k \\
v^{k+1} &= v^k + \alpha^k \hat{d}v^k \\
\mu^{k+1} &= \frac{\mu^k}{\beta}
\end{aligned}$$

Até convergir.

Critério de Convergência

Utilizamos o mesmo critério de convergência do algoritmo barreira logarítmica definido em (3.7) na Seção 3.5.1.

3.6.3 Método Primal-Dual Barreira Logarítmica

O sistema de equações (3.8) da Seção 3.5.2 é equivalente ao sistema:

$$\underbrace{\nabla L}_{J(x,y,u,v)} = \begin{bmatrix} A^t y \\ Ax + u - v - b \\ g - \mu U^{-1} e + y \\ g - \mu V^{-1} e - y \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}. \quad (3.11)$$

Fazendo $z_u = \mu U^{-1} e$ e $z_v = \mu V^{-1} e$ como em [13], e substituindo em (3.11), obtemos as condições de otimalidade:

$$\underbrace{\nabla L}_{J(x,y,u,v,z_u,z_v)} = \begin{bmatrix} A^t y \\ Ax + u - v - b \\ g - z_u + y \\ g - z_v - y \\ U Z_u e \\ V Z_v e \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \mu e \\ \mu e \end{bmatrix}, \quad (3.12)$$

onde:

$$Z_u = \text{diag}(z_u) = \begin{cases} z_{u_i}, & \text{se } i = j, \\ 0, & \text{se } i \neq j. \end{cases} \quad \text{e} \quad Z_v = \text{diag}(z_v) = \begin{cases} z_{v_i}, & \text{se } i = j, \\ 0, & \text{se } i \neq j. \end{cases}$$

Aplicando o método de Newton às condições de otimalidade, teremos o seguinte sistema:

$$\begin{bmatrix} 0 & A^t & 0 & 0 & 0 & 0 \\ A & 0 & I & -I & 0 & 0 \\ 0 & I & G & G & -I & 0 \\ 0 & -I & G & G & 0 & -I \\ 0 & 0 & Z_u & 0 & U & 0 \\ 0 & 0 & 0 & Z_v & 0 & V \end{bmatrix} \begin{bmatrix} dx \\ dy \\ du \\ dv \\ d_{z_u} \\ d_{z_v} \end{bmatrix} = \begin{bmatrix} r_1 \\ r_2 \\ r_3 \\ r_4 \\ r_5 \\ r_6 \end{bmatrix},$$

onde:

$$\begin{aligned}
 r_1 &= -A^t y, \\
 r_2 &= -Ax - u + v + b, \\
 r_3 &= -g + z_u - y, \\
 r_4 &= -g + z_v + y, \\
 r_5 &= -UZ_u e + \mu e, \\
 r_6 &= -VZ_v e + \mu e.
 \end{aligned}$$

Temos que resolver o sistema de equações abaixo para encontrarmos as direções.

$$\begin{aligned}
 A^t dy &= r_1 \\
 Adx + du - dv &= r_2 \\
 dy + Gdu + Gdv - dz_u &= r_3 \\
 -dy + Gdu + Gdv - dz_v &= r_4 \\
 Z_u du + Udz_u &= r_5 \\
 Z_v dv + Vdz_v &= r_6
 \end{aligned}$$

Resolvendo esse sistema encontramos as direções:

$$\begin{aligned}
 dx &= (A^t DA)^{-1} r, \\
 dy &= D \left[r_2 - Adx - Dz_u^{-1} (r_3 + U^{-1} r_5) + (Dz_v - G^2 Dz_u^{-1})^{-1} (I + Dz_u^{-1} G) E \right], \\
 dv &= (Dz_v - G^2 Dz_u^{-1})^{-1} [E + dy + Dz_u^{-1} G dy], \\
 du &= Dz_u^{-1} (r_3 - dy - Gdv + U^{-1} r_5), \\
 dz_v &= V^{-1} (r_6 - Z_v dv), \\
 dz_u &= U^{-1} (r_5 - Z_u du),
 \end{aligned}$$

onde:

$$\begin{aligned}
 D &= \left[-Dz_u^{-1} - (Dz_v - G^2 Dz_u^{-1})^{-1} (I + Dz_u^{-1}G)^2 \right]^{-1}, \\
 r &= - \left\{ r_1 - A^t D \left[r_2 - Dz_u^{-1} (r_3 + U^{-1}r_5) + (Dz_v - G^2 Dz_u^{-1})^{-1} (I + Dz_u^{-1}G) E \right] \right\}, \\
 E &= [r_4 + Dz_u^{-1} (-Gr_3 - GU^{-1}r_5) + V^{-1}r_6], \\
 Dz_u &= (G + U^{-1}Z_u), \\
 Dz_v &= (G + V^{-1}Z_v).
 \end{aligned}$$

Tendo calculadas as direções, devemos atualizar as variáveis usando o tamanho do passo α definido de forma a manter as variáveis u , v , dz_u e dz_v estritamente positivas.

$$\alpha = \min \left\{ \tau \left(\min_{du_i < 0} -\frac{u_i}{du_i} \right), \tau \left(\min_{dv_i < 0} -\frac{v_i}{dv_i} \right), \tau \left(\min_{dz_{ui} < 0} -\frac{z_{ui}}{dz_{ui}} \right), \tau \left(\min_{dz_{vi} < 0} -\frac{z_{vi}}{dz_{vi}} \right), 1 \right\},$$

onde $\tau = 0,99995$.

Devemos atualizar as variáveis seguindo o método descrito abaixo.

Algoritmo Primal-Dual Barreira Logarítmica

O ponto inicial é calculado como em (3.6).

$$z_u^0 = z_v^0 = e, \text{ onde } e = [1, 1, \dots, 1]^t.$$

Assim, dados $x^0, y^0, (u^0, v^0) > 0, \mu^0, (z_u^0, z_v^0) > 0$ e $\tau = 0,99995$.

Para $k = 0, 1, 2, \dots$ faça:

$$\begin{aligned}
 g^k &= p(u^k + v^k)^{p-1} \\
 G^k &= \begin{cases} \frac{p(p-1)}{(u_i^k + v_i^k)^{2-p}}, & \text{se } i = j, \\ 0, & \text{se } i \neq j. \end{cases} \\
 U^k &= \text{diag}(u^k) \\
 V^k &= \text{diag}(v^k) \\
 Zu^k &= \text{diag}(z_u^k) \\
 Zv^k &= \text{diag}(z_v^k) \\
 r_1^k &= -A^t y^k
 \end{aligned}$$

$$r_2^k = -Ax^k - u^k + v^k + b$$

$$r_3^k = -g^k + z_u^k - y^k$$

$$r_4^k = -g^k + z_v^k + y^k$$

$$r_5^k = -U^k Z u^k e + \mu^k e$$

$$r_6^k = -V^k Z v^k e + \mu^k e$$

$$Dz_u^k = (G^k + (U^k)^{-1} Z u^k)$$

$$Dz_v^k = (G^k + (V^k)^{-1} Z v^k)$$

$$F^k = (Dz_v^k - (G^k)^2 (Dz_u^k)^{-1})$$

$$D^k = \left[-(Dz_u^k)^{-1} - (F^k)^{-1} (I + (Dz_u^k)^{-1} G^k)^2 \right]^{-1}$$

$$E^k = [r_4^k + (Dz_u^k)^{-1} (-G^k r_3^k - G^k (U^k)^{-1} r_5^k) + (V^k)^{-1} r_6^k]$$

$$r^k = -\{r_1^k - A^t D^k [r_2^k - (Dz_u^k)^{-1} (r_3^k + (U^k)^{-1} r_5^k) + (F^k)^{-1} (I + (Dz_u^k)^{-1} G^k) E^k]\}$$

$$dx^k = (A^t D^k A)^{-1} r^k$$

$$dy^k = D^k [r_2^k - A dx^k - (Dz_u^k)^{-1} (r_3^k + (U^k)^{-1} r_5^k) + (F^k)^{-1} (I + (Dz_u^k)^{-1} G^k) E^k]$$

$$dv^k = (F^k)^{-1} [E^k + dy^k + (Dz_u^k)^{-1} G^k dy^k]$$

$$du^k = (Dz_u^k)^{-1} (r_3^k - dy^k - G^k dv^k + (U^k)^{-1} r_5^k)$$

$$dz_v^k = (V^k)^{-1} (r_6^k - Z v^k dv^k)$$

$$dz_u^k = (U^k)^{-1} (r_5^k - Z u^k du^k)$$

$$\alpha^k = \min \left\{ \tau \left(\min_{du_i^k < 0} -\frac{u_i^k}{du_i^k} \right), \tau \left(\min_{dv_i^k < 0} -\frac{v_i^k}{dv_i^k} \right), \tau \left(\min_{dz_{ui}^k < 0} -\frac{z_{ui}^k}{dz_{ui}^k} \right), \tau \left(\min_{dz_{vi}^k < 0} -\frac{z_{vi}^k}{dz_{vi}^k} \right), 1 \right\}$$

$$x^{k+1} = x^k + \alpha^k dx^k$$

$$y^{k+1} = y^k + \alpha^k dy^k$$

$$u^{k+1} = u^k + \alpha^k du^k$$

$$v^{k+1} = v^k + \alpha^k dv^k$$

$$z_u^{k+1} = z_u^k + \alpha^k dz_u^k$$

$$z_v^{k+1} = z_v^k + \alpha^k dz_v^k$$

$$\mu^{k+1} = \frac{\mu^k}{\beta}$$

Até convergir.

Critério de Convergência

Aqui também utilizamos dois critérios de convergência, um deles baseado nas condições de otimalidade e o outro baseado na diferença dos valores de N atual e da iteração anterior.

$$N^k = \frac{\|\nabla L^k\|}{(1 + \|x^k\| + \|u^k\| + \|v^k\| + \|y^k\| + \|z_u^k\| + \|z_v^k\|)(2m)} \leq \epsilon, \quad (3.13)$$

$$|N^{k+1} - N^k| \leq \epsilon_1.$$

A única diferença entre esse critério de convergência e o critério (3.7) da Seção 3.5.1 é que aqui aparecem as variáveis z_u e z_v , que são consideradas no cálculo de N^k .

Para saber mais sobre método de ponto interior primal-dual, ver [28].

3.6.4 Método Primal-Dual Barreira Logarítmica Preditor-Corretor

Assim como o método barreira logarítmica preditor-corretor, esse método também desenvolvido em [7] foi modificado nesse trabalho através do cálculo dos resíduos.

Como foi visto na Seção 3.5.3, o método de Newton aplicado ao sistema de equações (3.12) nos fornece o sistema:

$$\begin{bmatrix} 0 & A^t & 0 & 0 & 0 & 0 \\ A & 0 & I & -I & 0 & 0 \\ 0 & I & G & G & -I & 0 \\ 0 & -I & G & G & 0 & -I \\ 0 & 0 & Z_u & 0 & U & 0 \\ 0 & 0 & 0 & Z_v & 0 & V \end{bmatrix} \begin{bmatrix} dx \\ dy \\ du \\ dv \\ dz_u \\ dz_v \end{bmatrix} = \begin{bmatrix} r_1 \\ r_2 \\ r_3 \\ r_4 \\ r_5 \\ r_6 \end{bmatrix}, \quad (3.14)$$

onde:

$$\begin{aligned}
 r_1 &= -A^t y, \\
 r_2 &= -Ax - u + v + b, \\
 r_3 &= -g + z_u - y, \\
 r_4 &= -g + z_v + y, \\
 r_5 &= -UZ_u e + \mu e, \\
 r_6 &= -VZ_v e + \mu e.
 \end{aligned}$$

Passo preditor: Devemos fazer o parâmetro barreira $\mu = 0$ e calcularmos a direção resolvendo o sistema:

$$\begin{bmatrix} 0 & A^t & 0 & 0 & 0 & 0 \\ A & 0 & I & -I & 0 & 0 \\ 0 & I & G & G & -I & 0 \\ 0 & -I & G & G & 0 & -I \\ 0 & 0 & Z_u & 0 & U & 0 \\ 0 & 0 & 0 & Z_v & 0 & V \end{bmatrix} \begin{bmatrix} \bar{d}x \\ \bar{d}y \\ \bar{d}u \\ \bar{d}v \\ \bar{d}_{z_u} \\ \bar{d}_{z_v} \end{bmatrix} = \begin{bmatrix} \bar{r}_1 \\ \bar{r}_2 \\ \bar{r}_3 \\ \bar{r}_4 \\ \bar{r}_5 \\ \bar{r}_6 \end{bmatrix},$$

onde:

$$\begin{aligned}
 \bar{r}_1 &= -A^t y, \\
 \bar{r}_2 &= -Ax - u + v + b, \\
 \bar{r}_3 &= -g + z_u - y, \\
 \bar{r}_4 &= -g + z_v + y, \\
 \bar{r}_5 &= -UZ_u e, \\
 \bar{r}_6 &= -VZ_v e.
 \end{aligned}$$

Obtemos as direções do passo preditor:

$$\begin{aligned}
 \bar{d}x &= (A^t DA)^{-1} \bar{r}, \\
 \bar{d}y &= D \left[\bar{r}_2 - A\bar{d}x - Dz_u^{-1} (\bar{r}_3 + U^{-1}\bar{r}_5) + (Dz_v - G^2 Dz_u^{-1})^{-1} (I + Dz_u^{-1}G) \bar{E} \right], \\
 \bar{d}v &= (Dz_v - G^2 Dz_u^{-1})^{-1} [\bar{E} + \bar{d}y + Dz_u^{-1}G\bar{d}y], \\
 \bar{d}u &= Dz_u^{-1} (\bar{r}_3 - \bar{d}y - G\bar{d}v + U^{-1}\bar{r}_5), \\
 \bar{d}z_v &= V^{-1} (\bar{r}_6 - Z_v\bar{d}v), \\
 \bar{d}z_u &= U^{-1} (\bar{r}_5 - Z_u\bar{d}u),
 \end{aligned}$$

onde:

$$\begin{aligned}
 D &= \left[-Dz_u^{-1} - (Dz_v - G^2 Dz_u^{-1})^{-1} (I + Dz_u^{-1}G)^2 \right]^{-1}, \\
 \bar{r} &= - \left\{ \bar{r}_1 - A^t D \left[\bar{r}_2 - Dz_u^{-1} (\bar{r}_3 + U^{-1}\bar{r}_5) + (Dz_v - G^2 Dz_u^{-1})^{-1} (I + Dz_u^{-1}G) \bar{E} \right] \right\}, \\
 \bar{E} &= [\bar{r}_4 + Dz_u^{-1} (-G\bar{r}_3 - GU^{-1}\bar{r}_5) + V^{-1}\bar{r}_6], \\
 Dz_u &= (G + U^{-1}Z_u), \\
 Dz_v &= (G + V^{-1}Z_v).
 \end{aligned}$$

Para calcular os resíduos devemos comparar os sistemas de equações (3.12), substituindo x por $x + dx$, y por $y + dy$, u por $u + du$ e v por $v + dv$, z_u por $z_u + dz_u$ e z_v por $z_v + dz_v$ com o sistema de equações (3.14). As diferenças encontradas serão nossos resíduos.

Primeiro sistema de equações do sistema (3.12):

$$\begin{aligned}
 A^t(y + dy) &= 0 \\
 \Rightarrow A^t dy &= -A^t y.
 \end{aligned}$$

Comparando com o sistema (3.14), vemos que não existe resíduo.

Segundo sistema de equações do sistema (3.12):

$$\begin{aligned}
 A(x + dx) + u + du - v - dv - b &= 0 \\
 \Rightarrow Adx + du - dv &= b - Ax - u + v.
 \end{aligned}$$

Comparando com o sistema (3.14), vemos que nesse sistema também não existe resí-

duo.

Terceiro sistema de equações do sistema (3.12):

$$\tilde{g} - (z_u + dz_u) + y + dy = 0,$$

onde $\tilde{g}_i = p(u_i + du_i + v_i + dv_i)^{p-1}$, assim $\tilde{g} = \tilde{g}_i$, $i = 1, 2, \dots, m$. Entretanto, $p(u_i + du_i + v_i + dv_i)^{p-1}$ pode resultar em um número imaginário, para corrigir esse problema, devemos fazer $\tilde{g}_i = p(u_i + \beta du_i + v_i + \beta dv_i)^{p-1}$, e β é calculado como em (3.10).

Nesse sistema de equações temos o seguinte resíduo:

$$\begin{aligned} R_1 &= (\tilde{g} - z_u - dz_u + y + dy) - (dy + Gdu + Gdv - dz_u + g - z_u + y) \\ \Rightarrow R_1 &= \tilde{g} - z_u - dz_u + y + dy - dy - Gdu - Gdv + dz_u - g + z_u - y \\ \Rightarrow R_1 &= \tilde{g} - G(du + dv) - g. \end{aligned}$$

Quarto sistema de equações do sistema (3.12):

$$\tilde{g} - (z_v + dz_v) - (y + dy) = 0,$$

onde $\tilde{g}_i = p(u_i + \beta du_i + v_i + \beta dv_i)^{p-1}$, e β é calculado como em (3.10).

Nesse sistema de equações temos o seguinte resíduo:

$$\begin{aligned} R_2 &= (\tilde{g} - z_v - dz_v - y - dy) - (-dy + Gdu + Gdv - dz_v + g - z_v - y) \\ \Rightarrow R_2 &= \tilde{g} - z_v - dz_v - y - dy + dy - Gdu - Gdv + dz_v - g + z_v + y \\ \Rightarrow R_2 &= \tilde{g} - G(du + dv) - g = R_1. \end{aligned}$$

Quinto sistema de equações do sistema (3.12):

$$(U + dU)(Z_u + dZ_u)e = \mu e,$$

onde $U = \text{diag}(u)$, $dU = \text{diag}(du)$, $Z_u = \text{diag}(z_u)$, $dZ_u = \text{diag}(dz_u)$ e $e = [1, 1, \dots, 1]^t$.

Nesse sistema de equações temos o seguinte resíduo:

$$\begin{aligned} R_3 &= (UZ_u e + UdZ_u e + dUZ_u e + dUdZ_u e - \mu e) - (Z_u du + Udz_u + UZ_u e - \mu e) \\ \Rightarrow R_3 &= UZ_u e + UdZ_u e + dUZ_u e + dUdZ_u e - \mu e - Z_u du - Udz_u - UZ_u e + \mu e \end{aligned}$$

$$\Rightarrow R_3 = dU dZ_u e = dU dz_u.$$

Sexto sistema de equações do sistema (3.12):

$$(V + dV)(Z_v + dZ_v)e = \mu e,$$

onde $V = \text{diag}(v)$, $dV = \text{diag}(dv)$, $Z_v = \text{diag}(z_v)$, $dZ_v = \text{diag}(dz_v)$ e $e = [1, 1, \dots, 1]^t$.

Nesse sistema de equações temos o seguinte resíduo:

$$R_4 = (V Z_v e + V dZ_v e + dV Z_v e + dV dZ_v e - \mu e) - (Z_v dv + V dz_v + V Z_v e - \mu e)$$

$$\Rightarrow R_4 = V Z_v e + V dZ_v e + dV Z_v e + dV dZ_v e - \mu e - Z_v dv - V dz_v - V Z_v e + \mu e$$

$$\Rightarrow R_4 = dV dZ_v e = dV dz_v.$$

Passo corretor: Agora podemos impor um valor para o parâmetro barreira μ , como em [18], e resolver o sistema:

$$\begin{bmatrix} 0 & A^t & 0 & 0 & 0 & 0 \\ A & 0 & I & -I & 0 & 0 \\ 0 & I & G & G & -I & 0 \\ 0 & -I & G & G & 0 & -I \\ 0 & 0 & Z_u & 0 & U & 0 \\ 0 & 0 & 0 & Z_v & 0 & V \end{bmatrix} \begin{bmatrix} \hat{d}x \\ \hat{d}y \\ \hat{d}u \\ \hat{d}v \\ \hat{d}z_u \\ \hat{d}z_v \end{bmatrix} = \begin{bmatrix} \hat{r}_1 \\ \hat{r}_2 \\ \hat{r}_3 \\ \hat{r}_4 \\ \hat{r}_5 \\ \hat{r}_6 \end{bmatrix},$$

onde:

$$\begin{aligned}
\hat{r}_1 &= -A^t y, \\
\hat{r}_2 &= -Ax - u + v + b, \\
\hat{r}_3 &= -g + z_u - y - R_1, \\
\hat{r}_4 &= -g + z_v + y - R_2, \\
\hat{r}_5 &= -UZ_u e + \mu e - R_3, \\
\hat{r}_6 &= -VZ_v e + \mu e - R_4, \\
R_1 &= \tilde{g} - G(du + dv) - g, \\
R_2 &= \tilde{g} - G(du + dv) - g, \\
R_3 &= dUdz_u, \\
R_4 &= dVdz_v, \\
\tilde{g} &= p(u + \beta du + v + \beta dv)^{p-1}, \\
\beta &= \min \left\{ \tau \left(\min_{(d_{u_i} + d_{v_i}) < 0} -\frac{(u_i + v_i)}{(d_{u_i} + d_{v_i})} \right), 1 \right\}, \text{ onde } \tau = 0.9995.
\end{aligned}$$

Obtemos as direções do passo corretor:

$$\begin{aligned}
\hat{d}x &= (A^t DA)^{-1} \hat{r}, \\
\hat{d}y &= D \left[\hat{r}_2 - A\hat{d}x - Dz_u^{-1} (\hat{r}_3 + U^{-1}\hat{r}_5) + (Dz_v - G^2 Dz_u^{-1})^{-1} (I + Dz_u^{-1}G) \hat{E} \right], \\
\hat{d}v &= (Dz_v - G^2 Dz_u^{-1})^{-1} \left[\hat{E} + \hat{d}y + Dz_u^{-1}G\hat{d}y \right], \\
\hat{d}u &= Dz_u^{-1} \left(\hat{r}_3 - \hat{d}y - G\hat{d}v + U^{-1}\hat{r}_5 \right), \\
\hat{d}z_v &= V^{-1} \left(\hat{r}_6 - Z_v \hat{d}v \right), \\
\hat{d}z_u &= U^{-1} \left(\hat{r}_5 - Z_u \hat{d}u \right),
\end{aligned}$$

onde:

$$\begin{aligned}
 D &= \left[-Dz_u^{-1} - (Dz_v - G^2 Dz_u^{-1})^{-1} (I + Dz_u^{-1}G)^2 \right]^{-1}, \\
 \hat{r} &= - \left\{ \hat{r}_1 - A^t D \left[\hat{r}_2 - Dz_u^{-1} (\hat{r}_3 + U^{-1} \hat{r}_5) + (Dz_v - G^2 Dz_u^{-1})^{-1} (I + Dz_u^{-1}G) \hat{E} \right] \right\}, \\
 \hat{E} &= \left[\hat{r}_4 + Dz_u^{-1} (-G \hat{r}_3 - GU^{-1} \hat{r}_5) + V^{-1} \hat{r}_6 \right], \\
 Dz_u &= (G + U^{-1} Z_u), \\
 Dz_v &= (G + V^{-1} Z_v).
 \end{aligned}$$

Calculadas as direções, devemos atualizar as variáveis usando o tamanho do passo α definido de forma a manter as variáveis u , v , dz_u e dz_v estritamente positivas.

$$\alpha = \min \left\{ \tau \left(\min_{du_i < 0} -\frac{u_i}{du_i} \right), \tau \left(\min_{dv_i < 0} -\frac{v_i}{dv_i} \right), \tau \left(\min_{dz_{ui} < 0} -\frac{z_{ui}}{dz_{ui}} \right), \tau \left(\min_{dz_{vi} < 0} -\frac{z_{vi}}{dz_{vi}} \right), 1 \right\},$$

onde $\tau = 0,99995$.

Devemos atualizar as variáveis seguindo o método descrito abaixo.

Algoritmo Primal-Dual Barreira Logarítmica Preditor-Corretor

O ponto inicial é calculado como no método primal-dual barreira logarítmica na Seção 3.5.3.

Assim, dados $x^0, y^0, (u^0, v^0) > 0, \mu^0, (z_u^0, z_v^0) > 0$ e $\tau = 0,99995$.

Para $k = 0, 1, 2, \dots$ faça:

Passo Preditor:

$$\begin{aligned}
 g^k &= p(u^k + v^k)^{p-1} \\
 G^k &= \begin{cases} \frac{p(p-1)}{(u_i^k + v_i^k)^{2-p}}, & \text{se } i = j, \\ 0, & \text{se } i \neq j. \end{cases} \\
 U^k &= \text{diag}(u^k) \\
 V^k &= \text{diag}(v^k)
 \end{aligned}$$

$$Zu^k = \text{diag}(z_u^k)$$

$$Zv^k = \text{diag}(z_v^k)$$

$$\bar{r}_1^k = -A^t y^k$$

$$\bar{r}_2^k = -Ax^k - u^k + v^k + b$$

$$\bar{r}_3^k = -g^k + z_u^k - y^k$$

$$\bar{r}_4^k = -g^k + z_v^k + y^k$$

$$\bar{r}_5^k = -U^k Zu^k e$$

$$\bar{r}_6^k = -V^k Zv^k e$$

$$Dz_u^k = (G^k + (U^k)^{-1}Z_u^k)$$

$$Dz_v^k = (G^k + (V^k)^{-1}Z_v^k)$$

$$D^k = \left[-(Dz_u^k)^{-1} - (Dz_v^k - (G^k)^2(Dz_u^k)^{-1})^{-1} (I + (Dz_u^k)^{-1}G^k)^2 \right]^{-1}$$

$$\bar{E}^k = [\bar{r}_4^k + (Dz_u^k)^{-1} (-G^k \bar{r}_3^k - G^k (U^k)^{-1} \bar{r}_5^k) + (V^k)^{-1} \bar{r}_6^k]$$

$$\bar{r}^k = - \left\{ \bar{r}_1^k - A^t D^k \left[\bar{r}_2^k - (Dz_u^k)^{-1} (\bar{r}_3^k + (U^k)^{-1} \bar{r}_5^k) + (Dz_v^k - (G^k)^2(Dz_u^k)^{-1})^{-1} (I + Dz_u^{-1}G) \bar{E}^k \right] \right\}$$

$$\bar{d}x^k = (A^t D^k A)^{-1} \bar{r}^k$$

$$\bar{d}y^k = D^k \left[\bar{r}_2^k - A \bar{d}x^k - (Dz_u^k)^{-1} (\bar{r}_3^k + (U^k)^{-1} \bar{r}_5^k) + (Dz_v^k - (G^k)^2(Dz_u^k)^{-1})^{-1} (I + (Dz_u^k)^{-1}G^k) \bar{E}^k \right]$$

$$\bar{d}v^k = (Dz_v^k - (G^k)^2(Dz_u^k)^{-1})^{-1} [\bar{E}^k + \bar{d}y^k + (Dz_u^k)^{-1}G^k \bar{d}y^k]$$

$$\bar{d}u^k = (Dz_u^k)^{-1} (\bar{r}_3^k - \bar{d}y^k - G^k \bar{d}v^k + (U^k)^{-1} \bar{r}_5^k)$$

$$\bar{d}z_v^k = (V^k)^{-1} (\bar{r}_6^k - Z_v^k \bar{d}v^k)$$

$$\bar{d}z_u^k = (U^k)^{-1} (\bar{r}_5^k - Z_u^k \bar{d}u^k)$$

Passo Corretor:

$$\beta^k = \min \left\{ \tau \left(\min_{(\bar{d}_{u_i}^k + \bar{d}_{v_i}^k) < 0} - \frac{(u_i^k + v_i^k)}{(\bar{d}_{u_i}^k + \bar{d}_{v_i}^k)} \right), 1 \right\}$$

$$\tilde{g}^k = p(u^k + \beta^k \bar{d}u^k + v^k + \beta^k \bar{d}v^k)^{p-1}$$

$$\bar{d}U^k = \text{diag}(\bar{d}u^k)$$

$$\bar{d}V^k = \text{diag}(\bar{d}v^k)$$

$$R^k = \tilde{g}^k - G^k(\bar{d}u^k + \bar{d}v^k) - g^k$$

$$R_1^k = \bar{d}U^k \bar{d}z_u^k$$

$$R_2^k = \bar{d}V^k \bar{d}z_v^k$$

$$\hat{r}_1^k = -A^t y^k$$

$$\hat{r}_2^k = -Ax^k - u^k + v^k + b$$

$$\hat{r}_3^k = -g^k + z_u^k - y^k - R^k$$

$$\hat{r}_4^k = -g^k + z_v^k + y^k - R^k$$

$$\hat{r}_5^k = -U^k Z_u^k e + \mu^k e - R_1^k$$

$$\hat{r}_6^k = -V^k Z_v^k e + \mu^k e - R_2^k$$

$$\hat{E}^k = [\hat{r}_4^k + (Dz_u^k)^{-1} (-G^k \hat{r}_3^k - G^k (U^k)^{-1} \hat{r}_5^k) + (V^k)^{-1} \hat{r}_6^k]$$

$$\hat{r}^k = - \left\{ \hat{r}_1^k - A^t D^k \left[\hat{r}_2^k - (Dz_u^k)^{-1} (\hat{r}_3^k + (U^k)^{-1} \hat{r}_5^k) + (Dz_v^k - (G^k)^2 (Dz_u^k)^{-1})^{-1} (I + (Dz_u^k)^{-1} G^k) \hat{E}^k \right] \right\}$$

$$\hat{d}x^k = (A^t D^k A)^{-1} \hat{r}^k$$

$$\hat{d}y^k = D^k \left[\hat{r}_2^k - A \hat{d}x^k - (Dz_u^k)^{-1} (\hat{r}_3^k + (U^k)^{-1} \hat{r}_5^k) + (Dz_v^k - (G^k)^2 (Dz_u^k)^{-1})^{-1} (I + (Dz_u^k)^{-1} G^k) \hat{E}^k \right]$$

$$\hat{d}v^k = (Dz_v^k - (G^k)^2 (Dz_u^k)^{-1})^{-1} \left[\hat{E}^k + \hat{d}y^k + (Dz_u^k)^{-1} G^k \hat{d}y^k \right]$$

$$\hat{d}u^k = (Dz_u^k)^{-1} \left(\hat{r}_3^k - \hat{d}y^k - G^k \hat{d}v^k + (U^k)^{-1} \hat{r}_5^k \right)$$

$$\hat{d}z_v^k = (V^k)^{-1} \left(\hat{r}_6^k - Z_v^k \hat{d}v^k \right)$$

$$\hat{d}z_u^k = (U^k)^{-1} \left(\hat{r}_5^k - Z_u^k \hat{d}u^k \right)$$

$$\begin{aligned}
\alpha^k &= \min \left\{ \tau \left(\min_{d\hat{u}_i^k < 0} -\frac{u_i^k}{d\hat{u}_i^k} \right), \tau \left(\min_{d\hat{v}_i^k < 0} -\frac{v_i^k}{d\hat{v}_i^k} \right), \tau \left(\min_{d\hat{z}_{u_i}^k < 0} -\frac{z_{u_i}^k}{d\hat{z}_{u_i}^k} \right), \tau \left(\min_{d\hat{z}_{v_i}^k < 0} -\frac{z_{v_i}^k}{d\hat{z}_{v_i}^k} \right), 1 \right\} \\
x^{k+1} &= x^k + \alpha^k \hat{d}x^k \\
y^{k+1} &= y^k + \alpha^k \hat{d}y^k \\
u^{k+1} &= u^k + \alpha^k \hat{d}u^k \\
v^{k+1} &= v^k + \alpha^k \hat{d}v^k \\
z_u^{k+1} &= z_u^k + \alpha^k \hat{d}_{z_u}^k \\
z_v^{k+1} &= z_v^k + \alpha^k \hat{d}_{z_v}^k \\
\mu^{k+1} &= \frac{\mu^k}{\beta}
\end{aligned}$$

Até convergir.

Critério de Convergência

Utilizamos o mesmo critério de convergência do algoritmo primal-dual barreira logarítmica definido em (3.13) na Seção 3.5.3.

Capítulo 4

Modificação nos Métodos de Pontos Interiores Aplicados ao Problema de Regressão pela Norma L_p

Apresentaremos agora algumas modificações nos métodos de pontos interiores descritos no Capítulo 3 para resolver o problema de regressão pela norma L_p . Em todos eles modificamos a forma de calcular as direções no intuito de reduzir o número de operações necessárias por iteração. Modificamos também os pontos iniciais. As direções finais obtidas serão equivalentes às descritas pelos métodos do Capítulo 3.

Apresentaremos apenas as modificações e usaremos as informações do capítulo anterior.

4.1 Método Barreira Logarítmica

- Cálculo de dv :

$$dv = [D_v - VUG^2 D_u^{-1}]^{-1} \{r_4 + Vdy + D_u^{-1} (-VGr_3 + VUGdy)\}.$$

Multiplicando ambos os lados dessa igualdade por $[D_v - VUG^2 D_u^{-1}]$:

$$[D_v - VUG^2 D_u^{-1}] dv = r_4 + Vdy + D_u^{-1} (-VGr_3 + VUGdy).$$

Pré-multiplicando por D_u :

$$(D_u D_v - VUG^2) dv = D_u r_4 + D_u Vdy - VGr_3 + VUGdy.$$

Substituindo D_v por $[diag(g - y) + VG]$:

$$[D_u (diag(g - y) + VG) - VUG^2] dv = D_u r_4 + D_u Vdy - VGr_3 + VUGdy$$

$$\Rightarrow [D_u diag(g - y) + D_u VG - VUG^2] dv = D_u r_4 + D_u Vdy - VGr_3 + VUGdy.$$

Substituindo D_u por $[diag(g + y) + UG]$ apenas no segundo membro do lado esquerdo da igualdade:

$$[D_u diag(g - y) + (diag(g + y) + UG) VG - VUG^2] dv = D_u r_4 + D_u Vdy - VGr_3 + VUGdy$$

$$\Rightarrow [D_u diag(g - y) + diag(g + y)VG] dv = D_u r_4 + D_u Vdy - VGr_3 + VUGdy.$$

Temos então que:

$$dv = D_1^{-1} [D_u (r_4 + Vdy) - VGr_3 + VUGdy],$$

onde $D_1 = [D_u diag(g - y) + diag(g + y)VG]$.

- Cálculo de dy :

$$dy = D \left\{ r_2 - Adx - D_u^{-1} r_3 + [D_v - VUG^2 D_u^{-1}]^{-1} (I + D_u^{-1} UG) (r_4 - D_u^{-1} VGr_3) \right\},$$

$$\text{onde } D = \left\{ -D_u^{-1} U - [D_v - VUG^2 D_u^{-1}]^{-1} V (I + D_u^{-1} UG)^2 \right\}^{-1}.$$

$$\Rightarrow D^{-1} dy = \left\{ r_2 - Adx - D_u^{-1} r_3 + [D_v - VUG^2 D_u^{-1}]^{-1} (I + D_u^{-1} UG) (r_4 - D_u^{-1} VGr_3) \right\}$$

$$\begin{aligned} &\Rightarrow \left[-D_u^{-1}U - [D_v - VUG^2D_u^{-1}]^{-1}V (I + D_u^{-1}UG)^2 \right] dy = \\ &= \left\{ r_2 - Adx - D_u^{-1}r_3 + [D_v - VUG^2D_u^{-1}]^{-1} (I + D_u^{-1}UG) (r_4 - D_u^{-1}VGr_3) \right\}. \end{aligned}$$

Multiplicando ambos os membros dessa igualdade por $[D_v - VUG^2D_u^{-1}]$:

$$\begin{aligned} &\left[(D_v - VUG^2D_u^{-1}) (-D_u^{-1}U) - V (I + D_u^{-1}UG)^2 \right] dy = \\ &= (D_v - VUG^2D_u^{-1}) (r_2 - Adx - D_u^{-1}r_3) + (I + D_u^{-1}UG) (r_4 - D_u^{-1}VGr_3). \end{aligned}$$

Observe que $(I + D_u^{-1}UG)^2 = I + 2D_u^{-1}UG + D_u^{-2}U^2G^2$, pois estamos trabalhando com matrizes diagonais (U, V, G, D_u e D_v) e por isso podemos comutar as matrizes, o que simplifica os cálculos. Assim,

$$\begin{aligned} &\left[-D_vD_u^{-1}U + D_u^{-2}VU^2G^2 - V - 2VD_u^{-1}UG - D_u^{-2}VU^2G^2 \right] dy = (D_v - VUG^2D_u^{-1}) \\ &(r_2 - Adx) + (I + D_u^{-1}UG) r_4 - D_vD_u^{-1}r_3 + D_u^{-2}VUG^2r_3 - D_u^{-1}VGr_3 - D_u^{-2}VUG^2r_3 \\ &\Rightarrow \left[-D_vD_u^{-1}U - V - 2VD_u^{-1}UG \right] dy = (D_v - VUG^2D_u^{-1}) (r_2 - Adx) + (I + D_u^{-1}UG) r_4 \\ &- D_vD_u^{-1}r_3 - D_u^{-1}VGr_3. \end{aligned}$$

Multiplicando por D_u :

$$\left[-D_vU - D_uV - 2VUG \right] dy = (D_uD_v - VUG^2) (r_2 - Adx) + (D_u + UG) r_4 - (D_v + VG) r_3.$$

Observe que $(D_uD_v - VUG^2) = D_1$, então:

$$\begin{aligned} &\left[-V (D_u + UG) - U (D_v + VG) \right] dy = D_1 (r_2 - Adx) + (D_u + UG) r_4 - (D_v + VG) r_3 \\ &\Rightarrow \left[V (D_u + UG) + U (D_v + VG) \right] dy = D_1 (Adx - r_2) - (D_u + UG) r_4 + (D_v + VG) r_3. \end{aligned}$$

Temos então:

$$dy = D_2^{-1} \{ D_1 (Adx - r_2) - (D_u + UG) r_4 + (D_v + VG) r_3 \},$$

onde $D_2 = [V (D_u + UG) + U (D_v + VG)]$.

- Cálculo de dx :

$$A^T dy = r_1$$

$$\Rightarrow A^t D_2^{-1} \{D_1 (Adx - r_2) - (D_u + UG) r_4 + (D_v + VG) r_3\} = r_1$$

$$\Rightarrow A^t D_2^{-1} D_1 Adx - A^t D_2^{-1} \{D_1 r_2 + (D_u + UG) r_4 - (D_v + VG) r_3\} = r_1$$

$$\Rightarrow A^t D_2^{-1} D_1 Adx = r_1 + A^t D_2^{-1} \{D_1 r_2 + (D_u + UG) r_4 - (D_v + VG) r_3\}.$$

Temos então:

$$dx = (A^t D_2^{-1} D_1 A)^{-1} r,$$

onde $r = r_1 + A^t D_2^{-1} \{D_1 r_2 + (D_u + UG) r_4 - (D_v + VG) r_3\}$.

Com essas modificações teremos $5m$ operações a menos no cálculo de dx e m operações a menos no cálculo de dy em relação à versão anterior, por iteração.

Pode-se verificar em [14] um detalhamento sobre número de operações bem como a ordem de operações obtidas em um determinado cálculo.

A modificação no ponto inicial será feita somente nos vetores u e v :

$$u_i = \begin{cases} r_i + \sigma, & \text{se } r_i \geq 0, \\ \sigma, & \text{caso contrário.} \end{cases}$$

$$v_i = \begin{cases} \sigma, & \text{se } r_i \geq 0, \\ \sigma - r_i, & \text{caso contrário.} \end{cases}$$

onde $0 < \sigma \leq 1$.

Como foi visto no Capítulo 3, Seção 3.1, na solução ótima os problemas (3.1) e (3.3) serão equivalentes se tivermos $u_i v_i = 0$, para $i = 1, 2, \dots, m$. Com a modificação acima em u e v iniciais partimos de um r_i inicial que aproxima $u_i v_i$ de zero para todo $i = 1, 2, \dots, m$.

Algoritmo Barreira Logarítmica

Ponto inicial:

$$\begin{aligned}
 x^0 &= (A^t A)^{-1} A^t b, \\
 r^0 &= b - Ax^0, \\
 u_i^0 &= \begin{cases} r_i^0 + \sigma, & \text{se } r_i^0 \geq 0, \\ \sigma, & \text{caso contrário.} \end{cases} \\
 v_i^0 &= \begin{cases} \sigma, & \text{se } r_i^0 \geq 0, \\ \sigma - r_i^0, & \text{caso contrário.} \end{cases} \\
 y^0 &= \frac{\delta r^0}{\|r^0\|_\infty}, \text{ com } \delta = 0.975.
 \end{aligned} \tag{4.1}$$

Assim, dados $x^0, y^0, (u^0, v^0) > 0, \mu^0, \beta > 1, 0 < \sigma \leq 1$ e $\tau = 0,99995$.

Para $k = 0, 1, 2, \dots$ faça:

$$\begin{aligned}
 g^k &= p(u^k + v^k)^{p-1} \\
 G^k &= \begin{cases} \frac{p(p-1)}{(u_i^k + v_i^k)^{2-p}}, & \text{se } i = j, \\ 0, & \text{se } i \neq j. \end{cases} \\
 U^k &= \text{diag}(u^k) \\
 V^k &= \text{diag}(v^k) \\
 r_1^k &= -A^t y^k \\
 r_2^k &= -Ax^k - u^k + v^k + b \\
 r_3^k &= -U^k(g^k + y^k) + \mu^k e \\
 r_4^k &= -V^k(g^k - y^k) + \mu^k e \\
 D_u^k &= [\text{diag}(g^k + y^k) + U^k G^k] \\
 D_v^k &= [\text{diag}(g^k - y^k) + V^k G^k] \\
 D_1^k &= [D_u^k \text{diag}(g^k - y^k) + \text{diag}(g^k + y^k) V^k G^k] \\
 D_2^k &= [V^k (D_u^k + U^k G^k) + U^k (D_v^k + V^k G^k)]
 \end{aligned}$$

$$\begin{aligned}
r^k &= r_1^k + A^t(D_2^k)^{-1} \{D_1^k r_2^k + (D_u^k + U^k G^k) r_4^k - (D_v^k + V^k G^k) r_3^k\} \\
dx^k &= (A^t(D_2^k)^{-1} D_1^k A)^{-1} r^k \\
dy^k &= (D_2^k)^{-1} \{D_1^k (A dx^k - r_2^k) - (D_u^k + U^k G^k) r_4^k + (D_v^k + V^k G^k) r_3^k\} \\
dv^k &= (D_1^k)^{-1} [D_u^k (r_4^k + V^k dy^k) - V^k G^k r_3^k + V^k U^k G^k dy^k] \\
du^k &= (D_u^k)^{-1} (r_3^k - U^k dy^k - U^k G^k dv^k) \\
\alpha^k &= \min \left\{ \tau \left(\min_{du_i^k < 0} -\frac{u_i^k}{du_i^k} \right), \tau \left(\min_{dv_i^k < 0} -\frac{v_i^k}{dv_i^k} \right), 1 \right\} \\
x^{k+1} &= x^k + \alpha^k dx^k \\
y^{k+1} &= y^k + \alpha^k dy^k \\
u^{k+1} &= u^k + \alpha^k du^k \\
v^{k+1} &= v^k + \alpha^k dv^k \\
\mu^{k+1} &= \frac{\mu^k}{\beta}
\end{aligned}$$

Até convergir.

Critério de Convergência: O critério de convergência é o mesmo do método barreira logarítmica definido em (3.7) na Seção 3.5.1 do Capítulo 3.

4.2 Método Barreira Logarítmica Preditor-Corretor

As modificações nas direções desse método são análogas às do método barreira logarítmica.

Algoritmo Barreira Logarítmica Preditor-Corretor

O ponto inicial é calculado como em (4.1).

Assim, dados $x^0, y^0, (u^0, v^0) > 0, \mu^0, \beta > 1, 0 < \sigma \leq 1$, e $\tau = 0,99995$.

Para $k = 0, 1, 2, \dots$ faça:

Passo Preditor:

$$\begin{aligned}
 g^k &= p(u^k + v^k)^{p-1} \\
 g^k &= g_i^k, i = 1, \dots, m. \\
 G^k &= \begin{cases} \frac{p(p-1)}{(u_i^k + v_i^k)^{2-p}} & \text{se } i = j, \\ 0 & \text{se } i \neq j. \end{cases} \\
 U^k &= \text{diag}(u^k) \\
 V^k &= \text{diag}(v^k) \\
 \bar{r}_1^k &= -A^t y^k \\
 \bar{r}_2^k &= -Ax^k - u^k + v^k + b \\
 \bar{r}_3^k &= -U^k(g^k + y^k) \\
 \bar{r}_4^k &= -V^k(g^k - y^k) \\
 D_u^k &= [\text{diag}(g^k + y^k) + U^k G^k] \\
 D_v^k &= [\text{diag}(g^k - y^k) + V^k G^k] \\
 D_1^k &= [D_u^k \text{diag}(g^k - y^k) + \text{diag}(g^k + y^k) V^k G^k] \\
 D_2^k &= [V^k (D_u^k + U^k G^k) + U^k (D_v^k + V^k G^k)] \\
 \bar{r}^k &= \bar{r}_1^k + A^t (D_2^k)^{-1} \{ D_1^k \bar{r}_2^k + (D_u^k + U^k G^k) \bar{r}_4^k - (D_v^k + V^k G^k) \bar{r}_3^k \} \\
 \bar{d}x^k &= (A^t (D_2^k)^{-1} D_1^k A)^{-1} \bar{r}^k \\
 \bar{d}y^k &= (D_2^k)^{-1} \{ D_1^k (A \bar{d}x^k - \bar{r}_2^k) - (D_u^k + U^k G^k) \bar{r}_4^k + (D_v^k + V^k G^k) \bar{r}_3^k \} \\
 \bar{d}v^k &= (D_1^k)^{-1} [D_u^k (\bar{r}_4^k + V^k \bar{d}y^k) - V^k G^k \bar{r}_3^k + V^k U^k G^k \bar{d}y^k] \\
 \bar{d}u^k &= (D_u^k)^{-1} (\bar{r}_3^k - U^k \bar{d}y^k - U^k G^k \bar{d}v^k)
 \end{aligned}$$

Passo Corretor:

$$\begin{aligned} \beta^k &= \min \left\{ \tau \left(\min_{(d_{u_i}^k + d_{v_i}^k) < 0} -\frac{(u_i^k + v_i^k)}{(d_{u_i}^k + d_{v_i}^k)} \right), 1 \right\} \\ \tilde{g}^k &= p(u^k + \beta^k d\bar{u}^k + v^k + \beta^k d\bar{v}^k)^{p-1} \\ d\bar{U}^k &= \text{diag}(d\bar{u}^k) \\ d\bar{V}^k &= \text{diag}(d\bar{v}^k) \\ R_1^k &= (U^k + d\bar{U}^k)\tilde{g}^k + d\bar{U}^k d\bar{y}^k - d\bar{U}^k g^k - U^k G^k d\bar{u}^k - U^k G^k d\bar{v}^k - U^k g^k \\ R_2^k &= (V^k + d\bar{V}^k)\tilde{g}^k - d\bar{V}^k d\bar{y}^k - V^k G^k d\bar{u}^k - d\bar{V}^k g^k - V^k G^k d\bar{v}^k - V^k g^k \\ \hat{r}_1^k &= -A^t y^k \\ \hat{r}_2^k &= -Ax^k - u^k + v^k + b \\ \hat{r}_3^k &= -U^k(g^k + y^k) + \mu^k e - R_1^k \\ \hat{r}_4^k &= -V^k(g^k - y^k) + \mu^k e - R_2^k \\ \hat{r}^k &= \hat{r}_1^k + A^t (D_2^k)^{-1} \{ D_1^k \hat{r}_2^k + (D_u^k + U^k G^k) \hat{r}_4^k - (D_v^k + V^k G^k) \hat{r}_3^k \} \\ \hat{d}x^k &= (A^t (D_2^k)^{-1} D_1^k A)^{-1} \hat{r}^k \\ \hat{d}y^k &= (D_2^k)^{-1} \left\{ D_1^k (A \hat{d}x^k - \hat{r}_2^k) - (D_u^k + U^k G^k) \hat{r}_4^k + (D_v^k + V^k G^k) \hat{r}_3^k \right\} \\ \hat{d}v^k &= (D_1^k)^{-1} \left[D_u^k (\hat{r}_4^k + V^k \hat{d}y^k) - V^k G^k \hat{r}_3^k + V^k U^k G^k \hat{d}y^k \right] \\ \hat{d}u^k &= (D_u^k)^{-1} (\hat{r}_3^k - U^k \hat{d}y^k - U^k G^k \hat{d}v^k) \\ \alpha^k &= \min \left\{ \tau \left(\min_{d_{u_i}^k < 0} -\frac{u_i^k}{d_{u_i}^k} \right), \tau \left(\min_{d_{v_i}^k < 0} -\frac{v_i^k}{d_{v_i}^k} \right), 1 \right\} \\ x^{k+1} &= x^k + \alpha^k \hat{d}x^k \\ y^{k+1} &= y^k + \alpha^k \hat{d}y^k \\ u^{k+1} &= u^k + \alpha^k \hat{d}u^k \\ v^{k+1} &= v^k + \alpha^k \hat{d}v^k \\ \mu^{k+1} &= \frac{\mu^k}{\beta} \end{aligned}$$

Até convergir.

Critério de Convergência: O critério de convergência é o mesmo do método barreira logarítmica definido em (3.7) na Seção 3.5.1.

4.3 Método Primal-Dual Barreira Logarítmica

- Cálculo de dv :

$$dv = (F)^{-1} [E + dy + (Dz_u)^{-1}Gdy],$$

onde $F = (Dz_v - (G)^2(Dz_u)^{-1})$ e $E = [r_4 + (Dz_u)^{-1}(-Gr_3 - G(U)^{-1}r_5) + (V)^{-1}r_6]$.

Multiplicando ambos os lados dessa equação por F :

$$[Dz_v - (G)^2(Dz_u)^{-1}] dv = r_4 + (Dz_u)^{-1}(-Gr_3 - G(U)^{-1}r_5) + (V)^{-1}r_6 + dy + (Dz_u)^{-1}Gdy.$$

Multiplicando ambos os lados dessa equação por Dz_u :

$$[Dz_u Dz_v - (G)^2] dv = Dz_u r_4 - Gr_3 - G(U)^{-1}r_5 + Dz_u(V)^{-1}r_6 + Dz_u dy + Gdy.$$

Substituindo $Dz_u = (G + U^{-1}Zu)$ e $Dz_v = (G + V^{-1}Zv)$:

$$[(G)^2 + GV^{-1}Zv + GU^{-1}Zu + U^{-1}ZuV^{-1}Zv - (G)^2] dv = (G + U^{-1}Zu)(r_4 + (V)^{-1}r_6 + dy) - Gr_3 - G(U)^{-1}r_5 + Gdy.$$

Observe que podemos comutar as matrizes $U, U^{-1}, V, V^{-1}, G, Zu, Zv, Dz_u$ e Dz_v , pois são matrizes diagonais.

$$\Rightarrow [GV^{-1}Zv + GU^{-1}Zu + U^{-1}ZuV^{-1}Zv] dv = (G + U^{-1}Zu)(r_4 + (V)^{-1}r_6 + dy) - Gr_3 - G(U)^{-1}r_5 + Gdy.$$

Multiplicando ambos os lados dessa equação por UV :

$$[UGZv + VGZu + ZuZv] dv = (UVG + VZu)(r_4 + (V)^{-1}r_6 + dy) - UVGr_3 - VGr_5 + UVGdy.$$

De $Dz_u = (G + U^{-1}Zu)$, obtemos que $UG = (UDz_u - Zu)$. Substituindo UG no lado esquerdo da equação:

$$[(UDz_u - Zu)Zv + VGZu + ZuZv] dv = UVGr_4 + UGr_6 + UVGdy + VZur_4 + Zur_6 + VZudy - UVGr_3 - VGr_5 + UVGdy$$

$$\Rightarrow (UDz_uZv + VGZu) dv = UG(Vr_4 + r_6) + Zu(Vr_4 + r_6) + V(UG + Zu) dy - VG(Ur_3 + r_5) + UVGdy$$

$$\Rightarrow (UDz_uZv + VGZu) dv = (UG + Zu)(Vr_4 + r_6 + Vdy) - VG(Ur_3 + r_5) + UVGdy$$

$$\Rightarrow (UDz_uZv + VGZu) dv = (UG + Zu)(Vr_4 + r_6) - VG(Ur_3 + r_5) + ZuVdy + 2UVGdy$$

$$\Rightarrow (UDz_uZv + VGZu) dv = (UG + Zu)(Vr_4 + r_6) - VG(Ur_3 + r_5) + (Zu + 2UG)Vdy.$$

Fazendo $D_1 = (UDz_uZv + VGZu)$, temos:

$$dv = (D_1)^{-1} \{ (UG + Zu)(Vr_4 + r_6) - VG(Ur_3 + r_5) + (Zu + 2UG)Vdy \}.$$

- Cálculo de dy :

$$dy = D[r_2 - Adx - (Dz_u)^{-1}(r_3 + (U)^{-1}r_5) + (F)^{-1}(I + (Dz_u)^{-1}G)E],$$

onde:

$$D = \left[-(Dz_u)^{-1} - (F)^{-1}(I + (Dz_u)^{-1}G)^2 \right]^{-1},$$

$$F = (Dz_v - (G)^2(Dz_u)^{-1}),$$

$$E = [r_4 + (Dz_u)^{-1}(-Gr_3 - G(U)^{-1}r_5) + (V)^{-1}r_6].$$

Multiplicando ambos os lados dessa equação por D^{-1} :

$$\left[-(Dz_u)^{-1} - (Dz_v - (G)^2(Dz_u)^{-1})^{-1}(I + (Dz_u)^{-1}G)^2 \right] dy = r_2 - Adx - (Dz_u)^{-1}(r_3 + (U)^{-1}r_5) + (Dz_v - (G)^2(Dz_u)^{-1})^{-1}(I + (Dz_u)^{-1}G)[r_4 + (Dz_u)^{-1}(-Gr_3 - G(U)^{-1}r_5) + (V)^{-1}r_6].$$

Multiplicando ambos os lados dessa equação por $(Dz_v - (G)^2(Dz_u)^{-1})$:

$$\begin{aligned} & [-(Dz_v - (G)^2(Dz_u)^{-1})(Dz_u)^{-1} - I - 2(Dz_u)^{-1}G - (Dz_u)^{-2}G^2] dy = \\ & = (Dz_v - (G)^2(Dz_u)^{-1})[r_2 - Adx - (Dz_u)^{-1}(r_3 + (U)^{-1}r_5)] + (I + (Dz_u)^{-1}G)[r_4 + (Dz_u)^{-1} \\ & (-Gr_3 - G(U)^{-1}r_5) + (V)^{-1}r_6] \\ & \Rightarrow (-Dz_v(Dz_u)^{-1} - I - 2(Dz_u)^{-1}G) dy = (Dz_v - (G)^2(Dz_u)^{-1})[r_2 - Adx - (Dz_u)^{-1}(U)^{-1}r_5] + \\ & (I + (Dz_u)^{-1}G)[r_4 - (Dz_u)^{-1}G(U)^{-1}r_5 + (V)^{-1}r_6] - Dz_v(Dz_u)^{-1}r_3 + G^2(Dz_u)^{-2}r_3 - G^2(Dz_u)^{-2}r_3 \\ & - (Dz_u)^{-1}Gr_3. \end{aligned}$$

Multiplicando ambos os lados por $(-Dz_u)$:

$$\begin{aligned}
(Dz_v + Dz_u + 2G) dy &= (Dz_v Dz_u - G^2) [-r_2 + Adx + (Dz_u)^{-1}(U)^{-1}r_5] + (Dz_u + G) [-r_4 + \\
&(Dz_u)^{-1}G(U)^{-1}r_5 - (V)^{-1}r_6] + Dz_v r_3 + Gr_3 \\
\Rightarrow (Dz_v + Dz_u + 2G) dy &= (Dz_v Dz_u - G^2) [-r_2 + Adx] + Dz_v(U)^{-1}r_5 - G^2(Dz_u)^{-1}(U)^{-1}r_5 + \\
&(Dz_u + G) [-r_4 - (V)^{-1}r_6] + G(U)^{-1}r_5 + G^2(Dz_u)^{-1}(U)^{-1}r_5 + (Dz_v + G) r_3 \\
\Rightarrow (Dz_v + Dz_u + 2G) dy &= (Dz_v Dz_u - G^2) [-r_2 + Adx] + Dz_v(U)^{-1}r_5 + (Dz_u + G) [-r_4 - (V)^{-1}r_6] \\
&+ G(U)^{-1}r_5 + (Dz_v + G) r_3.
\end{aligned}$$

Multiplicando ambos os lados por UV e substituindo $UV(Dz_v Dz_u - G^2)$ por D_1 :

$$\begin{aligned}
(UVG + UZv + UVG + VZu + 2UVG) dy &= D_1(Adx - r_2) + (VG + Zv)r_5 + (UVG + VZu + \\
UVG)(-r_4 - (V)^{-1}r_6) + VGr_5 + (UVG + UZv + UVG)r_3 \\
\Rightarrow (UZv + VZu + 4UVG) dy &= D_1(Adx - r_2) + 2VGr_5 + Zvr_5 - (VZu + 2VUG)r_4 - \\
2UGr_6 - Zur_6 + (UZv + 2VUG)r_3 \\
\Rightarrow (UZv + VZu + 4UVG) dy &= D_1(Adx - r_2) + (Zv + 2VG)(r_5 + Ur_3) - (Zu + 2UG)(r_6 + Vr_4).
\end{aligned}$$

Fazendo $D_2 = (UZv + VZu + 4UVG)$, temos:

$$dy = (D_2)^{-1} \{D_1(Adx - r_2) + (Zv + 2VG)(r_5 + Ur_3) - (Zu + 2UG)(r_6 + Vr_4)\}.$$

- Cálculo de dx :

$$A^t dy = r_1$$

$$\Rightarrow A^t(D_2)^{-1} \{D_1(Adx - r_2) + (Zv + 2VG)(r_5 + Ur_3) - (Zu + 2UG)(r_6 + Vr_4)\} = r_1$$

$$\Rightarrow A^t(D_2)^{-1} D_1 Adx = r_1 + A^t(D_2)^{-1} \{D_1 r_2 - (Zv + 2VG)(r_5 + Ur_3) + (Zu + 2UG)(r_6 + Vr_4)\}.$$

$$\text{Fazendo } r = r_1 + A^t(D_2)^{-1} \{D_1 r_2 - (Zv + 2VG)(r_5 + Ur_3) + (Zu + 2UG)(r_6 + Vr_4)\},$$

temos:

$$dx = (A^t(D_2)^{-1} D_1 A)^{-1} r.$$

- Cálculo de du :

$$du = (Dz_u)^{-1} (r_3 - dy - Gdv + U^{-1}r_5).$$

Multiplicando ambos os lados dessa equação por Dz_u e substituindo Dz_u por $(G + U^{-1}Zu)$:

$$(G + U^{-1}Zu) du = r_3 - dy - Gdv + U^{-1}r_5.$$

Multiplicando por U :

$$(UG + Zu) du = U(r_3 - dy - Gdv) + r_5.$$

Obtemos então:

$$du = (UG + Zu)^{-1} [U(r_3 - dy - Gdv) + r_5].$$

Com essas modificações teremos $6m$ operações a menos no cálculo de dx , entretanto teremos $4m$ operações adicionais no cálculo de dv por iteração. Mesmo assim compensa fazer essas modificações porque ao todo teremos $2m$ operações a menos.

Modificaremos o cálculo de z_v^0 e z_u^0 . Na solução ótima $\mu \rightarrow 0$, então teremos $Uz_u = 0$ e $Vz_v = 0$ nas condições de otimalidade dadas em (3.12). Assim, se fizermos $z_v^0 = e$, de $Uz_u = Vz_v = 0$ teremos: $z_u^0 = U^{-1}Ve$.

Algoritmo Primal-Dual Barreira Logarítmica

Os pontos iniciais $x^0, y^0, (u^0, v^0) > 0, \mu^0$ são calculados como em (4.1).

$$z_v^0 = e, z_u^0 = U^{-1}Ve, \text{ onde } e = [1, 1, \dots, 1]^t.$$

Assim, dados $x^0, y^0, (u^0, v^0) > 0, \mu^0, \beta > 1, 0 < \sigma \leq 1$ e $\tau = 0,99995$.

Para $k = 0, 1, 2, \dots$ faça:

$$\begin{aligned}
g^k &= p(u^k + v^k)^{p-1} \\
G^k &= \begin{cases} \frac{p(p-1)}{(u_i^k + v_i^k)^{2-p}}, & \text{se } i = j, \\ 0, & \text{se } i \neq j. \end{cases} \\
U^k &= \text{diag}(u^k) \\
V^k &= \text{diag}(v^k) \\
Zu^k &= \text{diag}(z_u^k) \\
Zv^k &= \text{diag}(z_v^k) \\
r_1^k &= -A^t y^k \\
r_2^k &= -Ax^k - u^k + v^k + b \\
r_3^k &= -g^k + z_u^k - y^k \\
r_4^k &= -g^k + z_v^k + y^k \\
r_5^k &= -U^k Z u^k e + \mu^k e \\
r_6^k &= -V^k Z v^k e + \mu^k e \\
Dz_u^k &= (G^k + (U^k)^{-1} Z u^k) \\
D_1^k &= (U^k D z_u^k Z v^k + V^k G^k Z u^k) \\
D_2^k &= (U^k Z v^k + V^k Z u^k + 4U^k V^k G^k) \\
r^k &= r_1^k + A^t (D_2^k)^{-1} \{ D_1^k r_2^k - (Z v^k + 2V^k G^k) (r_5^k + U^k r_3^k) + (Z u^k + 2U^k G^k) (r_6^k + \\
&\quad + V^k r_4^k) \} \\
dx^k &= (A^t (D_2^k)^{-1} D_1^k A)^{-1} r^k \\
dy^k &= (D_2^k)^{-1} \{ D_1^k (A dx^k - r_2^k) + (Z v^k + 2V^k G^k) (r_5^k + U^k r_3^k) - (Z u^k + 2U^k G^k) (r_6^k + \\
&\quad + V^k r_4^k) \} \\
dv^k &= (D_1^k)^{-1} \{ (U^k G^k + Z u^k) (V^k r_4^k + r_6^k) - V^k G^k (U^k r_3^k + r_5^k) + (Z u^k + 2U^k G^k) \\
&\quad V^k dy^k \}
\end{aligned}$$

$$\begin{aligned}
du^k &= (U^k G^k + Z u^k)^{-1} [U^k (r_3^k - dy^k - G^k dv^k) + r_5^k] \\
dz_v^k &= (V^k)^{-1} (r_6^k - Z_v^k dv^k) \\
dz_u^k &= (U^k)^{-1} (r_5^k - Z_u^k du^k) \\
\alpha^k &= \min \left\{ \tau \left(\min_{du_i^k < 0} -\frac{u_i^k}{du_i^k} \right), \tau \left(\min_{dv_i^k < 0} -\frac{v_i^k}{dv_i^k} \right), \tau \left(\min_{dz_{u_i}^k < 0} -\frac{z_{u_i}^k}{dz_{u_i}^k} \right), \tau \left(\min_{dz_{v_i}^k < 0} -\frac{z_{v_i}^k}{dz_{v_i}^k} \right), 1 \right\} \\
x^{k+1} &= x^k + \alpha^k dx^k \\
y^{k+1} &= y^k + \alpha^k dy^k \\
u^{k+1} &= u^k + \alpha^k du^k \\
v^{k+1} &= v^k + \alpha^k dv^k \\
z_u^{k+1} &= z_u^k + \alpha^k dz_u^k \\
z_v^{k+1} &= z_v^k + \alpha^k dz_v^k \\
\mu^{k+1} &= \frac{\mu^k}{\beta}
\end{aligned}$$

Até convergir.

Cr terio de Converg ncia: O crit rio de converg ncia   o mesmo do m todo primal-dual barreira logar tmica definido em (3.13) na Se o 3.5.3.

4.4 M todo Primal-Dual Barreira Logar tmica Preditor-Corretor

As modifica es nas dire es desse m todo s o an logas  s do m todo primal-dual barreira logar tmica.

Algoritmo Primal-Dual Barreira Logarítmica Preditor-Corretor

O ponto inicial é calculado como no método primal-dual barreira logarítmica na Seção 4.3.

Assim, dados $x^0, y^0, (u^0, v^0) > 0, \mu^0, \beta > 1, 0 < \sigma \leq 1$ e $\tau = 0,99995$.

Para $k = 0, 1, 2, \dots$ faça:

Passo Preditor:

$$g^k = p(u^k + v^k)^{p-1}$$

$$G^k = \begin{cases} \frac{p(p-1)}{(u_i^k + v_i^k)^{2-p}}, & \text{se } i = j, \\ 0, & \text{se } i \neq j. \end{cases}$$

$$U^k = \text{diag}(u^k)$$

$$V^k = \text{diag}(v^k)$$

$$Zu^k = \text{diag}(z_u^k)$$

$$Zv^k = \text{diag}(z_v^k)$$

$$\bar{r}_1^k = -A^t y^k$$

$$\bar{r}_2^k = -Ax^k - u^k + v^k + b$$

$$\bar{r}_3^k = -g^k + z_u^k - y^k$$

$$\bar{r}_4^k = -g^k + z_v^k + y^k$$

$$\bar{r}_5^k = -U^k Zu^k e$$

$$\bar{r}_6^k = -V^k Zv^k e$$

$$Dz_u^k = (G^k + (U^k)^{-1} Zu^k)$$

$$Dz_v^k = (G^k + (V^k)^{-1} Zv^k)$$

$$D_1^k = (U^k Dz_u^k Zv^k + V^k G^k Zu^k)$$

$$D_2^k = (U^k Zv^k + V^k Zu^k + 4U^k V^k G^k)$$

$$\bar{r}^k = \bar{r}_1^k + A^t (D_2^k)^{-1} \{ D_1^k \bar{r}_2^k - (Zv^k + 2V^k G^k) (\bar{r}_5^k + U^k \bar{r}_3^k) + (Zu^k + 2U^k G^k) (\bar{r}_6^k + V^k \bar{r}_4^k) \}$$

$$\begin{aligned}
\bar{d}x^k &= (A^t(D_2^k)^{-1}D_1^kA)^{-1}\bar{r}^k \\
\bar{d}y^k &= (D_2^k)^{-1}\{D_1^k(A\bar{d}x^k - \bar{r}_2^k) + (Zv^k + 2V^kG^k)(\bar{r}_5^k + U^k\bar{r}_3^k) - (Zu^k + 2U^kG^k)(\bar{r}_6^k + \\
&\quad + V^k\bar{r}_4^k)\} \\
\bar{d}v^k &= (D_1^k)^{-1}\{(U^kG^k + Zu^k)(V^k\bar{r}_4^k + \bar{r}_6^k) - V^kG^k(U^k\bar{r}_3^k + \bar{r}_5^k) + (Zu^k + 2U^kG^k) \\
&\quad V^k\bar{d}y^k\} \\
\bar{d}u^k &= (Dz_u^k)^{-1}(\bar{r}_3^k - \bar{d}y^k - G^k\bar{d}v^k + (U^k)^{-1}\bar{r}_5^k) \\
\bar{d}z_v^k &= (V^k)^{-1}(\bar{r}_6^k - Zv^k\bar{d}v^k) \\
\bar{d}z_u^k &= (U^k)^{-1}(\bar{r}_5^k - Zu^k\bar{d}u^k)
\end{aligned}$$

Passo Corretor:

$$\begin{aligned}
\beta^k &= \min \left\{ \tau \left(\min_{(\bar{d}u_i^k + \bar{d}v_i^k) < 0} -\frac{(u_i^k + v_i^k)}{(\bar{d}u_i^k + \bar{d}v_i^k)} \right), 1 \right\} \\
\tilde{g}^k &= p(u^k + \beta^k\bar{d}u^k + v^k + \beta^k\bar{d}v^k)^{p-1} \\
\bar{d}U^k &= \text{diag}(\bar{d}u^k) \\
\bar{d}V^k &= \text{diag}(\bar{d}v^k) \\
R^k &= \tilde{g}^k - G^k(\bar{d}u^k + \bar{d}v^k) - g^k \\
R_1^k &= \bar{d}U^k\bar{d}z_u^k \\
R_2^k &= \bar{d}V^k\bar{d}z_v^k \\
\hat{r}_1^k &= -A^ty^k \\
\hat{r}_2^k &= -Ax^k - u^k + v^k + b \\
\hat{r}_3^k &= -g^k + z_u^k - y^k - R^k \\
\hat{r}_4^k &= -g^k + z_v^k + y^k - R^k \\
\hat{r}_5^k &= -U^kZ_u^ke + \mu^ke - R_1^k \\
\hat{r}_6^k &= -V^kZ_v^ke + \mu^ke - R_2^k
\end{aligned}$$

$$\begin{aligned}
\hat{r}^k &= \hat{r}_1^k + A^t(D_2^k)^{-1} \{D_1^k \hat{r}_2^k - (Zv^k + 2V^k G^k) (\hat{r}_5^k + U^k \hat{r}_3^k) + (Zu^k + 2U^k G^k) (\hat{r}_6^k + \\
&\quad + V^k \hat{r}_4^k) \} \\
\hat{d}x^k &= (A^t(D_2^k)^{-1} D_1^k A)^{-1} \hat{r}^k \\
\hat{d}y^k &= (D_2^k)^{-1} \{D_1^k (A \hat{d}x^k - \hat{r}_2^k) + (Zv^k + 2V^k G^k) (\hat{r}_5^k + U^k \hat{r}_3^k) - (Zu^k + 2U^k G^k) (\hat{r}_6^k + \\
&\quad + V^k \hat{r}_4^k) \} \\
\hat{d}v^k &= (D_1^k)^{-1} \{ (U^k G^k + Zu^k) (V^k \hat{r}_4^k + \hat{r}_6^k) - V^k G^k (U^k \hat{r}_3^k + \hat{r}_5^k) + (Zu^k + 2U^k G^k) \\
&\quad V^k \hat{d}y^k \} \\
\hat{d}u^k &= (Dz_u^k)^{-1} (\hat{r}_3^k - \hat{d}y^k - G^k \hat{d}v^k + (U^k)^{-1} \hat{r}_5^k) \\
\hat{d}z_v^k &= (V^k)^{-1} (\hat{r}_6^k - Zv^k \hat{d}v^k) \\
\hat{d}z_u^k &= (U^k)^{-1} (\hat{r}_5^k - Zu^k \hat{d}u^k) \\
\alpha^k &= \min \left\{ \tau \left(\min_{\hat{d}u_i^k < 0} -\frac{u_i^k}{\hat{d}u_i^k} \right), \tau \left(\min_{\hat{d}v_i^k < 0} -\frac{v_i^k}{\hat{d}v_i^k} \right), \tau \left(\min_{\hat{d}z_{ui}^k < 0} -\frac{z_{ui}^k}{\hat{d}z_{ui}^k} \right), \tau \left(\min_{\hat{d}z_{vi}^k < 0} -\frac{z_{vi}^k}{\hat{d}z_{vi}^k} \right), 1 \right\} \\
x^{k+1} &= x^k + \alpha^k \hat{d}x^k \\
y^{k+1} &= y^k + \alpha^k \hat{d}y^k \\
u^{k+1} &= u^k + \alpha^k \hat{d}u^k \\
v^{k+1} &= v^k + \alpha^k \hat{d}v^k \\
z_u^{k+1} &= z_u^k + \alpha^k \hat{d}z_u^k \\
z_v^{k+1} &= z_v^k + \alpha^k \hat{d}z_v^k \\
\mu^{k+1} &= \frac{\mu^k}{\beta}
\end{aligned}$$

Até convergir.

Critério de Convergência: O critério de convergência é o mesmo do método primal-dual barreira logarítmica definido em (3.13) na Seção 3.5.3.

Capítulo 5

Estudo da Estrutura Matricial

Para o estudo da estrutura matricial dos métodos de pontos interiores descritos no Capítulo 4, vamos fazer algumas observações acerca das matrizes envolvidas nos cálculos.

Nosso objetivo consiste em resolver o problema (3.1) descrito no Capítulo 3. Como o foco desse trabalho é resolver problemas de regressão polinomial, a matriz A é de Vandermonde. Assim, A é da seguinte forma:

$$A = \begin{bmatrix} 1 & a_1 & a_1^2 & \cdots & a_1^{n-1} \\ 1 & a_2 & a_2^2 & \cdots & a_2^{n-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & a_m & a_m^2 & \cdots & a_m^{n-1} \end{bmatrix}_{m \times n}$$

Usamos o fato de a matriz A ser de Vandermonde para ocupar pouca memória, armazenando apenas os valores dados pela segunda coluna da matriz A .

Para encontrar os pontos iniciais, em todos os métodos utilizamos a matriz $A^t A$.

$$A^t A = \begin{bmatrix} m & \sum_{i=1}^m a_i & \sum_{i=1}^m a_i^2 & \cdots & \sum_{i=1}^m a_i^{n-1} \\ \sum_{i=1}^m a_i & \sum_{i=1}^m a_i^2 & \sum_{i=1}^m a_i^3 & \cdots & \sum_{i=1}^m a_i^n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \sum_{i=1}^m a_i^{n-1} & \sum_{i=1}^m a_i^n & \sum_{i=1}^m a_i^{n+1} & \cdots & \sum_{i=1}^m a_i^{2n-2} \end{bmatrix}_{n \times n}$$

A matriz $A^t A$ é uma matriz inversível, pois A tem posto completo porque nenhum ponto se repete, ou seja, $a_i \neq a_j$, $\forall i, j = 1, \dots, m$, com $i \neq j$. Além de $A^t A$ ser inversível, tem uma propriedade especial, é uma matriz de Hankel, ou seja, os elementos de cada antidiagonal são iguais.

Em [25] e [15] encontramos uma forma eficiente de resolver sistemas lineares quando a matriz do lado esquerdo é uma matriz de Hankel. Isso será utilizado na implementação computacional, visto que na resolução dos problemas de regressão pela norma L_p utilizando métodos de pontos interiores, no cálculo do ponto inicial e das direções nos deparamos com matrizes de Hankel, consequência do fato da matriz A ser de Vandermonde.

Também nos deparamos com matrizes do tipo $A^t D A$ nos cálculos que envolvem as direções, onde D é uma matriz diagonal não-singular. Sendo assim, se definirmos D da forma:

$$D = \begin{bmatrix} d_1 & 0 & 0 & \cdots & 0 \\ 0 & d_2 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & d_m \end{bmatrix}_{m \times m},$$

então teremos:

$$A^tDA = \begin{bmatrix} \sum_{i=1}^m d_i & \sum_{i=1}^m d_i a_i & \sum_{i=1}^m d_i a_i^2 & \cdots & \sum_{i=1}^m d_i a_i^{n-1} \\ \sum_{i=1}^m d_i a_i & \sum_{i=1}^m d_i a_i^2 & \sum_{i=1}^m d_i a_i^3 & \cdots & \sum_{i=1}^m d_i a_i^n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \sum_{i=1}^m d_i a_i^{n-1} & \sum_{i=1}^m d_i a_i^n & \sum_{i=1}^m d_i a_i^{n+1} & \cdots & \sum_{i=1}^m d_i a_i^{2n-2} \end{bmatrix}_{n \times n} .$$

Podemos observar dessa forma que a matriz A^tDA também é uma matriz não-singular e de Hankel.

Vejamos agora cada caso que contribuirá para uma implementação eficiente dos métodos de pontos interiores.

5.1 Cálculo de Ax

Para efetuar a multiplicação da matriz $A_{m \times n}$ por um vetor x_n , usamos um método capaz de efetuar essa multiplicação em $(2mn - 2m)$ operações, enquanto que comumente levaríamos $2mn$ operações. Considerando um problema de grande porte, essa diferença dá mais velocidade ao método.

Para ficar claro o número de operações envolvidas, segue abaixo um exemplo de pequeno porte.

Considere a matriz $A_{1 \times 3}$:

$$A = \begin{bmatrix} 1 & 2 & 2^2 \end{bmatrix}, \text{ e } x = \begin{bmatrix} 2 \\ 5 \\ 7 \end{bmatrix} .$$

O produto Ax é feito da seguinte forma:

Dado a

$$\text{aux} = x[n]$$

Para $j = n - 1 : -1 : 1$

$$\text{aux} = x[j] + a * \text{aux}$$

Nesse exemplo esse procedimento faz o seguinte:

$$\text{aux} = 7$$

$j = 2$:

$$\text{aux} = x[2] + a * \text{aux} = 5 + 2 * 7 = 19$$

$j = 1$:

$$\text{aux} = x[1] + a * \text{aux} = 2 + 2 * 19 = 40$$

Então o procedimento fez o cálculo: $2 + 2 * 5 + 2^2 * 7$, ou seja, Ax em $(2mn - 2m)$ operações, para esse exemplo $m = 1$ e $n = 3$.

5.2 Cálculo de $A^t A$

Como vimos, o fato de a matriz A ser de Vandermonde faz com que $A^t A$ seja uma matriz de Hankel, sendo assim, muitos elementos de $A^t A$ são iguais, por isso em nossa implementação armazenamos apenas os elementos diferentes, ou seja, $2n - 1$ elementos. A multiplicação de $A^t A$ requer $2mn^2$ operações no caso de $A_{m \times n}$, enquanto que em nossa implementação o número de operações necessárias para realizar essa multiplicação é $(4mn - 3m)$ operações.

Seja $A_{m \times n}$, para calcular $A^t A$ fazemos:

$$h[0] = m$$

Para $i = 1 : m - 1$

$$\text{aux} = A[i, 2]$$

$$h[1] = h[1] + \text{aux}$$

Para $j = 2 : 2n$

$$\text{aux} = \text{aux} * A[i, 2]$$

$$h[j] = h[j] + \text{aux}$$

O vetor h guarda os elementos distintos de $A^t A$. Não é necessário armazenar todos os elementos de A porque a primeira coluna contém apenas 1, e poderíamos obter a terceira coluna até a n -ésima a partir da segunda coluna, mas na prática, tanto multiplicar A por um vetor como calcular $A^t A$, as colunas adicionais de A não precisam ser calculadas.

Vejam os um exemplo de como utilizar o procedimento acima para calcular $A^t A$. Considere $A_{3 \times 3}$:

$$A = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 4 \\ 1 & 3 & 9 \end{bmatrix} \Rightarrow A^t A = \begin{bmatrix} 3 & 6 & 14 \\ 6 & 14 & 36 \\ 14 & 36 & 98 \end{bmatrix}$$

Observe que nesse exemplo $m = n$, dessa forma a matriz de Vandermonde A é inversível e a solução para o problema de regressão pela norma L_p é direta. Porém, nosso intuito é trabalhar com problemas grandes em que $m > n$.

Para esse exemplo temos os seguintes cálculos:

$$h[0] = 3$$

$i = 1$:

$$\text{aux} = A[1, 2] = 1$$

$$h[1] = h[1] + 1 = 0 + 1 = 1$$

$j = 2$:

$$\text{aux} = \text{aux} * A[1, 2] = 1 * 1 = 1$$

$$h[2] = h[2] + \text{aux} = 0 + 1 = 1$$

$j = 3$:

$$\text{aux} = \text{aux} * A[1, 2] = 1 * 1 = 1$$

$$h[3] = h[3] + \text{aux} = 0 + 1 = 1$$

$$j = 4:$$

$$\text{aux} = \text{aux} * A[1, 2] = 1 * 1 = 1$$

$$h[4] = h[4] + \text{aux} = 0 + 1 = 1$$

$$i = 2:$$

$$\text{aux} = A[2, 2] = 2$$

$$h[1] = h[1] + 2 = 1 + 2 = 3$$

$$j = 2:$$

$$\text{aux} = \text{aux} * A[2, 2] = 2 * 2 = 4$$

$$h[2] = h[2] + \text{aux} = 1 + 4 = 5$$

$$j = 3:$$

$$\text{aux} = \text{aux} * A[2, 2] = 4 * 2 = 8$$

$$h[3] = h[3] + \text{aux} = 1 + 8 = 9$$

$$j = 4:$$

$$\text{aux} = \text{aux} * A[2, 2] = 8 * 2 = 16$$

$$h[4] = h[4] + \text{aux} = 1 + 16 = 17$$

$$i = 3:$$

$$\text{aux} = A[3, 2] = 3$$

$$h[1] = h[1] + 3 = 3 + 3 = 6$$

$$j = 2:$$

$$\text{aux} = \text{aux} * A[3, 2] = 3 * 3 = 9$$

$$h[2] = h[2] + \text{aux} = 5 + 9 = 14$$

$j = 3$:

$$\text{aux} = \text{aux} * A[3, 2] = 9 * 3 = 27$$

$$h[3] = h[3] + \text{aux} = 9 + 27 = 36$$

$j = 4$:

$$\text{aux} = \text{aux} * A[3, 2] = 27 * 3 = 81$$

$$h[4] = h[4] + \text{aux} = 17 + 81 = 98$$

Dessa forma encontramos o vetor h dado por:

$$h = \begin{bmatrix} 3 \\ 6 \\ 14 \\ 36 \\ 98 \end{bmatrix} .$$

Observe que para cada j efetuamos 2 operações para cada valor de i , daí temos um total de $2(2n - 2)$ operações para cada i , isso implica em um total de $2m(2n - 2)$ operações para todos os j dentro do laço de i , temos ainda m operações adicionais dentro do laço de i e fora do laço em j , isso resulta num total geral de $2m(2n - 2) + m = (4mn - 3m)$ operações.

5.3 Inversa de uma Matriz de Hankel

Será apresentado nessa seção um algoritmo para calcular a inversa de uma matriz de Hankel. Para mais detalhes e demonstrações do algoritmo, ver [25].

Podemos representar uma matriz de Hankel de ordem $n + 1$ da forma:

$$H_n = \begin{bmatrix} C_0 & C_1 & \cdots & C_n \\ C_1 & C_2 & \cdots & C_{n+1} \\ \vdots & \vdots & \ddots & \vdots \\ C_n & C_{n+1} & \cdots & C_{2n} \end{bmatrix}_{n+1 \times n+1} = (C_{i+j})_{i,j=0}^n.$$

Quando as matrizes H_0, H_1, \dots, H_n são não-singulares, o número de multiplicações neces-

sárias para inverter H_n é proporcional a $(n+1)^2$, ao invés de $(n+1)^3$ como ocorre em métodos para inverter uma matriz simétrica arbitrária de ordem $n+1$.

Observe também que n representa o grau do polinômio da regressão, o que na prática significa que não tem um valor muito alto.

Para calcular a inversa de H_n devemos seguir o método abaixo.

Método para inversão de uma matriz de Hankel

Vamos supor que seja necessário inverter uma matriz H_{m+1} , e que $H_{k-1}, H_k, \dots, H_{m+1}$ são não-singulares para $k \leq m$.

- Encontre $u_{0,k-1}, u_{1,k-1}, \dots, u_{k-1,k-1}$ e $u_{0,k}, u_{1,k}, \dots, u_{k,k}$ resolvendo o sistema:

$$\sum_{j=0}^n C_{r+j} u_{jn} = -C_{n+r+1}, \quad 0 \leq r \leq n,$$

para $n = k-1$ e $n = k$.

- Encontre λ_k :

$$\lambda_k = \sum_{s=0}^k C_{k+s} u_{s,k-1}.$$

- Encontre γ_k :

$$\gamma_k = \sum_{j=0}^k C_{k+j+1} u_{j,k-1}.$$

- Para $k \leq n \leq m - 1$, calcule recursivamente:

$$\lambda_{n+1} = \sum_{j=0}^{n+1} C_{n+j+1} u_{jn},$$

$$\gamma_{n+1} = \sum_{j=0}^{n+1} C_{n+j+2} u_{jn}.$$

- Para $0 \leq s \leq n + 1$, calcule:

$$u_{s,n+1} = (\lambda_n^{-1} \gamma_n - \lambda_{n+1}^{-1} \gamma_{n+1}) u_{sn} + u_{s-1,n} - \lambda_n^{-1} \lambda_{n+1} u_{s,n-1},$$

onde $u_{s-1,n} = u_{n+1,n-1} = 0$ e $u_{n+1,n} = 1$.

- Para $0 \leq r \leq s \leq m$, calcule:

$$b_{rsm} = b_{r-1,s+1,m} + \lambda_m^{-1} (u_{r,m-1} u_{s+1,m} - u_{r,m} u_{s+1,m-1}),$$

onde $b_{r-1,s+1,m} = b_{r-1,m+1,m} = 0$.

- Calcule λ_{m+1} :

$$\lambda_{m+1} = \sum_{j=0}^{m+1} C_{m+j+1} u_{jm}.$$

- Para $0 \leq r \leq s \leq m + 1$, calcule:

$$b_{rs,m+1} = b_{rsm} + \lambda_{m+1}^{-1} u_{rm} u_{sm}.$$

- Para $0 \leq s \leq r \leq m + 1$, temos:

$$b_{rs,m+1} = b_{sr,m+1}.$$

Dessa forma, a matriz H_{m+1}^{-1} é dada por:

$$H_{m+1}^{-1} = B_{m+1} = (b_{r,s,m+1})_{r,s=0}^{m+1}.$$

Usaremos esse algoritmo na implementação eficiente para calcular a inversa de uma matriz de Hankel, matriz que aparece no cálculo das direções dos métodos de pontos interiores.

Capítulo 6

Resultados Computacionais

No Capítulo 3 apresentamos os métodos de pontos interiores descritos em [7] já com as modificações no cálculo dos resíduos, e no Capítulo 4 apresentamos modificações no cálculo das direções e do ponto inicial. Nesse trabalho fizemos essas três modificações com o intuito de aperfeiçoar os métodos, e fizemos uma implementação eficiente. Nesse capítulo apresentaremos os resultados computacionais da implementação em MATLAB e da implementação eficiente. Os testes computacionais foram feitos no Matlab R2009a, sistema computacional Linux, memória 4GB, processador Intel(R) Core(TM)2.

A notação utilizada em todos os exemplos será a mesma:

Para os métodos atuais, aperfeiçoados nesse trabalho:

- BL: Método Barreira Logarítmica
- BLPC: Método Barreira Logarítmica Preditor-Corretor
- PDBL: Método Primal-Dual Barreira Logarítmica
- PDBLPC: Método Primal-Dual Barreira Logarítmica Preditor-Corretor

Para os métodos já existentes:

- BL(antigo): Método Barreira Logarítmica
- BLPC(antigo): Método Barreira Logarítmica Preditor-Corretor
- PDBL(antigo): Método Primal-Dual Barreira Logarítmica
- PDBLPC(antigo): Método Primal-Dual Barreira Logarítmica Preditor-Corretor

Usaremos it para número de iterações e F_0 para o valor da função objetivo. O tempo está em segundos. Em todos os casos usaremos $\zeta = 2$, ζ é usado no cálculo do ponto inicial dos métodos de pontos interiores já existentes .

6.1 Problema de Grande Porte

Para ficar claro o quanto cada modificação contribuiu ou não, vamos fazer inicialmente três testes, um apenas com a modificação no ponto inicial, outro apenas com a modificação no resíduo e o último com a modificação apenas nas direções. Faremos esses testes com o problema de grande porte que está descrito em [7]. Esse problema compreende valores de juros diários ao longo de 40 anos, totalizando 10958 valores observados, os dados foram normalizados no intervalo $[0,1]$. Em seguida, apresentaremos os testes com todas as modificações.

Utilizaremos $\tau = 0,99995$, $\beta = 10$, $\epsilon = 10^{-10}$, $\epsilon_1 = 10^{-8}$, onde ϵ e ϵ_1 são usados para a verificação do critério de convergência definido em (3.7), e $\sigma = 10^{-1}$.

6.1.1 Modificação no Ponto Inicial

Vejamos os resultados obtidos apenas com a modificação do ponto inicial.

p	BL			BL(antigo)		
	it	tempo	Fo	it	tempo	Fo
1,1	25	0,2782	2,4550e+04	4	0,0942	2,6040e+04
1,2	19	0,2319	2,8274e+04	24	0,2855	2,8274e+04
1,3	13	0,1642	3,2781e+04	19	0,2320	3,2781e+04
1,4	10	0,1316	3,8243e+04	16	0,2006	3,8243e+04
1,5	9	0,1283	4,4881e+04	14	0,1838	4,4881e+04
1,6	8	0,0980	5,2952e+04	13	0,1539	5,2952e+04
1,7	8	0,0954	6,2775e+04	12	0,1577	6,2775e+04
1,8	8	0,1195	7,4734e+04	11	0,1554	7,4734e+04
1,9	7	0,1086	8,9311e+04	9	0,1335	8,9311e+04

Tabela 6.1.1: Barreira Logarítmica – Ponto Inicial.

p	BLPC			BLPC(antigo)		
	it	tempo	Fo	it	tempo	Fo
1,1	14	0,2280	2,4570e+04	6	0,1325	2,6040e+04
1,2	18	0,2316	2,8274e+04	26	0,3702	2,8274e+04
1,3	15	0,2338	3,2781e+04	19	0,2819	3,2781e+04
1,4	11	0,1793	3,8243e+04	14	0,2136	3,8243e+04
1,5	9	0,1317	4,4881e+04	12	0,1832	4,4881e+04
1,6	8	0,1461	5,2952e+04	10	0,1607	5,2952e+04
1,7	8	0,1355	6,2775e+04	9	0,1495	6,2775e+04
1,8	7	0,1234	7,4734e+04	9	0,1584	7,4734e+04
1,9	7	0,1276	8,9311e+04	9	0,1514	8,9311e+04

Tabela 6.1.2: Barreira Logarítmica Preditor-Corretor – Ponto Inicial.

p	PDBL			PDBL(antigo)		
	it	tempo	Fo	it	tempo	Fo
1,1	20	0,2692	2,4550e+04	33	0,4288	2,4551e+04
1,2	16	0,2206	2,8274e+04	27	0,3473	2,8274e+04
1,3	13	0,1922	3,2781e+04	22	0,3013	3,2781e+04
1,4	11	0,1597	3,8243e+04	20	0,2616	3,8243e+04
1,5	10	0,1502	4,4881e+04	18	0,2543	4,4881e+04
1,6	9	0,1406	5,2952e+04	18	0,2464	5,2952e+04
1,7	9	0,1406	6,2775e+04	18	0,2497	6,2775e+04
1,8	9	0,1390	7,4734e+04	18	0,2566	7,4734e+04
1,9	10	0,1465	8,9311e+04	19	0,2193	8,9311e+04

Tabela 6.1.3: Primal-Dual Barreira Logarítmica – Ponto Inicial.

p	PDBLPC			PDBLPC(antigo)		
	it	tempo	Fo	it	tempo	Fo
1,1	15	0,2845	2,4550e+04	1	0,0658	2,6118e+04
1,2	12	0,2111	2,8274e+04	1	0,0502	2,9726e+04
1,3	9	0,1700	3,2781e+04	1	0,0545	3,4090e+04
1,4	7	0,1430	3,8243e+04	1	0,0536	3,9382e+04
1,5	7	0,1506	4,4881e+04	1	0,0537	4,5816e+04
1,6	6	0,1357	5,3098e+04	1	0,0540	5,3658e+04
1,7	9	0,1846	6,2775e+04	1	0,0535	6,3239e+04
1,8	8	0,1413	7,4734e+04	1	0,0529	7,4975e+04
1,9	7	0,1436	8,9311e+04	1	0,0518	8,9381e+04

Tabela 6.1.4: Primal-Dual Barreira Logarítmica Predictor-Corretor – Ponto Inicial.

A modificação no ponto inicial colabora para reduzir o valor da função objetivo, e de modo geral, também para a diminuição do número de iterações. Aqui estamos apresentando os resultados desse teste com $\sigma = 10^{-1}$, lembrando que σ é usado no cálculo do ponto inicial. Entretanto vale ressaltar que fizemos testes com valores menores de σ (por exemplo $\sigma = 10^{-10}$) e verificamos nesse caso um leve aumento no valor da função objetivo.

Observe que nosso problema possui mínimo global, assim a solução ótima é única. O fato de conseguirmos diminuir o valor da função objetivo na solução ótima mudando o ponto inicial, mostra que os métodos antigos estavam parando antes, pelo segundo critério de parada, definido em (3.7). Com a modificação no ponto inicial, nos aproximamos mais da solução ótima global. Mas vale ressaltar que uma modificação no critério de convergência também poderia fazer com que a solução ótima fosse mais próxima da solução ótima global. Situação parecida aconteceu nos experimentos que serão apresentados a seguir.

6.1.2 Modificação no Cálculo do Resíduo

Alterando apenas o cálculo dos resíduos obtemos os seguintes resultados:

p	BLPC			BLPC(antigo)		
	it	tempo	Fo	it	tempo	Fo
1,1	3	0,1136	2,6165e+04	6	0,1325	2,6040e+04
1,2	4	0,1217	2,9793e+04	26	0,3702	2,8274e+04
1,3	6	0,1322	3,6030e+04	19	0,2819	3,2781e+04
1,4	4	0,1195	3,9351e+04	14	0,2136	3,8243e+04
1,5	7	0,1556	4,4882e+04	12	0,1832	4,4881e+04
1,6	5	0,1361	5,2953e+04	10	0,1607	5,2952e+04
1,7	4	0,1209	6,2775e+04	9	0,1495	6,2775e+04
1,8	4	0,1229	7,4734e+04	9	0,1584	7,4734e+04
1,9	4	0,1206	8,9311e+04	9	0,1514	8,9311e+04

Tabela 6.1.5: Barreira Logarítmica Preditor-Corretor – Resíduo.

p	PDBLPC			PDBLPC(antigo)		
	it	tempo	Fo	it	tempo	Fo
1,1	1	0,0629	2,6118e+04	1	0,0658	2,6118e+04
1,2	1	0,0588	2,9726e+04	1	0,0502	2,9726e+04
1,3	1	0,0563	3,4090e+04	1	0,0545	3,4090e+04
1,4	1	0,0589	3,9382e+04	1	0,0536	3,9382e+04
1,5	1	0,0539	4,5816e+04	1	0,0537	4,5816e+04
1,6	1	0,0502	5,3658e+04	1	0,0540	5,3658e+04
1,7	1	0,0578	6,3239e+04	1	0,0535	6,3239e+04
1,8	1	0,0574	7,4975e+04	1	0,0529	7,4975e+04
1,9	1	0,0566	8,9381e+04	1	0,0518	8,9381e+04

Tabela 6.1.6: Primal-Dual Barreira Logarítmica Preditor-Corretor – Resíduo.

Podemos observar que a modificação no cálculo dos resíduos reduz o número de iterações no primeiro caso, mas aumenta levemente o tempo de execução no último caso. Veja que na Tabela 6.1.6 temos o mesmo número de iterações, mas em alguns casos o tempo aumentou. Isso acontece porque a nova forma de encontrar os resíduos exige mais cálculos do que a forma antiga. Essa modificação no cálculo colabora para a diminuição no número de iterações, mas pode aumentar o tempo computacional. Observe que essas alterações modificam somente os métodos do tipo preditor-corretor.

6.1.3 Modificação no Cálculo das Direções

No Capítulo 4 apresentamos uma nova forma de calcular as direções dos métodos, essa nova maneira reduz o número de operações necessárias em cada iteração. Vejamos o que acontece quando mudamos apenas a forma de calcular essas direções:

p	BL			BL(antigo)		
	it	tempo	Fo	it	tempo	Fo
1,1	4	0,0678	2,6040e+04	4	0,0942	2,6040e+04
1,2	24	0,2486	2,8274e+04	24	0,2855	2,8274e+04
1,3	19	0,1955	3,2781e+04	19	0,2320	3,2781e+04
1,4	16	0,1749	3,8243e+04	16	0,2006	3,8243e+04
1,5	14	0,1242	4,4881e+04	14	0,1838	4,4881e+04
1,6	13	0,1178	5,2952e+04	13	0,1539	5,2952e+04
1,7	12	0,1280	6,2775e+04	12	0,1577	6,2775e+04
1,8	11	0,1086	7,4734e+04	11	0,1554	7,4734e+04
1,9	9	0,0997	8,9311e+04	9	0,1335	8,9311e+04

Tabela 6.1.7: Barreira Logarítmica – Direções.

p	BLPC			BLPC(antigo)		
	it	tempo	Fo	it	tempo	Fo
1,1	6	0,1042	2,6040e+04	6	0,1325	2,6040e+04
1,2	26	0,2623	2,8274e+04	26	0,3702	2,8274e+04
1,3	19	0,2071	3,2781e+04	19	0,2819	3,2781e+04
1,4	14	0,1528	3,8243e+04	14	0,2136	3,8243e+04
1,5	12	0,1318	4,4881e+04	12	0,1832	4,4881e+04
1,6	10	0,1127	5,2952e+04	10	0,1607	5,2952e+04
1,7	9	0,1146	6,2775e+04	9	0,1495	6,2775e+04
1,8	9	0,1084	7,4734e+04	9	0,1584	7,4734e+04
1,9	9	0,1216	8,9311e+04	9	0,1514	8,9311e+04

Tabela 6.1.8: Barreira Logarítmica Preditor-Corretor – Direções.

p	PDBL			PDBL(antigo)		
	it	tempo	Fo	it	tempo	Fo
1,1	33	0,4169	2,4551e+04	33	0,4288	2,4551e+04
1,2	27	0,3264	2,8274e+04	27	0,3473	2,8274e+04
1,3	22	0,2774	3,2781e+04	22	0,3013	3,2781e+04
1,4	20	0,2467	3,8243e+04	20	0,2616	3,8243e+04
1,5	18	0,2342	4,4881e+04	18	0,2543	4,4881e+04
1,6	18	0,2353	5,2952e+04	18	0,2464	5,2952e+04
1,7	18	0,2435	6,2775e+04	18	0,2497	6,2775e+04
1,8	18	0,1941	7,4734e+04	18	0,2566	7,4734e+04
1,9	19	0,2046	8,9311e+04	19	0,2193	8,9311e+04

Tabela 6.1.9: Primal-Dual Barreira Logarítmica – Direções.

p	PDBLPC			PDBLPC(antigo)		
	it	tempo	Fo	it	tempo	Fo
1,1	1	0,0357	2,6118e+04	1	0,0658	2,6118e+04
1,2	1	0,0369	2,9726e+04	1	0,0502	2,9726e+04
1,3	1	0,0401	3,4090e+04	1	0,0545	3,4090e+04
1,4	1	0,0380	3,9382e+04	1	0,0536	3,9382e+04
1,5	1	0,0481	4,5816e+04	1	0,0537	4,5816e+04
1,6	1	0,0390	5,3658e+04	1	0,0540	5,3658e+04
1,7	1	0,0433	6,3239e+04	1	0,0535	6,3239e+04
1,8	1	0,0382	7,4975e+04	1	0,0529	7,4975e+04
1,9	1	0,0386	8,9381e+04	1	0,0518	8,9381e+04

Tabela 6.1.10: Primal-Dual Barreira Logarítmica Preditor-Corretor – Direções.

Com a modificação no cálculo das direções, conseguimos diminuir o número de operações necessárias em cada iteração. Podemos visualizar isso através do número de operações de pontos flutuantes obtidos pela função *flops* do MATLAB 5.2. Veja abaixo os resultados do número de *flops* obtido em uma iteração para o problema de grande porte:

MÉTODO	FLOPS	MÉTODO	FLOPS
BL(antigo)	1.519.590	BL	1.383.964
BLPC(antigo)	2.180.059	BLPC	1.953.801
PDBL(antigo)	1.859.292	PDBL	1.834.092
PDBLPC(antigo)	2.549.677	PDBLPC	2.535.425

Em todos os métodos podemos verificar uma diminuição no número de operações. Isso faz com que o esforço computacional seja menor, contribuindo para a implementação eficiente e também faz com que os métodos fiquem mais rápidos.

6.1.4 Resultados Finais

Agora vamos unir todas as modificações apresentadas e comparar com os métodos já existentes.

p	BL			BL(antigo)		
	it	tempo	Fo	it	tempo	Fo
1,1	25	0,2035	2,4550e+04	4	0,2581	2,6040e+04
1,2	19	0,2000	2,8274e+04	24	0,2855	2,8274e+04
1,3	13	0,1445	3,2781e+04	19	0,2320	3,2781e+04
1,4	10	0,1159	3,8243e+04	16	0,2006	3,8243e+04
1,5	9	0,0982	4,4881e+04	14	0,1838	4,4881e+04
1,6	8	0,0973	5,2952e+04	13	0,1539	5,2952e+04
1,7	8	0,0955	6,2775e+04	12	0,1577	6,2775e+04
1,8	8	0,0950	7,4734e+04	11	0,1554	7,4734e+04
1,9	7	0,0814	8,9311e+04	9	0,1335	8,9311e+04

Tabela 6.1.11: Barreira Logarítmica.

p	BLPC			BLPC(antigo)		
	it	tempo	Fo	it	tempo	Fo
1,1	5	0,1139	2,4705e+04	6	0,1325	2,6040e+04
1,2	8	0,1512	2,8274e+04	26	0,3702	2,8274e+04
1,3	6	0,1179	3,2781e+04	19	0,2819	3,2781e+04
1,4	5	0,1002	3,8243e+04	14	0,2136	3,8243e+04
1,5	4	0,0766	4,4881e+04	12	0,1832	4,4881e+04
1,6	4	0,0832	5,2952e+04	10	0,1607	5,2952e+04
1,7	3	0,0679	6,2775e+04	9	0,1495	6,2775e+04
1,8	3	0,0668	7,4734e+04	9	0,1584	7,4734e+04
1,9	3	0,0674	8,9311e+04	9	0,1514	8,9311e+04

Tabela 6.1.12: Barreira Logarítmica Preditor-Corretor.

p	PDBL			PDBL(antigo)		
	it	tempo	Fo	it	tempo	Fo
1,1	20	0,3038	2,4550e+04	33	0,4288	2,4551e+04
1,2	16	0,2455	2,8274e+04	27	0,3473	2,8274e+04
1,3	13	0,2037	3,2781e+04	22	0,3013	3,2781e+04
1,4	11	0,1828	3,8243e+04	20	0,2616	3,8243e+04
1,5	10	0,1704	4,4881e+04	18	0,2543	4,4881e+04
1,6	9	0,1553	5,2952e+04	18	0,2464	5,2952e+04
1,7	9	0,1518	6,2775e+04	18	0,2497	6,2775e+04
1,8	9	0,1543	7,4734e+04	18	0,2566	7,4734e+04
1,9	10	0,1633	8,9311e+04	19	0,2193	8,9311e+04

Tabela 6.1.13: Primal-Dual Barreira Logarítmica.

p	PDBLPC			PDBLPC(antigo)		
	it	tempo	Fo	it	tempo	Fo
1,1	11	0,1858	2,4550e+04	1	0,0658	2,6118e+04
1,2	11	0,1895	2,8274e+04	1	0,0502	2,9726e+04
1,3	7	0,1395	3,2781e+04	1	0,0545	3,4090e+04
1,4	7	0,1397	3,8243e+04	1	0,0536	3,9382e+04
1,5	6	0,1123	4,4881e+04	1	0,0537	4,5816e+04
1,6	6	0,1231	5,2952e+04	1	0,0540	5,3658e+04
1,7	6	0,1222	6,2775e+04	1	0,0535	6,3239e+04
1,8	13	0,1320	7,4734e+04	1	0,0529	7,4975e+04
1,9	8	0,1488	8,9311e+04	1	0,0518	8,9381e+04

Tabela 6.1.14: Primal-Dual Barreira Logarítmica Predictor-Corretor.

Percebemos uma redução no número de iterações e no valor da função objetivo, exceto para o método primal-dual barreira logarítmica preditor-corretor, onde o número de iterações aumentou, mas o valor da função objetivo diminuiu. Nas tabelas 6.1.11 e 6.1.13 verificamos a diminuição do tempo, enquanto que nas tabelas 6.1.12 e 6.1.14 o tempo aumentou por causa da nova forma de calcular os resíduos.

Na Tabela 6.1.11 para $p = 1, 1$ o número de iterações aumentou, porém o valor da função objetivo foi menor. Para os outros valores de p , houve redução do número de iterações e, conseqüentemente, do tempo também. Nas Tabelas 6.1.12 e 6.1.13 observamos redução no número de iterações. Na Tabela 6.1.14 houve um aumento no número de iterações, mas uma redução no valor da função objetivo.

Considere o caso em que resolvemos esse problema utilizando o método primal-dual preditor-corretor, com $p = 1, 5$. O resultado encontrado foi $x^* = \begin{bmatrix} 4, 2080 \\ 6, 2531 \end{bmatrix}$. Lembrando que x^* representa os coeficientes de um polinômio, nesse caso de grau 1, então $p(t) = 6, 2531t + 4, 2080$.

Graficamente:

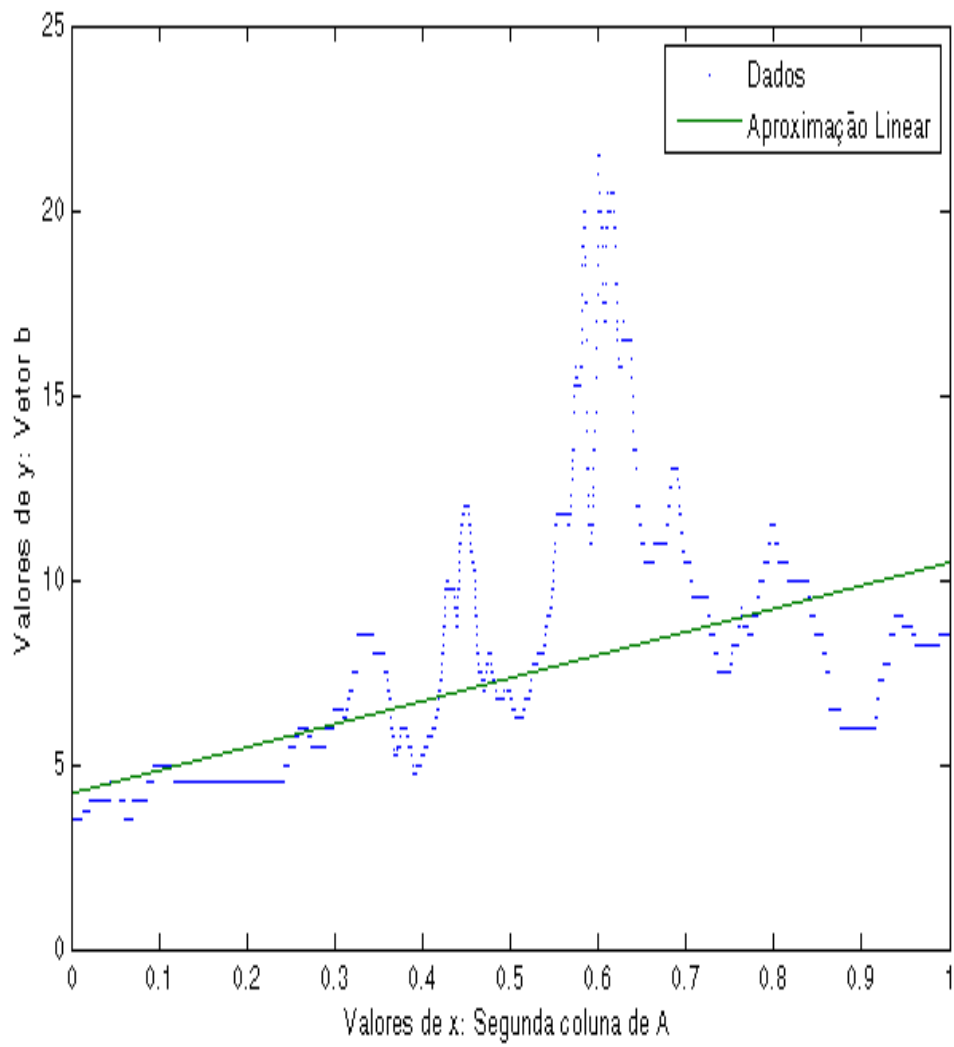


Figura 6.1: Problema de Grande Porte

Na Figura 6.1 podemos visualizar a aproximação linear obtida para o problema. Os valores de y representam a taxa de juros, e os valores de x representam as datas que foram normalizadas no intervalo $[0,1]$.

6.2 Função Cosseno

Deste exemplo em diante utilizaremos os métodos descritos no Capítulo 4, ou seja, os métodos modificados, para resolver os problemas.

Esse problema contém 20001 pontos. A segunda coluna da matriz A é formada por elementos no intervalo $[0, 2\pi]$, o vetor b consiste no cosseno desses valores. Vamos utilizar aproximação por um polinômio de grau 1, ou seja, aproximação por uma reta. Utilizamos $\mu = 0,001$, $\tau = 0,99995$, $\beta = 10$, $\epsilon = 10^{-10}$, $\epsilon_1 = 10^{-8}$, onde ϵ e ϵ_1 são usados para a verificação do critério de convergência, e $\sigma = 10^{-10}$.

p	BL			BL(antigo)		
	it	tempo	Fo	it	tempo	Fo
1,1	2	0,0549	1,2355e+004	11	0,2111	1,2355e+004
1,2	3	0,0699	1,2011e+004	12	0,2202	1,2011e+004
1,3	3	0,0714	1,1693e+004	8	0,1721	1,1693e+004
1,4	3	0,0769	1,1399e+004	6	0,1389	1,1399e+004
1,5	4	0,0092	1,1125e+004	6	0,1381	1,1125e+004
1,6	5	0,0999	1,0869e+004	6	0,1419	1,0869e+004
1,7	6	0,1163	1,0630e+004	6	0,1395	1,0630e+004
1,8	6	0,1132	1,0406e+004	5	0,1214	1,0406e+004
1,9	6	0,1343	1,0195e+004	6	0,1380	1,0195e+004

Tabela 6.2.1: Barreira Logarítmica – Função Cosseno.

p	BLPC			BLPC(antigo)		
	it	tempo	Fo	it	tempo	Fo
1,1	2	0,1044	1,2355e+004	2	0,1020	1,2355e+004
1,2	4	0,1780	1,2011e+004	13	0,3024	1,2011e+004
1,3	3	0,1438	1,693e+004	10	0,2450	1,1693e+004
1,4	3	0,1436	1,1399e+004	8	0,2114	1,1399e+004
1,5	3	0,1332	1,1125e+004	7	0,1960	1,1125e+004
1,6	4	0,1748	1,0869e+004	6	0,1687	1,0869e+004
1,7	4	0,1720	1,0630e+004	6	0,1748	1,0630e+004
1,8	5	0,2152	1,0406e+004	6	0,1702	1,0406e+004
1,9	6	0,2422	1,0195e+004	6	0,1748	1,0195e+004

Tabela 6.2.2: Barreira Logarítmica Preditor-Corretor – Função Cosseno.

p	PDBL			PDBL(antigo)		
	it	tempo	Fo	it	tempo	Fo
1,1	1	0,0670	1,2355e+004	8	0,1874	1,2355e+004
1,2	1	0,0667	1,2011e+004	6	0,1519	1,2011e+004
1,3	1	0,0657	1,1693e+004	6	0,1529	1,1693e+004
1,4	1	0,0664	1,1399e+004	6	0,1537	1,1399e+004
1,5	1	0,0668	1,1125e+004	6	0,1497	1,1125e+004
1,6	1	0,0658	1,0869e+004	6	0,1533	1,0869e+004
1,7	1	0,0667	1,0630e+004	6	0,1485	1,0630e+004
1,8	1	0,0643	1,0406e+004	6	0,1523	1,0406e+004
1,9	1	0,0649	1,0195e+004	6	0,1512	1,0195e+004

Tabela 6.2.3: Primal-Dual Barreira Logarítmica – Função Cosseno.

p	PDBLPC			PDBLPC(antigo)		
	it	tempo	Fo	it	tempo	Fo
1,1	1	0,1062	1,2355e+004	3	0,1138	1,2355e+004
1,2	1	0,1120	1,2011e+004	3	0,1116	1,2011e+004
1,3	1	0,1088	1,1693e+004	3	0,1058	1,1693e+004
1,4	1	0,1055	1,1399e+004	4	0,1373	1,1399e+004
1,5	1	0,1042	1,1125e+004	4	0,1320	1,1125e+004
1,6	1	0,1076	1,0869e+004	4	0,1370	1,0869e+004
1,7	1	0,1075	1,0630e+004	4	0,1262	1,0630e+004
1,8	1	0,1062	1,0406e+004	4	0,1336	1,0406e+004
1,9	1	0,1076	1,0195e+004	4	0,1343	1,0195e+004

Tabela 6.2.4: Primal-Dual Barreira Logarítmica Preditor-Corretor – função cosseno.

Analisando as tabelas percebemos uma redução no número de iterações e no tempo, o valor da função objetivo não se alterou.

Vamos considerar o caso em que $p = 1,9$ e o método primal-dual barreira logarítmica para observar graficamente o resultado. Nesse caso obtivemos $x^* = \begin{bmatrix} 0,0016 \\ -0,0006 \end{bmatrix}$.

Temos como resultado uma aproximação linear dada pelo polinômio $p(t) = -0,0006t + 0,0016$.

Graficamente:

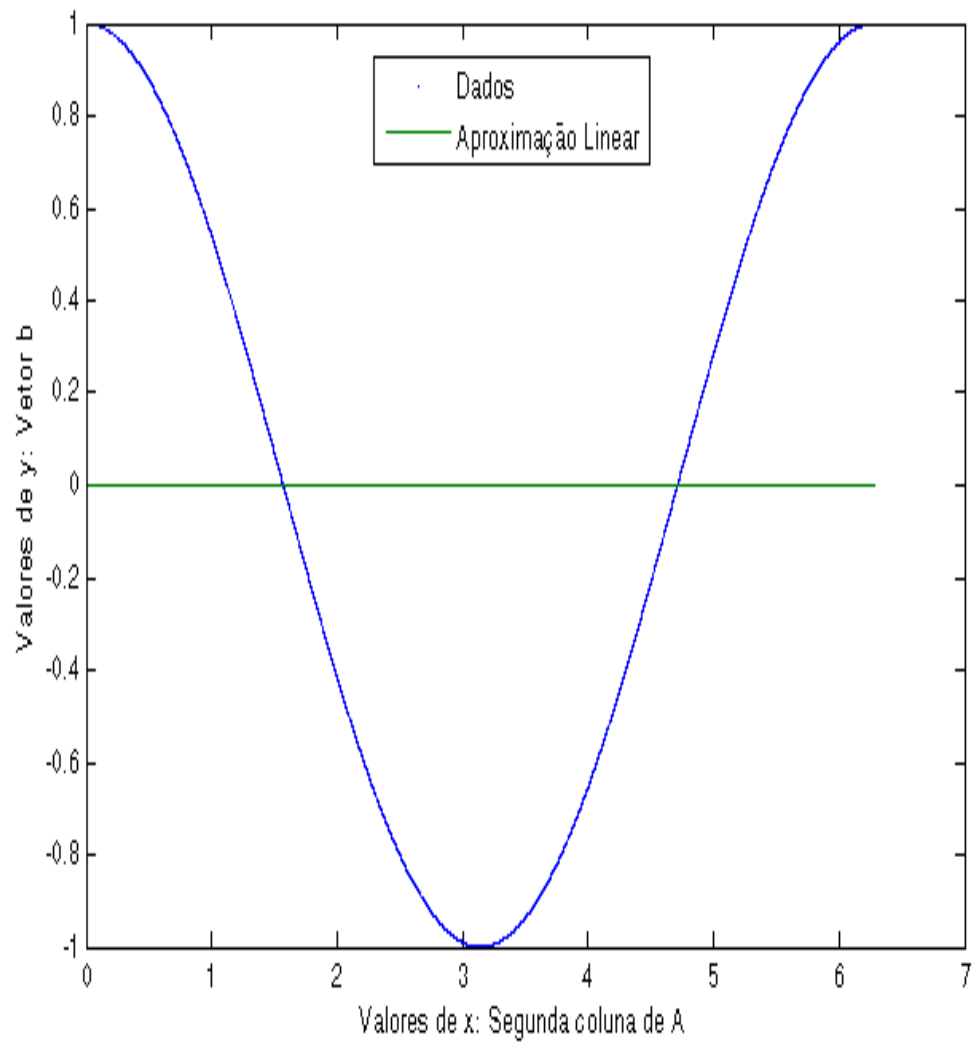


Figura 6.2: Função Cosseno: Aproximação Linear

Observando a Figura 6.2 vemos a aproximação linear para a função cosseno. Os valores de x estão no intervalo $[0, 2\pi]$ mostrados no gráfico em radianos.

Consideremos agora o caso em que $p = 1,9$ e o método primal-dual barreira logarítmica para observar graficamente o resultado quando fazemos aproximação por polinômio de

grau 2. Nesse caso obtivemos $x^* = \begin{bmatrix} 1,5209 \\ -1,4525 \\ 0,2312 \end{bmatrix}$.

Temos como resultado uma aproximação quadrática dada pelo polinômio $p(t) = 0,2312t^2 - 1,4525t + 1,5209$.

Graficamente:

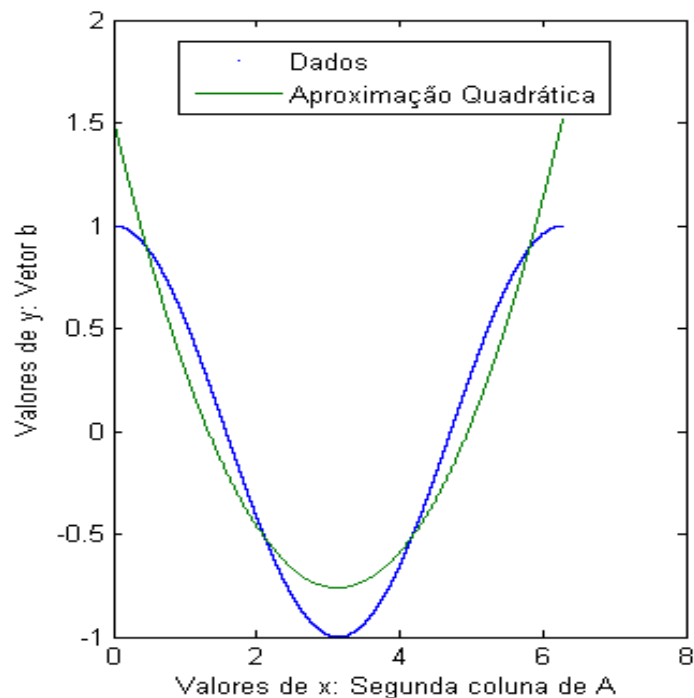


Figura 6.3: Função Cosseno: Aproximação Quadrática

6.3 Função Logaritmo

Esse problema contém 15 mil pontos, os valores da segunda coluna de A são obtidos dividindo-se o intervalo $[1,4]$ em 15 mil. O vetor b é o valor do logaritmo neperiano do valor correspondente à segunda coluna de A .

Obtivemos a matriz $A_{15000 \times 2}$ e o vetor b . Vamos aproximar a função logaritmo por um polinômio de grau 1. Utilizamos $\mu = 0,001$, $\tau = 0,99995$, $\beta = 10$, $\epsilon = 10^{-10}$, $\epsilon_1 = 10^{-8}$, onde ϵ e ϵ_1 são usados para a verificação do critério de convergência, e $\sigma = 8.10^{-2}$.

p	BL			BL(antigo)		
	it	tempo	Fo	it	tempo	Fo
1,1	15	0,1655	607,9389	2	0,0510	635,1600
1,2	2	0,0382	483,1952	3	0,0612	485,3428
1,3	2	0,0381	372,2146	2	0,0511	372,4476
1,4	2	0,0383	287,5839	2	0,0507	287,4594
1,5	2	0,0389	222,8549	2	0,0501	222,6275
1,6	2	0,0381	173,2170	2	0,0507	173,0235
1,7	3	0,0491	135,0870	2	0,0522	134,9931
1,8	2	0,0378	105,6851	2	0,0510	105,9290
1,9	2	0,0377	83,0445	6	0,0929	83,9296

Tabela 6.3.1: Barreira Logarítmica – Função Logaritmo.

p	BLPC			BLPC(antigo)		
	it	tempo	Fo	it	tempo	Fo
1,1	1	0,0558	632,6171	1	0,0586	635,4659
1,2	2	0,0731	484,8184	3	0,0773	486,5972
1,3	3	0,0970	372,9440	3	0,0771	372,5131
1,4	2	0,0740	287,5702	3	0,0784	287,5263
1,5	3	0,0895	222,7532	2	0,0653	222,6095
1,6	2	0,0734	173,1270	2	0,0652	172,9877
1,7	2	0,0732	135,0390	2	0,0654	134,9850
1,8	2	0,0751	105,7751	2	0,0653	105,9557
1,9	2	0,0731	83,3012	4	0,0909	83,2252

Tabela 6.3.2: Barreira Logarítmica Preditor-Corretor – Função Logaritmo.

p	PDBL			PDBL(antigo)		
	it	tempo	Fo	it	tempo	Fo
1,1	13	0,2073	607,8013	24	0,3377	607,9149
1,2	11	0,1832	470,5602	22	0,3035	470,5663
1,3	10	0,1618	365,1753	19	0,2652	365,1758
1,4	9	0,1524	283,9891	17	0,2397	283,9891
1,5	9	0,1513	221,2673	16	0,2276	221,2673
1,6	8	0,1378	172,6894	15	0,2149	172,6894
1,7	8	0,1402	134,9821	14	0,2055	134,9821
1,8	7	0,1272	105,6547	13	0,1926	105,6547
1,9	6	0,1133	82,8040	11	0,1664	82,8040

Tabela 6.3.3: Primal-Dual Barreira Logarítmica – Função Logaritmo.

p	PDBLPC			PDBLPC(antigo)		
	it	tempo	Fo	it	tempo	Fo
1,1	7	0,2110	607,9596	1	0,0520	635,5307
1,2	9	0,2478	470,8493	1	0,0450	487,9858
1,3	11	0,2796	365,2191	1	0,0461	375,7753
1,4	6	0,1910	284,0064	1	0,0455	290,1729
1,5	5	0,1539	221,2803	1	0,0454	224,6762
1,6	5	0,1668	172,6981	1	0,0455	174,4212
1,7	5	0,1674	134,9873	1	0,0447	135,7555
1,8	4	0,1488	105,6571	10	0,1908	105,6571
1,9	5	0,1683	82,8040	9	0,1704	82,8040

Tabela 6.3.4: Primal-Dual Barreira Logarítmica Preditor-Corretor – Função Logaritmo.

Na Tabela 6.3.1 vemos que para $p = 1,1$ o número de iterações aumentou, mas o valor da função objetivo reduziu. Na Tabela 6.3.3 notamos a redução no número de iterações. Na Tabela 6.3.4, onde o número de iterações está maior percebemos que o valor da função objetivo é menor.

Vamos considerar o caso em que $p = 1,9$ e o método primal-dual barreira logarítmica para observar graficamente o resultado. Nesse caso obtivemos $x^* = \begin{bmatrix} -0,2330 \\ 0,4331 \end{bmatrix}$.

Temos como resultado uma aproximação linear dada pelo polinômio $p(t) = 0,4331t - 0,2330$.

Graficamente:

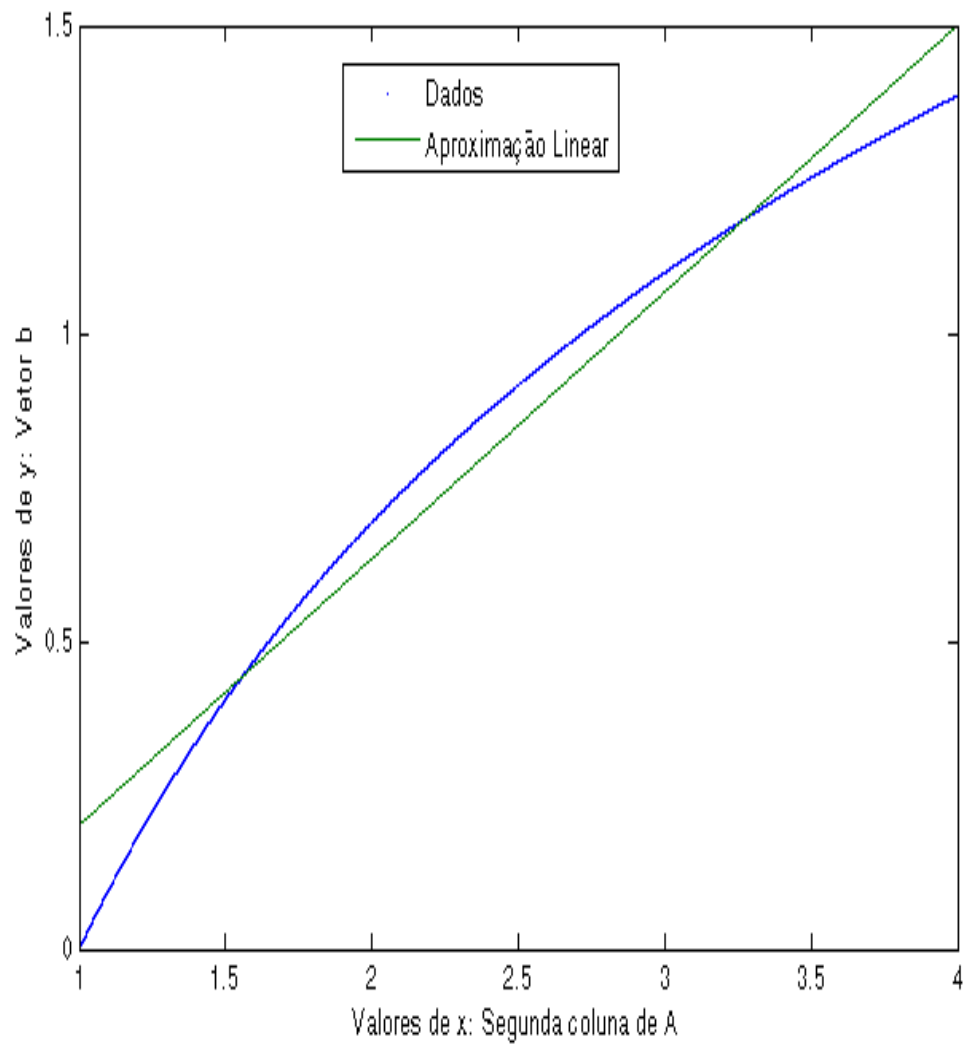


Figura 6.4: Função logaritmo: Aproximação Linear

Na Figura 6.4 temos uma aproximação linear para a função logaritmo no intervalo $[1,4]$.

6.4 Função Seno Hiperbólico

Esse problema contém 40 mil pontos, os valores da segunda coluna de A são obtidos dividindo-se o intervalo $[-2,2]$ em 40 mil. O vetor b é o valor do seno hiperbólico do valor correspondente na segunda coluna de A . Lembramos que $\sinh(x) = \frac{e^x - e^{-x}}{2}$.

Obtivemos a matriz $A_{40000 \times 2}$ e o vetor b . Vamos aproximar a função seno hiperbólico por um polinômio de grau 1. Utilizamos $\mu = 0,001$, $\tau = 0,99995$, $\beta = 10$, $\epsilon = 10^{-10}$, $\epsilon_1 = 10^{-8}$, onde ϵ e ϵ_1 são usados para a verificação do critério de convergência, e $\sigma = 10^{-1}$.

p	BL			BL(antigo)		
	it	tempo	Fo	it	tempo	Fo
1,1	5	0,2037	7,4691e+003	1	0,1233	7,4721e+003
1,2	5	0,2098	6,5534e+003	1	0,1201	6,5561e+003
1,3	5	0,2068	5,7622e+003	1	0,1203	5,7673e+003
1,4	2	0,1151	5,0782e+003	5	0,2502	5,0635e+003
1,5	2	0,1159	4,4674e+003	4	0,2149	4,4664e+003
1,6	3	0,1465	3,9556e+003	7	0,3151	3,9568e+003
1,7	3	0,1453	3,5238e+003	5	0,2478	3,5253e+003
1,8	5	0,2089	3,1472e+003	6	0,2799	3,1471e+003
1,9	7	0,2629	2,8141e+003	8	0,3382	2,8141e+003

Tabela 6.4.1: Barreira Logarítmica – Função Seno Hiperbólico.

p	BLPC			BLPC(antigo)		
	it	tempo	Fo	it	tempo	Fo
1,1	2	0,2289	7,4725e+003	1	0,1640	7,4721e+003
1,2	2	0,2202	6,5565e+003	1	0,1571	6,5561e+003
1,3	2	0,2156	5,7681e+003	17	0,7589	5,6138e+003
1,4	2	0,2178	5,0873e+003	3	0,2321	5,0691e+003
1,5	2	0,2056	4,4953e+003	10	0,5064	4,4339e+003
1,6	2	0,2239	3,9709e+003	4	0,2705	3,9632e+003
1,7	4	0,3671	3,6159e+003	10	0,4992	3,5244e+003
1,8	6	0,5069	3,1489e+003	6	0,5109	3,1489e+003
1,9	6	0,5073	2,8141e+003	6	0,5032	2,8141e+003

Tabela 6.4.2: Barreira Logarítmica Preditor-Corretor – Função Seno Hiperbólico.

p	PDBL			PDBL(antigo)		
	it	tempo	Fo	it	tempo	Fo
1,1	13	0,5840	7,1614e+003	1	0,1534	7,4722e+003
1,2	10	0,4750	6,3338e+003	1	0,1517	6,5562e+003
1,3	9	0,4321	5,6138e+003	17	0,6981	5,6138e+003
1,4	8	0,3998	4,9849e+003	3	0,2236	5,0872e+003
1,5	8	0,4015	4,4339e+003	10	0,4627	4,4339e+003
1,6	8	0,4020	3,4999e+003	4	0,2542	3,9632e+003
1,7	8	0,3986	3,5235e+003	10	0,4572	3,5244e+003
1,8	8	0,3974	3,1471e+003	6	0,3243	3,1471e+003
1,9	8	0,3962	2,8141e+003	6	0,3248	2,8141e+003

Tabela 6.4.3: Primal-Dual Barreira Logarítmica – Função Seno Hiperbólico.

p	PDBLPC			PDBLPC(antigo)		
	it	tempo	Fo	it	tempo	Fo
1,1	1	0,2214	7,4725e+003	1	0,1441	7,4725e+003
1,2	1	0,2351	6,5565e+003	1	0,1308	6,5565e+003
1,3	1	0,2117	5,7681e+003	1	0,1299	5,7681e+003
1,4	1	0,2186	5,0875e+003	1	0,1280	5,0875e+003
1,5	1	0,2080	4,4984e+003	1	0,1292	4,4985e+003
1,6	1	0,2149	3,9872e+003	1	0,1288	3,9872e+003
1,7	1	0,2144	3,5425e+003	1	0,1282	3,5425e+003
1,8	1	0,2155	3,1547e+003	1	0,1290	3,1547e+003
1,9	1	0,2161	2,8158e+003	1	0,1280	2,8158e+003

Tabela 6.4.4: Primal-Dual Barreira Logarítmica Preditor-Corretor – Função Seno Hiperbólico.

Na tabela 6.4.1, onde o número de iterações aumentou notamos redução no valor da função objetivo. Na tabela 6.4.2 podemos observar diminuição no número de iterações. Na Tabela 6.4.4 o tempo aumentou, isso acontece por causa da forma de calcular os resíduos.

Vamos considerar o caso em que $p = 1,9$ e o método primal-dual barreira logarítmica para observar graficamente o resultado. Nesse caso obtivemos $x^* = \begin{bmatrix} 0 \\ 1,4615 \end{bmatrix}$.

Temos como resultado uma aproximação linear dada pelo polinômio $p(t) = 1,4615t$.

Graficamente:

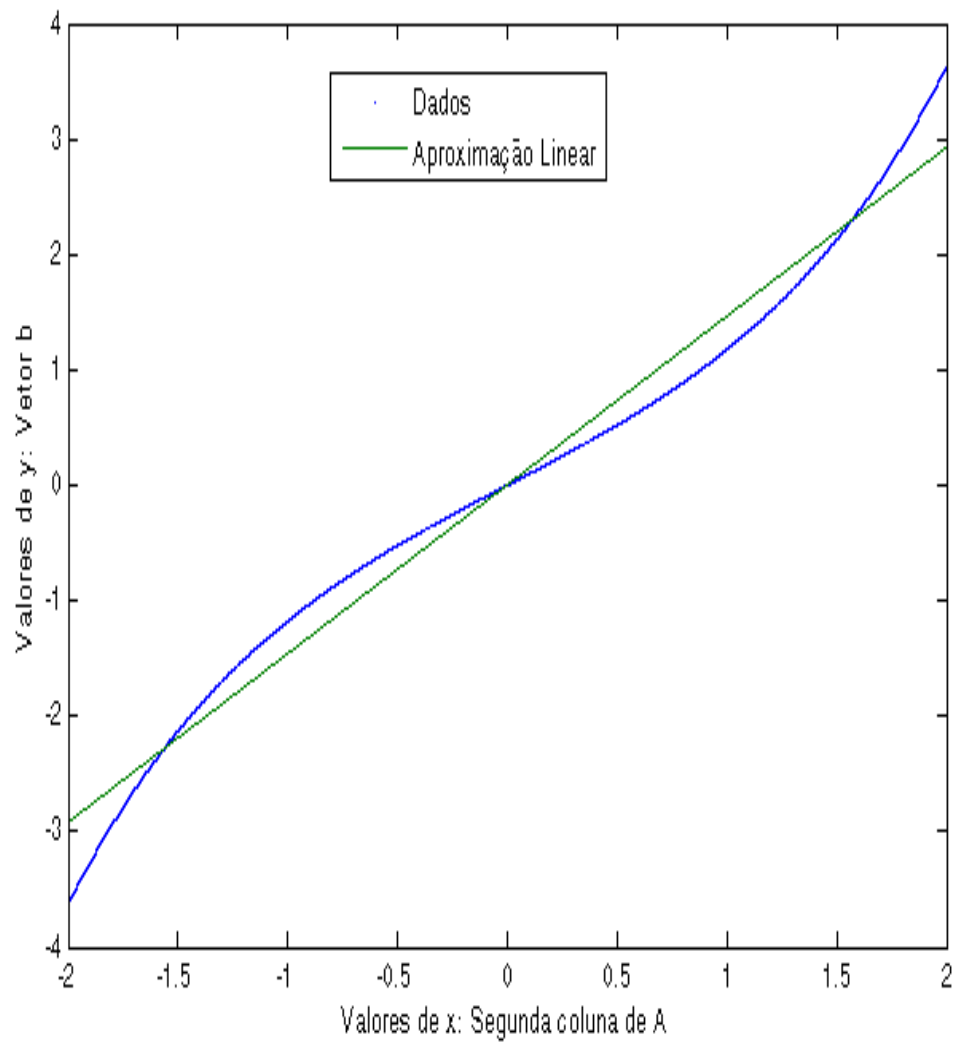


Figura 6.5: Função Seno Hiperbólico: Aproximação Linear

A Figura 6.5 mostra a aproximação linear obtida através do método de pontos interiores para a função seno hiperbólico.

6.5 Implementação Eficiente em C

Apresentaremos agora os resultados computacionais da implementação eficiente em C para resolver problemas de grande porte.

6.5.1 Problema de Grande Porte

Esse problema que contém 109570 observações e foi obtido a partir do problema de grande porte citado na Seção 6.1. Utilizamos $\mu = 0,001$, $\tau = 0,99995$, $\beta = 10$, $\epsilon = 10^{-10}$, $\epsilon_1 = 10^{-8}$, onde ϵ e ϵ_1 são usados para a verificação do critério de convergência, e $\sigma = 10^{-10}$.

Aproximação por Reta

p	BL			BLPC		
	it	tempo	Fo	it	tempo	Fo
1,1	18	1,010	2,5640e+05	4	0,4400	2,5640e+05
1,2	21	1,1300	2,9067e+05	3	0,3400	2,9068e+05
1,3	2	0,2200	3,3194e+05	4	0,4400	3,3194e+05
1,4	2	0,2400	3,8173e+05	2	0,2900	3,8173e+05
1,5	4	0,3200	5,2823e+05	2	0,2900	4,4195e+05
1,6	4	0,3300	6,5216e+05	2	0,2800	5,1499e+05
1,7	4	0,3300	9,1413e+05	2	0,2900	6,0379e+05
1,8	3	0,2800	8,8434e+05	2	0,2900	7,1203e+05
1,9	4	0,3300	1,0320e+06	2	0,2900	8,4430e+05

Tabela 6.5.1: BL e BLPC – Aproximação Linear.

p	PDBL			PDBLPC		
	it	tempo	Fo	it	tempo	Fo
1,1	2	0,3100	2,5640e+05	2	0,3200	2,5640e+05
1,2	2	0,3200	2,9068e+05	2	0,3200	2,9068e+05
1,3	2	0,3100	3,3194e+05	2	0,3200	3,3194e+05
1,4	2	0,2700	3,8173e+05	2	0,3200	3,8173e+05
1,5	2	0,2700	4,4195e+05	2	0,3200	4,4195e+05
1,6	2	0,2700	5,1499e+05	2	0,3200	5,1499e+05
1,7	2	0,3200	6,0379e+05	2	0,3200	6,0379e+05
1,8	2	0,3100	7,1203e+05	2	0,3300	7,1203e+05
1,9	2	0,3100	8,4430e+05	2	0,3300	8,4430e+05

Tabela 6.5.2: PDBL e PDBLPC – Aproximação Linear.

Na Tabela 6.5.1 vemos a redução no número de iterações no método barreira logarítmica preditor-corretor e mesmo o problema sendo grande, o tempo está sendo pequeno. Na Tabela 6.5.2 o número de iterações é o mesmo para ambos os métodos. O tempo é levemente menor no método primal-dual barreira logarítmica, isso acontece porque no método primal-dual barreira logarítmica preditor-corretor temos o cálculo dos resíduos.

Vamos considerar o caso em que $p = 1,5$ e o método primal-dual barreira logarítmica para observar graficamente o resultado. Nesse caso obtivemos $x^* = \begin{bmatrix} 4,825866 \\ 5,770012 \end{bmatrix}$.

Temos como resultado uma aproximação linear dada pelo polinômio $p(t) = 5,770012t + 4,825866$.

Graficamente:

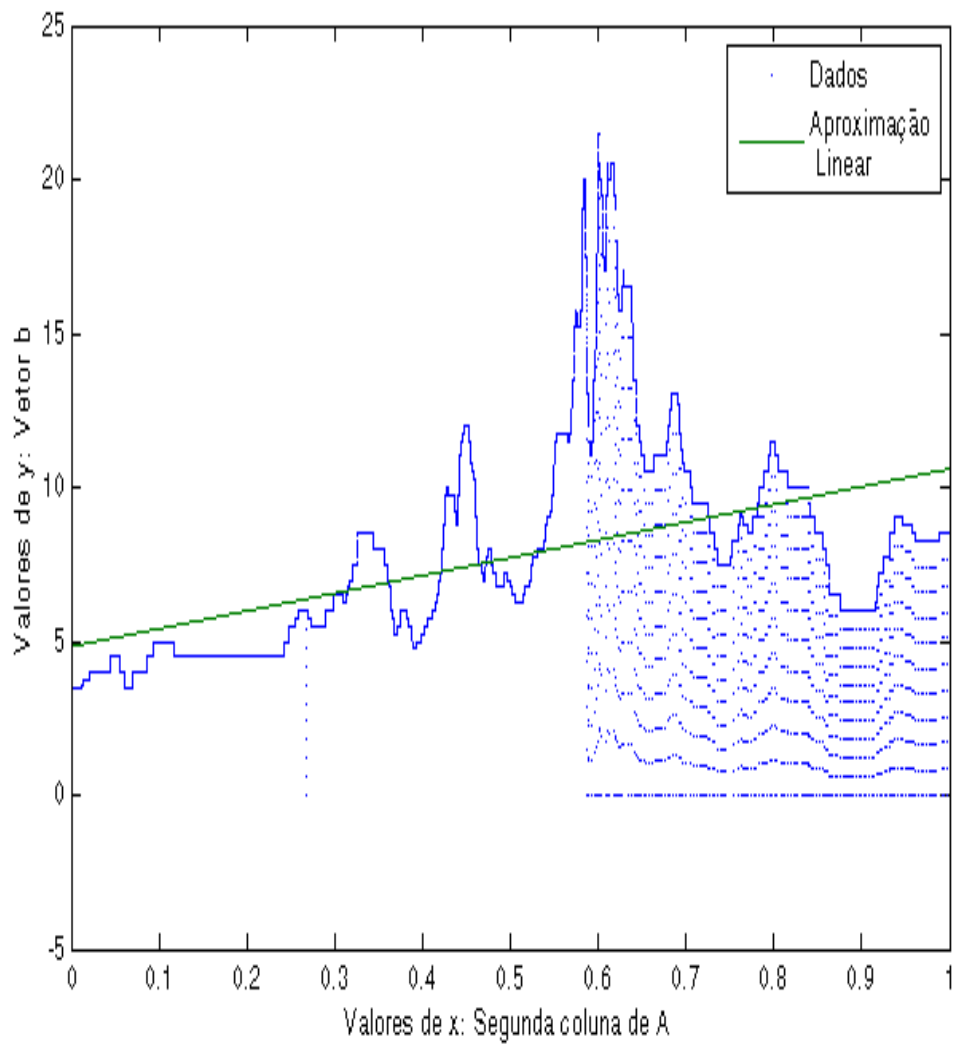


Figura 6.6: Aproximação por Reta

Na Figura 6.6 temos uma aproximação linear para esse problema no intervalo $[0,1]$.

Aproximação por Polinômio de grau 8

p	BL			BLPC		
	it	tempo	Fo	it	tempo	Fo
1,1	2	0,3100	1,7511e+05	2	0,3900	1,7511e+05
1,2	2	0,3200	1,9302e+05	2	0,3900	1,9302e+05
1,3	2	0,3100	2,1414e+05	2	0,3900	2,1414e+05
1,4	2	0,3000	2,3902e+05	2	0,3900	2,3902e+05
1,5	2	0,3100	2,6834e+05	2	0,3900	2,6834e+05
1,6	2	0,3100	3,0293e+05	2	0,3700	3,0293e+05
1,7	2	0,3000	3,4379e+05	2	0,3900	3,4379e+05
1,8	2	0,3200	3,9214e+05	2	0,3900	3,9214e+05
1,9	2	0,3100	4,4946e+05	2	0,3800	4,4946e+05

Tabela 6.5.3: BL e BLPC – Polinômio de Grau 8.

p	PDBL			PDBLPC		
	it	tempo	Fo	it	tempo	Fo
1,1	2	0,4200	1,7511e+05	2	0,4200	1,7511e+05
1,2	2	0,4100	1,9302e+05	1	0,3500	1,9302e+05
1,3	2	0,4100	2,1414e+05	2	0,4300	2,1414e+05
1,4	2	0,4200	2,3902e+05	2	0,4200	2,3902e+05
1,5	2	0,4100	2,6834e+05	2	0,4000	2,6834e+05
1,6	2	0,4100	3,0293e+05	2	0,4200	3,0293e+05
1,7	2	0,4100	3,4379e+05	2	0,4300	3,4379e+05
1,8	2	0,3500	3,9214e+05	2	0,4200	3,9214e+05
1,9	2	0,4200	4,4946e+05	2	0,4400	4,4946e+05

Tabela 6.5.4: PDBL e PDBLPC – Polinômio de Grau 8.

Nas Tabelas 6.5.3 e 6.5.4 acontece algo parecido com a Tabela 6.5.2, onde o número de iterações é o mesmo e o tempo é menor nos métodos barreira logarítmica e primal-dual barreira logarítmica. A explicação para esse fato é a mesma que foi descrita sobre a Tabela 6.5.2.

Vamos considerar o caso em que $p = 1, 5$ e o método primal-dual barreira logarítmica para observar graficamente o resultado. Nesse caso obtivemos

$$x^* = \begin{bmatrix} 3,467644 \\ 29,391207 \\ -541,218631 \\ 5099,439627 \\ -23844,731862 \\ 60182,576581 \\ -82228,738047 \\ 56945,331080 \\ -15638,339315 \end{bmatrix} .$$

Temos como resultado uma aproximação de grau 8 dada pelo polinômio $p(t) = -15638,339315t^8 + 56945,331080t^7 - 82228,738047t^6 + 60182,576581t^5 - 23844,731862t^4 + 5099,439627t^3 - 541,218631t^2 + 29,391207t + 3,467644$.

Graficamente:

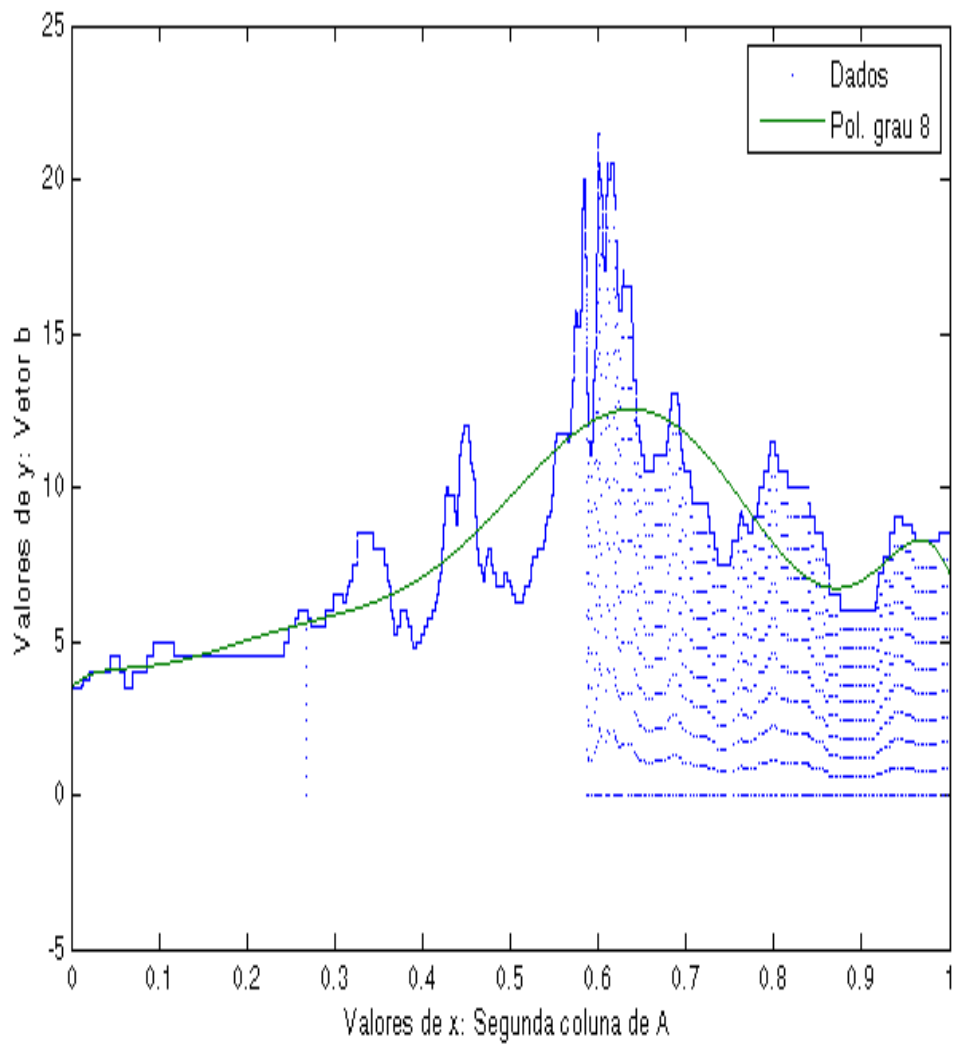


Figura 6.7: Problema de Grande Porte: Aproximação por Polinômio de Grau 8

Observe que a Figura 6.7 é uma aproximação por polinômio de grau 8 para o problema, essa aproximação é melhor que a aproximação linear, quanto maior o grau do polinômio melhor será a aproximação no intervalo considerado, neste caso $I=[0,1]$.

6.5.2 Função Seno

Esse problema contém 150 mil dados. A segunda coluna da matriz A é formada por elementos no intervalo $[0, \frac{3}{2}\pi]$, o vetor b consiste no seno desses valores. Vamos utilizar aproximação por um polinômio de grau 2. Utilizamos $\mu = 0,001$, $\tau = 0,99995$, $\beta = 10$, $\epsilon = 10^{-10}$, $\epsilon_1 = 10^{-8}$, onde ϵ e ϵ_1 são usados para a verificação do critério de convergência, e $\sigma = 10^{-10}$.

p	BL			BLPC		
	it	tempo	Fo	it	tempo	Fo
1,1	12	1,0500	1,9369e+04	4	0,6600	1,9369e+04
1,2	12	1,2600	1,6417e+04	4	0,6700	1,6418e+04
1,3	3	0,4200	1,3951e+04	2	0,4400	1,3951e+04
1,4	3	0,4300	1,1882e+04	2	0,4400	1,1882e+04
1,5	2	0,3600	1,0144e+04	2	0,4500	1,0144e+04
1,6	2	0,3600	8,6791e+03	2	0,46700	8,6791e+03
1,7	2	0,3600	7,4418e+03	2	0,4400	7,4418e+03
1,8	2	0,3600	6,3946e+03	2	0,4700	6,3945e+03
1,9	3	0,4300	5,5082e+03	2	0,3800	5,5063e+03

Tabela 6.5.5: BL e BLPC.

p	PDBL			PDBLPC		
	it	tempo	Fo	it	tempo	Fo
1,1	2	0,4000	1,9369e+04	2	0,4900	1,9369e+04
1,2	2	0,3900	1,6418e+04	2	0,4800	1,6418e+04
1,3	2	0,3900	1,3951e+04	2	0,4800	1,3951e+04
1,4	2	0,3900	1,1882e+04	2	0,4800	1,1882e+04
1,5	2	0,4000	1,0144e+04	2	0,4800	1,0144e+04
1,6	2	0,3900	8,6791e+03	2	0,4900	8,6791e+03
1,7	2	0,4000	7,4418e+03	2	0,4900	7,4418e+03
1,8	2	0,4200	6,3945e+03	2	0,4900	6,3945e+03
1,9	2	0,3900	5,5063e+03	2	0,5000	5,5063e+03

Tabela 6.5.6: PDBL e PDBLPC.

Na Tabela 6.5.5 notamos a redução no número de iterações no método predictor-corretor.

Vamos considerar o caso em que $p = 1,9$ e o método primal-dual barreira logarítmica para observar graficamente o resultado. Nesse caso obtivemos

$$x^* = \begin{bmatrix} 0,254878 \\ 0,715752 \\ -0,233634 \end{bmatrix}.$$

Temos como resultado uma aproximação quadrática dada pelo polinômio $p(t) = -0,233634t^2 + 0,715752t + 0,254878$.

Graficamente:

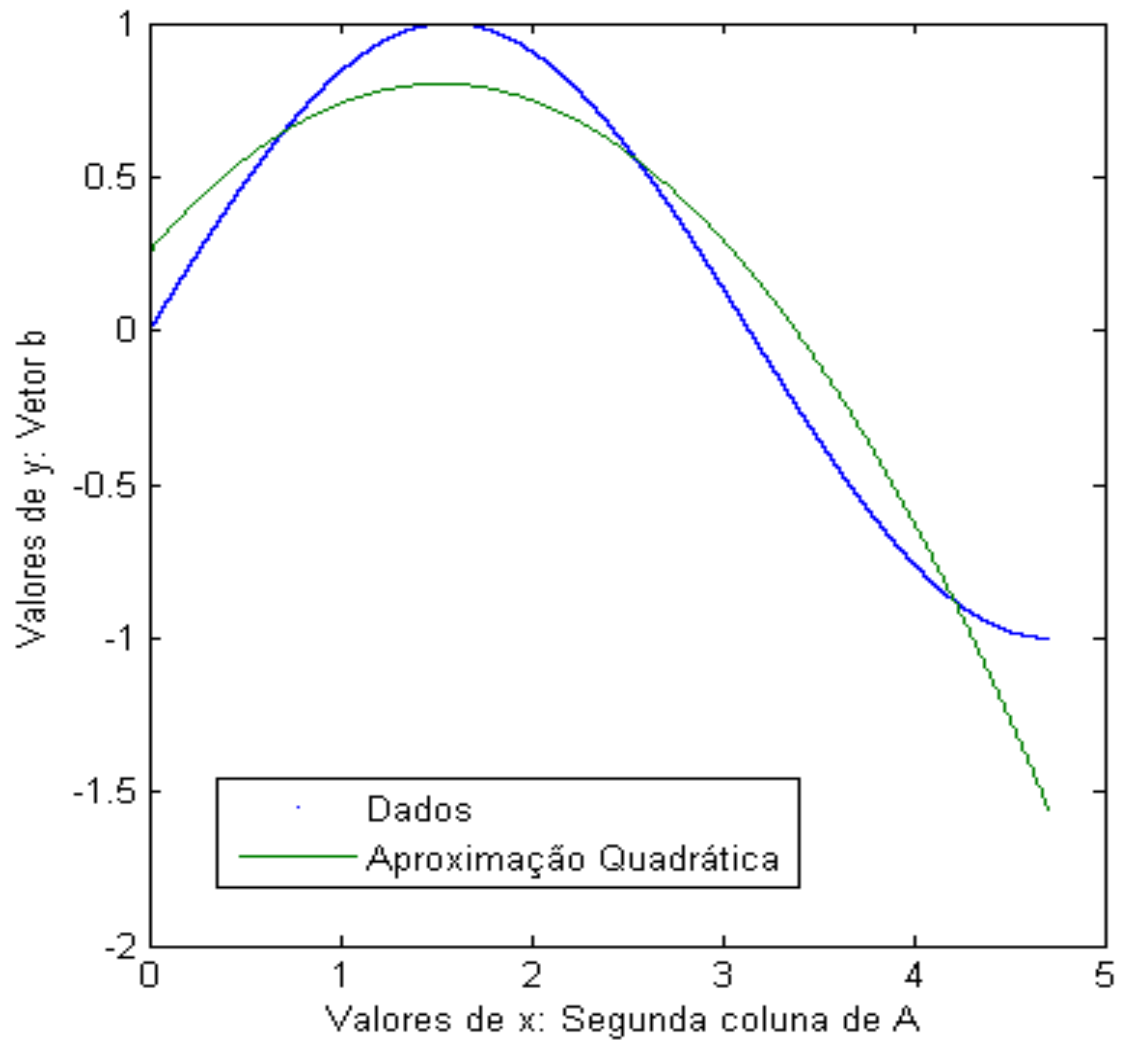


Figura 6.8: Função Seno: Aproximação Quadrática

Veja na Figura 6.8 que obtemos uma aproximação quadrática para a função seno em $[0,5]$ quando utilizamos implementação eficiente dos métodos de pontos interiores.

Capítulo 7

Conclusões

Os métodos de pontos interiores para resolver problemas de regressão polinomial apresentam melhor desempenho e menor esforço computacional que o método GNCS, podemos verificar isso em [7].

Nesse trabalho aperfeiçoamos os métodos de pontos interiores já existentes. Modificamos a forma de calcular o ponto inicial, os resíduos e as direções dos métodos. Cada modificação teve sua contribuição para aperfeiçoar os métodos.

Com as modificações no ponto inicial conseguimos reduzir o número de iterações e o valor da função objetivo. O fato de conseguirmos diminuir o valor da função objetivo na solução ótima, mostra que os métodos antigos estavam parando antes. Com a modificação no ponto inicial, nos aproximamos mais da solução ótima global. Assim temos a possibilidade de escolher bons pontos de partida.

Com as modificações no cálculo dos resíduos, o tempo por iteração aumenta porque

a nova forma de calculá-los é mais trabalhosa que a forma antiga, mas a diferença de tempo é quase insignificante. Entretanto conseguimos diminuir o número de iterações nos métodos barreira logarítmica preditor-corretor e primal-dual barreira logarítmica preditor-corretor.

Com as modificações na forma de calcular as direções não conseguimos visualizar melhora apenas nos testes computacionais apresentados no Capítulo 6, entretanto verificamos que essas modificações realmente reduzem o número de operações necessárias em cada iteração através do número de flops obtidos.

A nova formulação obtida através das modificações para os métodos de pontos interiores possibilita melhoria tanto nos testes computacionais em MATLAB quanto na implementação eficiente.

Na implementação eficiente em C aproveitamos a estrutura matricial dos métodos para economizar memória. Isso é possível porque muitos valores deixam de ser armazenados. Com isso a complexidade computacional é reduzida e podemos resolver problemas de maior porte de forma eficiente. Em particular, não é necessário armazenar nenhuma matriz durante a solução de um problema.

Os métodos de pontos interiores apresentam uma estrutura matricial que faz com que a implementação seja menos trabalhosa comparada à implementação do método GNCS, por exemplo, visto que o método necessita de um procedimento de busca linear que é computacionalmente muito caro.

Experimentos numéricos mostram que de todos os métodos apresentados, os métodos

de pontos interiores especializados para o problema de regressão pela norma L_p apresentam melhor desempenho. Dentre os métodos de pontos interiores, o que apresenta melhor desempenho em problemas de grande porte é o primal-dual barreira logarítmica preditor-corretor que, de fato, é o mais robusto entre aqueles discutidos neste trabalho.

Referências Bibliográficas

- [1] N. N. ABDELMALEK, *l_1 solution of overdetermined systems of linear equations*, ACM Trans. on Mathematical Software, 6 (1980), pp. 220–227.
- [2] R. D. ARMSTRONG, E. L. FROME, AND D. S. KUNG, *A revised simplex algorithm for the absolute deviation curve fitting problem*, Commun. Statist., B8 (1979), pp. 175–190.
- [3] T. S. ARTHANARI AND Y. DODGE, *Mathematical Programming in Statistics*, John Wiley, New York, 1981.
- [4] I. BARRODALE AND F. D. K. ROBERTS, *An improved algorithm for discrete l_1 linear approximation*, SIAM J. Numer. Anal., 10 (1973), pp. 839–848.
- [5] M. BAZARAA AND C. M. SHETTY, *Nonlinear Programming-Theory and Algorithms*, John Wiley, 1993.
- [6] D. P. BERTSEKAS, *Nonlinear Programming*, Athena Scientific, 1999.
- [7] D. R. CANTANE, *Métodos de pontos interiores aplicados ao problema de regressão pela norma L_p* , tech. report, Dissertação de Mestrado, ICMC – USP, São Carlos SP, Março, 2004. <http://www.teses.usp.br/teses/disponiveis/55/55134/tde-18112004-135942>.
- [8] T. F. COLEMAN AND Y. LI, *A global and quadratically convergent method for linear l_∞ problems*, SIAM J. Numer. Anal., 29 (1992), pp. 1166–1186.

- [9] T. F. COLEMAN AND Y. LI, *A globally and quadratically convergent affine scaling method for linear l_1 problems*, Mathematical Programming, 56 (1992), pp. 189–222.
- [10] G. DANTZIG, *Linear Programming and Extensions*, Princeton, 1963.
- [11] A. DAX AND B. BERKOWITZ, *Column relaxation methods for least norm problems*, SIAM J. Sci. Stat. Comput., 11 (1990), pp. 975–989.
- [12] A. R. L. O. E DANIELA RENATA CANTANE, *Métodos de pontos interiores aplicados ao problema de regressão pela norma L_p* , Tendências em Matemática Aplicada e Computacional, 5 (2004), pp. 281–291.
- [13] A. S. EL-BAKRY, R. A. TAPIA, T. TSUCHIYA, AND Y. ZHANG, *On the formulation and the theory of the Newton interior-point method for nonlinear programming*, Journal of Optimization Theory and Applications, 89 (1996), pp. 507–541.
- [14] G. H. GOLUB AND C. F. VAN LOAN, *Matrix Computations 2nd Edition*, The Johns Hopkins University Press, Baltimore, Maryland, 1989.
- [15] G. HEINIG AND P. JANKOWSKI, *Parallel and superfast algorithms for Hankel systems*, Numerische Mathematik, 58 (1990), pp. 109–127.
- [16] Y. LI, *A globally convergent method for l_p problems*, SIAM J. Optimization, 3 (1993), pp. 609–629.
- [17] D. G. LUENBERGER AND Y. YE, *Linear and Nonlinear Programming*, Springer, New York, 2008.
- [18] S. MEHROTRA, *On the implementation of a primal-dual interior point method*, SIAM Journal on Optimization, 2 (1992), pp. 575–601.
- [19] G. MERLE AND H. SPÄTH, *Computational experience with discrete l_p approximation*, Computing, 12 (1974), pp. 315–321.

- [20] A. R. L. OLIVEIRA AND C. LYRA, *Interior point methods for the polynomial l_∞ fitting problem*, International Transactions in Operational Research, 11 (2004), pp. 309–322.
- [21] A. R. L. OLIVEIRA, M. A. NASCIMENTO, AND C. LYRA, *Efficient implementation and benchmark of interior point methods for the polynomial L_1 fitting problem*, Computational Statistics & Data Analysis., 35 (2000), pp. 119–135.
- [22] M. R. OSBORNE, *Finite Algorithms in Optimization and Data Analysis*, John Wiley, New York, 1985.
- [23] M. SAKAROVITCH, *Linear Programming*, Springer Verlag, 1983.
- [24] T. TERLAKI, *Interior Point Methods of Mathematical Programming*, Kluwer Academic, 1996.
- [25] W. F. TRENCH, *An algorithm for the inversion of finite Hankel matrices*, J. Soc. Indust. Appl. Math., 13 (1965), pp. 1102–1107.
- [26] R. J. VANDERBEI, *Linear Programming – Foundations and Extensions*, Kluwer Academic Publishers, Boston, USA, 1996.
- [27] W. H. WAGNER, *Linear programming techniques for regression analysis*, J. Amer. Statist. Assoc., 54 (1959), pp. 206–212.
- [28] S. J. WRIGHT, *Primal–Dual Interior–Point Methods*, SIAM Publications, SIAM, Philadelphia, PA, USA, 1996.