



LINO MARCOS DA SILVA

**MODIFICAÇÕES NA FATORAÇÃO CONTROLADA DE CHOLSKY
PARA ACELERAR O PRECONDICIONAMENTO DE SISTEMAS
LINEARES NO CONTEXTO DE PONTOS INTERIORES**

**CAMPINAS
2014**



UNIVERSIDADE ESTADUAL DE CAMPINAS

Instituto de Matemática, Estatística
e Computação Científica

LINO MARCOS DA SILVA

**MODIFICAÇÕES NA FATORAÇÃO CONTROLADA DE CHOLESKY
PARA ACELERAR O PRECONDICIONAMENTO DE SISTEMAS
LINEARES NO CONTEXTO DE PONTOS INTERIORES**

Tese apresentada ao Instituto de Matemática, Estatística e Computação Científica da Universidade Estadual de Campinas como parte dos requisitos exigidos para a obtenção do título de Doutor em matemática aplicada.

Orientador: Aurelio Ribeiro Leite de Oliveira

ESTE EXEMPLAR CORRESPONDE À VERSÃO FINAL DA
TESE DEFENDIDA PELO ALUNO LINO MARCOS DA SILVA,
E ORIENTADA PELO PROF. DR. AURELIO RIBEIRO
LEITE DE OLIVEIRA.

Assinatura do Orientador

Aurelio R L de Oliveira

CAMPINAS
2014

Ficha catalográfica
Universidade Estadual de Campinas
Biblioteca do Instituto de Matemática, Estatística e Computação Científica
Maria Fabiana Bezerra Muller - CRB 8/6162

Si38m Silva, Lino Marcos da, 1978-
Modificações na fatoração controlada de Cholesky para acelerar o
precondicionamento de sistemas lineares no contexto de pontos interiores / Lino
Marcos da Silva. – Campinas, SP : [s.n.], 2014.

Orientador: Aurelio Ribeiro Leite de Oliveira.
Tese (doutorado) – Universidade Estadual de Campinas, Instituto de
Matemática, Estatística e Computação Científica.

1. Precondicionadores. 2. Fatoração de Cholesky incompleta. 3. Método de
pontos interiores. I. Oliveira, Aurelio Ribeiro Leite de, 1962-. II. Universidade
Estadual de Campinas. Instituto de Matemática, Estatística e Computação
Científica. III. Título.

Informações para Biblioteca Digital

Título em outro idioma: Modifications on controlled Cholesky factorization to improve the
preconditioning in interior point method

Palavras-chave em inglês:

Preconditioners

Incomplete Cholesky factorization

Interior point method

Área de concentração: Matemática Aplicada

Titulação: Doutor em Matemática Aplicada

Banca examinadora:

Aurelio Ribeiro Leite de Oliveira [Orientador]

Márcia Aparecida Gomes Ruggiero

Carla Tavianes Lucke da Silva Ghidini

Frederico Ferreira Campos Filho

Roberto Quirino do Nascimento

Data de defesa: 02-09-2014

Programa de Pós-Graduação: Matemática Aplicada

Tese de Doutorado defendida em 02 de setembro de 2014 e aprovada

Pela Banca Examinadora composta pelos Profs. Drs.

Aurelio N Leite de Oliveira

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

Frederico Ferreira Campos Filho

Prof(a). Dr(a). FREDERICO FERREIRA CAMPOS FILHO

Roberto Quirino do Nascimento

Prof(a). Dr(a). ROBERTO QUIRINO DO NASCIMENTO

Carla Tavianes Lucke da Silva Ghidini

Prof(a). Dr(a). CARLA TAVIANES LUCKE DA SILVA GHIDINI

Márcia Aparecida Gomes Ruggiero

Prof(a). Dr(a). MÁRCIA APARECIDA GOMES RUGGIERO

Abstract

The interior point method solves large linear programming problems in few iterations. However, each iteration requires computing the solution of one or more linear systems. This constitutes the most expensive step of the method by greatly increasing the processing time and the need for data storage. According to it, reducing the time to solve the linear system is a way of improving the method performance. In general, large linear programming problems have sparse matrices. Since the linear systems to be solved are symmetric positive definite, iterative methods such as the preconditioned conjugate gradient method can be used to solve them. Furthermore, incomplete Cholesky factor can be used as a preconditioner to the problem. On the other hand, breakdown may occur during incomplete factorizations. When such failure occur, a correction is made by adding a positive number to diagonal elements of the linear system matrix and the factorization of the new matrix is restarted, thus increasing the time of preconditioning, either due to computing the preconditioner, or due to loss of its quality. The controlled Cholesky factorization preconditioner performs well in early iterations of interior point methods and has been important on implementations of hybrid preconditioning approaches. However, being an incomplete factorization, it is not free from faulty pivots. In this study we propose two modifications to the controlled Cholesky factorization in order to avoid or decrease the refactorizing diagonally modified matrices number. Computational results show that the proposed techniques can significantly reduces the time for solving linear programming problems by interior point method.

Keywords: Interior point method, Preconditioners, Incomplete Cholesky Factorization.

Resumo

O método de pontos interiores para programação linear resolve em poucas iterações problemas de grande porte. No entanto, requer a cada iteração a resolução de dois sistemas lineares, os quais possuem a mesma matriz de coeficientes. Essa etapa se constitui no passo mais caro do método por aumentar consideravelmente o tempo de processamento e a necessidade de armazenamento de dados. Reduzir o tempo de solução dos sistemas lineares é, portanto, uma forma de melhorar o desempenho do método. De um modo geral, problemas de programação linear de grande porte possuem matrizes esparsas. Uma vez que os sistemas lineares a serem resolvidos são simétricos

positivos definidos, métodos iterativos como o método dos gradientes conjugados preconditionado podem ser utilizados na resolução dos mesmos. Além disso, fatores de Cholesky incompletos podem ser utilizados como preconditionadores para o problema. Por outro lado, fatorações incompletas podem sofrer falhas na diagonal durante o processo de fatoração, e quando tais falhas ocorrem uma correção é efetuada somando-se um valor positivo aos elementos da diagonal da matriz do sistema linear e a fatoração da nova matriz é reiniciada, aumentando dessa forma o tempo de preconditionamento, quer seja devido a reconstrução do preconditionador, quer seja devido a perda de qualidade do mesmo. O preconditionador fatoração controlada de Cholesky tem um bom desempenho nas iterações iniciais do método de pontos interiores e tem sido importante nas implementações de abordagens de preconditionamento híbrido. No entanto, sendo uma fatoração incompleta, o mesmo não está livre da ocorrência de falhas no cálculo do pivô. Neste estudo propomos duas modificações à fatoração controlada de Cholesky a fim de evitar ou diminuir o número de reinícios da fatoração das matrizes diagonalmente modificadas. Resultados computacionais mostram que a técnica pode reduzir significativamente o tempo de resolução de certas classes de problemas de programação linear via método de pontos interiores

Palavras-chave: Método de pontos interiores, Preconditionadores, Fatoração de Cholesky incompleta.

Sumário

Dedicatória	xi
Agradecimentos	xiii
1 Introdução	1
1.1 Notação	2
2 Programação Linear	3
2.1 Introdução	3
2.2 Problema de Programação Linear	4
2.3 Método de Pontos Interiores	6
2.3.1 Abordagens para resolver o Sistema Linear	10
3 Técnicas de Precondicionamento de Sistemas Lineares	17
3.1 Introdução	17
3.2 Precondicionamento para $\mathcal{A}x = b$	18
3.2.1 Fatorações Incompletas	20
3.3 Precondicionamento em Métodos de Pontos Interiores	23
3.3.1 Precondicionadores para as Equações Normais	24
3.3.2 Precondicionadores para o Sistema Aumentado	27
3.3.3 Precondicionadores Híbridos	30
4 Fatoração Controlada de Cholesky Modificada	33
4.1 Introdução	33
4.2 Correção na diagonal	34
4.2.1 Incremento linear	35
4.2.2 Incremento exponencial	35
4.2.3 Correção na diagonal na fatoração controlada de Cholesky	36
4.3 Uma nova abordagem para o cálculo de α na fatoração controlada de Cholesky	37
4.3.1 Estratégia para o cálculo de α	38
4.3.2 Fatoração Controlada de Cholesky Modificada Alfa	38
4.4 Precondicionadores para Sistemas Lineares diagonalmente modificados	40
4.4.1 Uma classe de precondicionadores atualizados para $\{\mathcal{A} + \Delta_k\}$	40
4.5 Uma proposta de Atualização para a Fatoração Controlada de Cholesky	44

4.5.1	Características do Precondicionador Atualizado	48
4.5.2	Fatoração Controlada de Cholesky Modificada Beta	50
5	Resultados Numéricos	53
5.1	Introdução	53
5.1.1	PCx	54
5.1.2	Perfil de desempenho	54
5.2	Problemas Testes	55
5.3	Resultados Computacionais	55
5.3.1	FCC modificada α	58
5.3.2	FCC Modificada sem o reinício de fatorações	63
5.4	Algoritmo de Ajustamento Ótimo para p Coordenadas	70
5.5	Potencialidades da FCC β	71
5.6	FCC modificada $\times$ FCC original	74
6	Conclusões e Trabalhos Futuros	75
6.1	Conclusões	75
6.2	Trabalhos Futuros	76
	Referências	77

Aos meus pais Dulce Gonçalves e Manoel Lino.

Agradecimentos

A Deus pela vida e por ter me abençoado com uma família adorável, amigos maravilhosos e o dom da perseverança.

Ao meu pai e à minha mãe pelas orações de todos os dias e por terem me ensinado os mais nobres valores da vida.

Aos meus irmãos e as minhas irmãs por compartilharem comigo dificuldades e sonhos e, por me apoiarem em todas as decisões que me trouxeram até aqui.

Ao prof. Dr. Aurelio Ribeiro Leite de Oliveira, pela confiança, pelo projeto, pelo apoio e colaboração sem os quais não seriam possíveis a realização deste trabalho.

Aos amigos Adson Mota, Valter Soares e Porfirio Suñagua, pelas experiências pessoais e profissionais compartilhadas, pelo carinho, pelos incentivos, auxílios e conselhos mútuos.

A todos os amigos que conheci no IMECC e com os quais aprendi muito sobre matemática e sobre a vida. Em especial, aos paulistanos Felipe Bacani, Marcus Vinicius Nascimento e Roberta Delboni; aos colegas nordestinos Rinaldo Vieira, Cícero Alfredo, Ingrid Sampaio e Tatiana Rocha; e aos “gringos” Brian Calander e Rollo Jenkins.

À amiga profa. Dra. Elizabeth Gasparim por ter possibilitado-me um convívio mais amplo com o universo da matemática e, também, pelos importantes conselhos pessoais e profissionais.

Ao amigo de fé e irmão camarada Prof. Dantas, primeiro por ter acreditado em mim como professor de matemática e me incentivado a continuar nessa área, e depois por todo o apoio e carinho despendido durante os estudos do doutorado.

A UNIVASF, em particular ao Colegiado de Engenharia Elétrica - CENEL.

Aos colegas professores de Matemática da UNIVASF, em especial aos amigos Damião Silva e Edson Leite.

Às Profas. Dras. Carla Ghidini e Silvana Bocanegra pelo precioso auxílio com o código do PCx.

Aos colegas do IMECC Raniere Silva, Tiago Macedo e Júnior Soares (e demais colaboradores) pelo excelente trabalho na elaboração deste modelo de tese e, também, pelo espírito de solidariedade por compartilhá-lo com toda a comunidade do IMECC, facilitando imensamente o laborioso trabalho de redação de dissertações e teses.

Ao IMECC por ter uma excelente estrutura de trabalho e por ser um importante centro de excelência para o Brasil, em especial aos seus funcionários pela presteza no atendimento.

A CAPES, pelo apoio financeiro.

A Banca Examinadora pelas sugestões, críticas, pela paciência e pela compreensão em avaliar o nosso trabalho.

Aos professores Adolfo M. Junior, Aurelio R.L de OLiveira, Cid C. de Souza (Instituto de Computação), Maria A. D. Ehrhardt (Cheti), Francisco A. M. Gomes Neto (Chico), Márcia A.G. Rugiero e Samuel R. de Oliveira que muito contribuíram para minha formação profissional e científica com seus ensinamentos acadêmicos e postura profissional.

Nobody said it was easy
No one ever said it would be this hard.
The Scientist by Coldplay

Capítulo 1

Introdução

Métodos de pontos interiores têm sido utilizados com sucesso na resolução de problemas lineares de grande porte há várias décadas. Uma importante característica desses métodos é o fato dos mesmos encontrarem uma solução para o problema de programação linear em poucas iterações. No entanto, cada iteração do método demanda um alto custo computacional, principalmente, pela necessidade da resolução de dois sistemas lineares por iteração [24].

Uma maneira de resolver os sistemas lineares em métodos de pontos interiores, como na variante preditor-corretor [32], por exemplo, é obter o sistema de equações normais, cuja matriz é simétrica definida positiva e utilizar o método dos gradientes conjugados preconditionado. Por outro lado, as matrizes dos coeficientes dos sistemas lineares em métodos de pontos interiores tendem a ficar mal condicionadas ao longo das iterações do método. Devido a isso, técnicas de preconditionamento específicas para esses problemas têm sido amplamente estudadas [10, 12, 21, 24, 38, 43].

Fatorações incompletas, como a fatoração controlada de Cholesky [15, 16], são preconditionadores que têm um bom desempenho nas iterações iniciais do método de pontos interiores, mas o mesmo não acontece nas iterações próximas à convergência, quando a matriz se torna muito mal condicionada [12, 38]. Contudo esse preconditionador tem um importante papel em técnicas de preconditionamento híbrido. Em particular, abordagens híbridas que usam a fatoração controlada de Cholesky nas iterações iniciais e o preconditionador separador nas iterações finais do método de pontos interiores têm reportado bons resultados [12, 21].

Uma desvantagem da fatoração controlada de Cholesky é que a mesma é suscetível a falhas na diagonal durante o processo de fatoração. A técnica utilizada para corrigir essas falhas, em geral, aumenta consideravelmente o tempo de preconditionamento do problema e, conseqüentemente, do método de pontos interiores [12].

Neste trabalho, duas propostas de modificação na fatoração controlada de Cholesky são apresentadas. De um modo geral, o objetivo de ambas é o mesmo: reduzir o tempo computacional gasto na resolução de problemas de programação linear via método de pontos interiores. Mas, de um modo específico, a filosofia dessas modificações são distintas. A primeira proposta modifica a maneira de calcular o parâmetro de correção das falhas que ocorrem na diagonal. Permite efetuar correções começando com parâmetros de valores maiores do que o de costume e também o reinício da fatoração para construir o preconditionador. A segunda, apesar de usar o mesmo modelo para o cálculo do parâmetro de correção na diagonal, não permite o reinício da fatoração durante a

construção do preconditionador. Em vez disso efetua uma atualização nos elementos calculados antes da ocorrência da falha na diagonal.

Esta tese está organizada do seguinte modo: no Capítulo 2, é apresentada a teoria básica de programação linear, de métodos de pontos interiores e a álgebra linear dos métodos de pontos interiores. As técnicas de preconditionamento utilizadas para acelerar a convergência do método dos gradientes conjugados são discutidas no Capítulo 3. O Capítulo 4 apresenta as propostas de modificação na fatoração controlada de Cholesky, que é o tema central deste trabalho. No Capítulo 5, são apresentados os resultados computacionais, e por fim, no Capítulo 6, são apresentados as conclusões e as propostas de trabalhos futuros.

1.1 Notação

Dada uma matriz A , $m \times n$, o escalar a_{ij} representa o elemento de A que está na i -ésima linha e na j -ésima coluna. O símbolo 0 pode denotar tanto o número real zero, como o vetor coluna com entradas todas nulas, ou ainda a matriz nula, cuja dimensão ficará clara de acordo com o contexto. A matriz identidade de ordem n será denotada por I_n , porém quando não houver possibilidade de confusão quanto às dimensões das matrizes envolvidas, a matriz identidade será denotada simplesmente por I . O símbolo e_j indicará o vetor coluna cuja j -ésima componente é o número um e as demais componentes são zeros. A letra e denotará o vetor coluna cujas componentes são todas iguais ao número um. As dimensões desses vetores estarão de acordo com o contexto.

A norma Euclidiana é representada por $||\cdot||$. Este símbolo também denotará a norma-2 para matrizes. Já o símbolo $||\cdot||_F$ indicará a norma de Frobenius da respectiva matriz. A relação $X = \text{diag}(x)$ significa que X é a matriz diagonal cujos elementos da diagonal são aqueles da n -upla $(x_1, x_2, \dots, x_n)$. Por outro lado, $\text{diag}(A)$ poderá representar o vetor coluna formado pelos elementos da diagonal de A ou a matriz diagonal cujos elementos da diagonal coincidem com os elementos de A . O símbolo $\text{off}(A)$ indicará a matriz formada pelos elementos da matriz A que não estão na diagonal, isto é,

$$\text{off}(A) = A - \text{diag}(A).$$

Os símbolos X^k , x^k e x_k indicarão, respectivamente, o valor da matriz X , do vetor x e do escalar x no k -ésimo passo de um procedimento iterativo. Embora x_k possa denotar também a k -ésima coordenada do vetor x , espera-se não haver confusão a respeito do significado do símbolo devido ao contexto em que os mesmos estarão sendo utilizados.

Capítulo 2

Programação Linear

2.1 Introdução

A programação linear trata do problema de minimizar ou maximizar uma função linear sujeita a restrições impostas por equações e/ou inequações lineares. Desde o seu desenvolvimento por George B. Dantzig na década de 1940, tem se tornado o centro das atenções da área de otimização [24], sendo apontada como um dos grandes sucessos desta área [44] e também como o mais natural mecanismo para formulação, a um baixo custo, de uma ampla classe de problemas [30]. Exaustivamente utilizada nos campos militar, industrial, governamental, planejamento urbano, dentre outros, a programação linear tem, entre as razões que justificam o seu sucesso, a habilidade em modelar problemas grandes e complexos e de possibilitar a seus usuários resolver tais problemas por meio de algoritmos eficientes e a um razoável custo computacional [4].

O reconhecido sucesso da área de programação linear, também está relacionado à existência de um poderoso, e não menos famoso, método para resolver problemas típicos da área: o método simplex. O seu desenvolvimento por Dantzig em 1949 é apontado como o evento mais significativo da história da programação linear e, desde então, este foi o único método disponível para resolver problemas de programação linear por muitos anos. Porém, apesar de ser um método eficiente e confiável na prática, teoricamente o método possui complexidade não polinomial, isto é, no pior caso, o simplex pode realizar um número de passos que depende exponencialmente do tamanho do problema [24]. Este fato incomodava alguns pesquisadores que acreditavam ser possível o desenvolvimento de um bom método que tivesse um número de passos polinomial em vez de exponencial como o simplex [30].

Em 1984 outro evento voltaria a chamar a atenção da comunidade de programação linear e se tornaria o evento mais importante da área desde o desenvolvimento do método simplex: a publicação de um novo método para resolver o problema de programação linear por N. K. Karmarkar [26]. A razão para isso se deve, em parte, ao fato do método recém-proposto possuir complexidade polinomial, uma desejada propriedade teórica. Apesar do fato de uma única iteração do método de Karmarkar ser muito cara computacionalmente, o mesmo se tornou muito atrativo, devido a possibilidade de encontrar uma solução ótima do problema em número reduzido de iterações.

O trabalho de Karmarkar, juntamente com outros trabalhos realizados anteriormente como,

por exemplo, o método elipsoidal de Khachiyan [28], impulsionaram o desenvolvimento de métodos para a resolução do problema de programação linear em tempo polinomial, o que culminou com o desenvolvimento de uma família de métodos conhecida atualmente por métodos de pontos interiores.

Na década de 1990, já havia evidências suficientes da eficiência dos métodos de pontos interiores na resolução de problemas de programação linear de grande porte [24]. Até chegar a uma solução ótima, métodos de pontos interiores realizam um número de iterações bem inferior às realizadas pelo método simplex. Porém, o custo computacional de cada iteração é muito superior a uma iteração realizada pelo método simplex. Atualmente, os dois métodos estão bem estabelecidos como métodos eficientes e confiáveis para a resolução do problema de programação linear. No entanto, há uma preferência pelo método simplex quando se trata de resolver problemas menores, enquanto os de maior porte podem ser resolvidos mais eficientemente por um método de pontos interiores.

Neste capítulo, será apresentada a teoria básica de programação linear, métodos de pontos interiores, bem como a álgebra linear envolvida na resolução de problemas de programação linear via métodos de pontos interiores.

2.2 Problema de Programação Linear

Considere o problema de programação linear na forma padrão, chamado de *problema primal*, o qual é dado por

$$\begin{array}{ll} \min & c^T x \\ \text{s.a} & Ax = b, \\ & x \geq 0, \end{array} \quad (2.2.1)$$

sendo $A \in \mathbb{R}^{m \times n}$ a matriz das restrições do problema, $c \in \mathbb{R}^n$ o vetor dos coeficientes da função objetivo, $b \in \mathbb{R}^m$ o vetor dos termos independentes e $x \in \mathbb{R}^n$ o vetor das variáveis a serem determinadas, também chamadas de variáveis de decisão. Considera-se que a matriz A tem posto completo m . A desigualdade $x \geq 0$ indica que cada componente x_i , $i = 1, \dots, n$, do vetor x é não negativa.

Associado ao problema primal (2.2.1) existe o *problema dual* que é dado por

$$\begin{array}{ll} \max & b^T y \\ \text{s.a} & A^T y + z = c, \\ & z \geq 0, \end{array} \quad (2.2.2)$$

onde $y \in \mathbb{R}^m$ é o vetor de variáveis duais livres e $z \in \mathbb{R}^n$ representa as variáveis de folga duais.

Chama-se de *solução factível* dos problemas (2.2.1) e (2.2.2), respectivamente, aos vetores x e (y, z) que satisfazem $Ax = b$, $x \geq 0$, $A^T y + z = c$ e $z \geq 0$. Dada uma solução factível x do problema primal e uma solução factível (y, z) do problema dual, o *gap de dualidade* γ é definido como sendo

$$\gamma = c^T x - b^T y = x^T z \geq 0. \quad (2.2.3)$$

Métodos de pontos interiores exploram um importante conceito em programação linear que é o das *condições de otimalidade*. Estas são condições algébricas que devem ser satisfeitas pelas

soluções do problema de programação linear e são derivadas do primeiro princípio de dualidade e da teoria relacionada com o mesmo. Esta teoria é apresentada nos mais diversos textos de programação linear, como por exemplo [42, 30, 36]. De outro modo, tais condições também podem ser descritas como um caso particular das condições de otimalidade para os problemas gerais de otimização restrita, conhecidas como as condições de Karush-Kuhn-Tucker, ou abreviadamente, *condições KKT* [5]. Para os problemas (2.2.1) e (2.2.2) as condições KKT são dadas por

$$Ax = b, \quad (2.2.4a)$$

$$A^T y + z = c, \quad (2.2.4b)$$

$$x_i z_i = 0 \quad i = 1, \dots, n, \quad (2.2.4c)$$

$$x, z \geq 0. \quad (2.2.4d)$$

Qualquer ponto (x, y, z) que satisfaz as condições (2.2.4a), (2.2.4b) e (2.2.4d) é chamado de *ponto factível*, enquanto aqueles que satisfazem estritamente a última condição, isto é, $(x, z) > 0$, além das duas primeiras é dito ser *estritamente factível*. O conjunto de pontos que satisfazem as condições dos problemas primal e dual definem um polígono conhecido como a região factível do problema de programação linear.

As condições em (2.2.4c) implicam que para cada índice $i = 1, 2, \dots, n$, uma das componentes x_i ou z_i deve ser igual a zero. Estas condições são chamadas *condições de complementaridade* por implicarem que as componentes não nulas de x e de z aparecem em posições complementares.

Introduzindo as notações

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \text{ e } X = \text{diag}(x) = \begin{bmatrix} x_1 & & & \\ & x_2 & & \\ & & \ddots & \\ & & & x_n \end{bmatrix},$$

as condições de complementaridade (2.2.4c) podem ser reescritas como $XZe = 0$, onde $Z = \text{diag}(z)$ e $e = (1 \dots 1)^T$. Dessa forma, obtém-se uma forma mais compacta de escrever as condições KKT:

$$\begin{aligned} Ax &= b, \\ A^T y + z &= c, \\ XZe &= 0, \\ x, z &\geq 0. \end{aligned} \quad (2.2.5)$$

Observa-se que no sistema (2.2.5) os dois primeiros conjuntos de equações são lineares enquanto aquele que representa as condições de complementaridade é não linear. Desse modo o sistema KKT envolvendo apenas essas equações não oferecem grandes dificuldades de solução. Contudo o problema se torna mais difícil de resolver quando acrescentamos as restrições de não negatividade $(x, z) \geq 0$.

Entre os métodos conhecidos para resolver o problema de programação linear, dois deles se destacam por estarem bem estabelecidos como métodos eficientes para resolver o problema (2.2.1),

são eles: o *Método Simplex* e o *Método de Pontos Interiores*. O modo como a condição de complementaridade é tratada durante a resolução do problema de programação linear, é a origem das principais diferenças entre esses dois métodos de programação. O método simplex, por exemplo, escolhe um subconjunto de índices $j \in \mathcal{B} \subset \{1, 2, \dots, n\}$ tal que x_j pode ser não nulo e força o correspondente $z_j = 0$, enquanto os índices restantes $j \in \mathcal{N} \subset \{1, 2, \dots, n\} \setminus \mathcal{B}$ força $x_j = 0$ e permite z_j assumir valores diferentes de zero. Já os métodos de pontos interiores perturbam a condição de complementaridade, substituindo a condição $x_j z_j = 0$ por $x_j z_j = \mu$, e forçam a redução gradativa do parâmetro μ até uma vizinhança de zero, a cada iteração do método. A maneira como as condições de complementaridade são tratadas impacta diretamente na implementação de cada método, na sua complexidade teórica e na sua eficiência prática [24, 44]. Isso é responsável, por exemplo, pelo tipo de técnica de álgebra linear a ser utilizada e, consequentemente, pelo custo computacional do mesmo. De um modo geral, o método simplex realiza muito mais iterações que os métodos de pontos interiores, porém estas são muito mais baratas. Na prática, o número de iterações do método de pontos interiores é pequeno, independentemente da dimensão do problema. Além disso, os métodos de pontos interiores têm se mostrado uma excelente alternativa ao método Simplex na resolução de problemas lineares de grande porte [24] e serão apresentados em mais detalhes na próxima seção.

2.3 Método de Pontos Interiores

Os métodos de pontos interiores encontram uma solução ótima do problema de programação linear seguindo uma trajetória no interior do ortante positivo, chamada de *caminho central*, e aproximam-se da fronteira desta região apenas no limite. Algumas informações gerais sobre esses métodos serão apresentadas no próximo parágrafo, o qual propõe uma breve incursão ao cerne dos métodos de pontos interiores, notadamente, aqueles do tipo primal-dual. Apesar da exposição ser bem sucinta espera-se que a mesma possa clarear pontos-chave que ajudam a entender melhor aspectos teóricos e computacionais de tais métodos.

O *método de pontos interiores do tipo primal-dual* encontra soluções primais-duais (x^*, y^*, z^*) aplicando o método de Newton às três primeiras condições de otimalidade em (2.2.5) para encontrar as direções de busca. Em seguida, modifica essas direções e os comprimentos dos passos de modo que as desigualdades $x \geq 0$ e $z \geq 0$ sejam satisfeitas estritamente em cada iteração. Segundo Wright [44] essa exigência simples, e aparentemente inócua, sobre os limitantes das variáveis x e y , é a razão de todas as complicações na construção, análise e implementação dos métodos de pontos interiores.

Considere a função $F : \mathbb{R}^{2n+m} \rightarrow \mathbb{R}^{2n+m}$ dada por

$$F(x, y, z) = \begin{bmatrix} A^T y + z - c \\ Ax - b \\ XZe \end{bmatrix} = 0,$$

para $(x, z) \geq 0$. É importante observar que a função F é na verdade outra forma de escrever as condições (2.2.5). Observa-se que a mesma é linear nos argumentos $A^T y + z - c$ e $Ax - b$, e não linear no argumento XZe .

Todos os métodos de pontos interiores do tipo primal-dual geram iterandos (x^k, y^k, z^k) que satisfazem os limitantes $(x, y) > 0$. Isto é, $x^k, y^k > 0$. Esta propriedade é responsável pela origem do termo *pontos interiores*. Além disso, abordagens desse tipo evitam as chamadas soluções espúrias, que são aquelas que satisfazem $F(x, y, z) = 0$, mas não $(x, z) \geq 0$. Soluções espúrias são abundantes, mas nenhuma delas dá qualquer informação útil sobre as soluções de (2.2.1) e (2.2.2). Logo, é uma boa prática excluí-las da região de busca.

Assim como a maioria dos métodos iterativos em otimização, os métodos de pontos interiores primal-dual também realizam, pelo menos, dois passos básicos: um procedimento para determinar uma direção de busca e uma medida de quanto se deseja andar nessa direção, ou seja, o tamanho do passo a ser considerado.

O procedimento adotado por métodos de pontos interiores do tipo primal-dual para encontrar a direção de busca tem origens no método de Newton para equações não lineares. O método de Newton forma um modelo linear para a função F na vizinhança do ponto corrente e obtém a direção de busca $(\Delta x, \Delta y, \Delta z)$ através da resolução do seguinte sistema de equações lineares:

$$J(x, y, z) \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix} = -F(x, y, z),$$

sendo J o Jacobiano da função F . Se o ponto corrente é estritamente factível, tem-se $A^T y + z - c = 0$ e $Ax - b = 0$. Logo, a equação do passo de Newton pode ser reescrita como:

$$\begin{bmatrix} 0 & A^T & I \\ A & 0 & 0 \\ Z & 0 & X \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -XZe \end{bmatrix}. \quad (2.3.1)$$

Um passo completo nessa direção, em geral, não é permitido, pois isso poderia violar as condições $(x, y) \geq 0$. Para evitar esta dificuldade, o método realiza uma busca linear ao longo da direção de Newton, de modo que a nova iterada é determinada por

$$(x, y, z) + \alpha(\Delta x, \Delta y, \Delta z)$$

para algum parâmetro $\alpha \in (0, 1]$. Usualmente, apenas um pequeno passo ao longo da direção de Newton é permitido, pois do contrário a condição $(x, z) > 0$ pode ser violada. Deste modo, a direção de Newton pura, também conhecida como *direção afim escala*, frequentemente não permite um bom desempenho na busca da solução do sistema em questão.

Muitos dos métodos primais-duais usam um método de Newton menos agressivo, que não visa encontrar diretamente uma solução do sistema formado pelas equações (2.2.4a), (2.2.4b) e (2.2.4c) mas, em vez disso, encontrar um ponto cujos produtos $x_i z_i$ sejam reduzidos, na mesma razão, ao longo das iterações do método. Na prática, toma-se um passo na direção de Newton para o qual $x_i z_i = \sigma \mu$, onde μ é a medida do *gap de dualidade* corrente definida como

$$\mu = \frac{1}{n} \sum_{i=1}^n x_i z_i = \frac{x^T z}{n},$$

e $\sigma \in [0, 1]$ é o fator de redução que deseja-se encontrar na medida do *gap* de dualidade no passo em questão. O passo modificado fica

$$\begin{bmatrix} 0 & A^T & I \\ A & 0 & 0 \\ Z & 0 & X \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix} = \begin{bmatrix} -r_d \\ -r_p \\ -XZe + \sigma \mu e \end{bmatrix}, \quad (2.3.2)$$

onde $r_d = c - A^T y - z$ é chamado de *resíduo dual* e $r_p = b - Ax$ é chamado de *resíduo primal*. Devido à razões geométricas, σ é chamado de *parâmetro de centragem*. Quando $\sigma > 0$, usualmente é possível tomar um passo α maior na direção de Newton.

Neste trabalho, será utilizada uma variante do método de pontos interiores do tipo primal-dual chamada de *preditor-corretor*. Esta escolha se baseia no fato de que esta variante possui melhores propriedades teóricas e computacionais que os demais [44]. Suas principais propriedades serão apresentadas na próxima seção.

2.3.0.1 Método Preditor-Corretor de Mehrotra

O método Preditor-Corretor de Mehrotra [32] difere do método primal-dual padrão na escolha das direções de busca. As condições KKT são modificadas de modo a manter a não negatividade das componentes (x, z) e a incorporar o parâmetro de centragem. Em seguida, o método de Newton é aplicado. A direção de busca, em cada iteração k do método, é obtida pela resolução de dois sistemas lineares distintos, porém com a mesma matriz dos coeficientes. Primeiro, o método calcula a direção *afim-escala*, $(\Delta_a x^k, \Delta_a y^k, \Delta_a z^k)$, também chamada de *direção preditora*, resolvendo o sistema

$$\begin{bmatrix} A & 0 & 0 \\ 0 & A^T & I \\ Z^k & 0 & X^k \end{bmatrix} \begin{bmatrix} \Delta_a x^k \\ \Delta_a y^k \\ \Delta_a z^k \end{bmatrix} = \begin{bmatrix} r_p^k \\ r_d^k \\ r_a^k \end{bmatrix}, \quad (2.3.3)$$

onde $r_p^k = b - Ax^k$, $r_d^k = c - A^T y^k - z^k$ e $r_a^k = -X^k Z^k e$.

Em seguida, o lado direito do sistema (2.3.3) é modificado fazendo $r_p^k = r_d^k = 0$ e substituindo r_a^k por

$$r_c^k = \mu^k e - \Delta_a X^k \Delta_a Z^k e,$$

onde o número μ^k é o parâmetro de centragem, $\Delta_a X^k = \text{diag}(\Delta_a x^k)$ e $\Delta_a Z^k = \text{diag}(\Delta_a z^k)$. O sistema obtido fica da forma

$$\begin{bmatrix} A & 0 & 0 \\ 0 & A^T & I \\ Z^k & 0 & X^k \end{bmatrix} \begin{bmatrix} \Delta_c x^k \\ \Delta_c y^k \\ \Delta_c z^k \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ r_c^k \end{bmatrix}, \quad (2.3.4)$$

e com a solução desse novo sistema obtém-se a chamada *direção corretora de centragem*, $(\Delta_c x^k, \Delta_c y^k, \Delta_c z^k)$.

Por fim, a direção de busca $(\Delta x, \Delta y, \Delta z)$ é obtida da soma das duas direções anteriores. Isto é,

$$(\Delta x^k, \Delta y^k, \Delta z^k) = (\Delta_a x^k, \Delta_a y^k, \Delta_a z^k) + (\Delta_c x^k, \Delta_c y^k, \Delta_c z^k).$$

Porém, para evitar essa soma, as direções de busca são calculadas diretamente resolvendo o sistema (2.3.3) substituindo r_a^k por

$$\mu^k e - \Delta_a X^k \Delta_a Z^k e - X^k Z^k e,$$

o qual será denotado por r_m^k . Em outras palavras, resolve-se o seguinte sistema de equações

$$\begin{bmatrix} A & 0 & 0 \\ 0 & A^T & I \\ Z^k & 0 & X^k \end{bmatrix} \begin{bmatrix} \Delta x^k \\ \Delta y^k \\ \Delta z^k \end{bmatrix} = \begin{bmatrix} r_p^k \\ r_d^k \\ r_a^k + r_c^k \end{bmatrix}. \quad (2.3.5)$$

2.3.0.2 Direções de Busca

Nos métodos de pontos interiores do tipo primal-dual, a resolução de sistemas lineares como (2.3.3) é o passo computacionalmente mais caro do método. Felizmente, neste caso, os dois sistemas lineares a serem resolvidos em cada iteração compartilham a mesma matriz de coeficientes a qual, em geral, é de grande porte e esparsa. A estrutura da matriz do sistema (2.3.3) permite reformulá-lo de modo a obter sistemas lineares com matrizes dos coeficientes que são simétricas, mais compactas e mais fáceis de manusear do que a original. Essa reformulação é possível porque em cada ponto (x^k, y^k, z^k) as componentes (x^k, z^k) são estritamente positivas e assim as matrizes diagonais X^k e Z^k são não-singulares. Dessa forma, é possível eliminar a variável Δz^k em (2.3.3) para obter o seguinte sistema equivalente

$$\begin{bmatrix} -D^k & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} \Delta x^k \\ \Delta y^k \end{bmatrix} = \begin{bmatrix} r_d^k - (X^k)^{-1} r_a^k \\ r_p^k \end{bmatrix}, \quad (2.3.6)$$

onde

$$D^k = (X^k)^{-1} Z^k. \quad (2.3.7)$$

Após a resolução de (2.3.6) é possível calcular Δz^k usando a equação

$$\Delta z^k = (X^k)^{-1} (r_a^k - Z^k \Delta x^k).$$

Neste formato, o sistema (2.3.6) é conhecido como *sistema aumentado*. Agora, como a matriz D^k é invertível, podemos também reduzir o sistema aumentado eliminando Δx^k da primeira equação e substituindo na segunda. Assim, obtemos o seguinte sistema linear

$$A(D^k)^{-1} A^T \Delta y^k = r_p^k + A \left((D^k)^{-1} r_d^k - (Z^k)^{-1} r_a^k \right). \quad (2.3.8)$$

Esta forma é chamada de *equações normais* porque no caso de $r_p^k = 0$ o sistema (2.3.8) se reduz as equações normais para o problema de quadrados mínimos com matriz de coeficientes $(D^k)^{-1/2} A^T$.

É importante observar que D^k é uma matriz diagonal com entradas positivas, sendo a única mudança que ocorre nas matrizes dos sistemas (2.3.6) e (2.3.8) a cada iteração do método de pontos interiores. De acordo com (2.3.7), os elementos d_{ii}^k da matriz D^k são do tipo

$$d_{ii}^k = \frac{z_i^k}{x_i^k}$$

para todo $i = 1, \dots, n$.

À medida que o método de pontos interiores se aproxima de uma solução ótima, as variáveis primais e as variáveis duais de folga convergem para os seus respectivos valores ótimos de modo que o conjunto de índices $\{1, 2, 3, \dots, n\}$ se particiona em dois subconjuntos $\mathcal{B}$ e $\mathcal{N}$, tal que,

$$x_i^k \rightarrow \tilde{x}_i > 0 \text{ e } z_i \rightarrow \tilde{z}_i = 0 \quad (2.3.9)$$

para $i \in \mathcal{B}$, enquanto

$$x_i^k \rightarrow \tilde{x}_i = 0 \text{ e } z_i \rightarrow \tilde{z}_i > 0 \quad (2.3.10)$$

para os demais índices. Isto significa que no primeiro caso, à medida que o método converge, temos $d_{ii}^k \rightarrow 0$ e no segundo $d_{ii}^k \rightarrow \infty$. A consequência imediata deste fato é que as matrizes dos sistemas (2.3.6) e (2.3.8) ficam mal condicionadas à medida que o método se aproxima de uma solução ótima. Este fato torna a resolução desses sistemas lineares muito mais difícil. Técnicas utilizadas para melhorar o condicionamento dos mesmos, de modo a tornar suas resoluções mais eficientes através de um método iterativo, serão discutidas no Capítulo 3 e são objetos centrais deste trabalho.

2.3.1 Abordagens para resolver o Sistema Linear

Deste ponto em diante, por questão de simplicidade, o índice k que indica as iterações do método de pontos interiores será suprimido. Além disso, será utilizada a notação $\Theta = (D^k)^{-1}$. Dessa forma, os sistemas (2.3.6) e (2.3.8) serão reescritos como

$$\begin{bmatrix} -\Theta^{-1} & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} r_d - X^{-1}r_a \\ r_p \end{bmatrix}, \quad (2.3.11)$$

e

$$A\Theta A^T \Delta y = r_p + A(\Theta r_d - Z^{-1}r_a). \quad (2.3.12)$$

2.3.1.1 Equações Normais

O sistema linear apresentando as equações normais, sistema (2.3.12), é usado pela maioria dos códigos de métodos de pontos interiores do tipo primal-dual porque apresenta a vantagem da matriz $A\Theta A^T$ ser simétrica positiva definida, sempre que a matriz A tem posto completo [44]. Esta é uma qualidade desejável em um sistema linear porque permite a sua resolução por meio de um método direto usando a fatoração de Cholesky ou iterativamente usando o método dos gradientes conjugados preconditionado.

Seja K a matriz dos coeficientes em (2.3.12) após um possível reordenamento simétrico de suas linhas e colunas, isto é,

$$K = P(A\Theta A^T)P^T,$$

onde P é uma matriz de permutação quadrada de ordem m . Assumindo que A tem posto completo, a matriz K é simétrica positiva definida e a sua fatoração de Cholesky produz uma matriz triangular inferior L com elementos da diagonal positivos tal que

$$K = LL^T.$$

Além do mais, L é única. O Algoritmo 2.1 descreve completamente uma variante da fatoração de Cholesky.

Algoritmo 2.1 Fatoração de Cholesky

```

para  $i = 1, \dots, m$  faça
     $l_{ii} = \sqrt{k_{ii}};$ 
    para  $j = i + 1, \dots, m$  faça
         $l_{ji} = k_{ji}/l_{ii};$ 
        para  $k = i + 1, \dots, j$  faça
             $k_{jk} = k_{jk} - l_{ji}l_{ki}$ 
        fim para
    fim para
fim para

```

Uma vez disponível a fatoração $K = LL^T$, o sistema linear $Ku = v$ pode ser resolvido efetuando duas substituições triangulares:

$$Lw = v,$$

para encontrar w , e

$$L^T u = w,$$

para encontrar u . Em casos como o de métodos de pontos interiores, nos quais a matriz dos coeficientes é simétrica e positiva definida, o processo da fatoração de Cholesky seguido por substituições triangulares é um método estável [44].

Uma matriz é dita *esparsa* quando possui, de um modo geral, um número de elementos não nulos bem inferior do que o número de elementos nulos. Esta característica, também chamada de *esparsidade*, frequentemente está presente nos sistemas lineares oriundos de problemas de programação linear de grande porte e mantê-la durante a resolução do sistema linear (2.3.12), por exemplo, é muito importante, principalmente, por causa da questão do armazenamento de dados. No entanto, após uma fatoração $K = LL^T$, quando o fator de Cholesky L é comparado com a matriz K , nota-se normalmente que a quantidade de elementos não nulos em L é superior àqueles presentes na metade triangular inferior de K . Os novos elementos não nulos que aparecem em L , nas posições anteriormente ocupadas por zeros na matriz K , são chamados de *preenchimento* ou *fill-in*. A quantidade de preenchimento que pode aparecer em uma fatoração de Cholesky depende de como as equações do sistema e, portanto, as linhas e/ou colunas da matriz estão ordenadas.

Não é difícil inferir que quanto maior for o preenchimento em uma fatoração, maior será a quantidade de armazenamento adicional demandada e o custo computacional para resolver os sistemas triangulares e calcular as matrizes L e L^T , que são necessários para recuperar o vetor solução. Dessa maneira, conclui-se que tornar uma fatoração de Cholesky esparsa mais econômica implica em manter a quantidade de preenchimento sob controle. O ideal seria encontrar um ordenamento das linhas e/ou colunas de K que minimizasse o preenchimento de L . Porém, o problema de encontrar tal ordenamento é, em geral, muito difícil de resolver. Na verdade, trata-se de um problema NP-completo. Felizmente, há várias heurísticas que funcionam muito bem na maioria dos exemplos práticos.

Uma heurística de reordenamento de matrizes muito conhecida é a heurística *mínimo grau*. Esta heurística tenta minimizar o preenchimento no fator L , analisando o padrão de esparsidade da matriz K usando métodos da Teoria dos Grafos. Mais detalhes sobre esta e outras técnicas de reordenamento de matrizes podem ser obtidos em [44].

Como mencionado anteriormente, a fatoração de Cholesky seguida de substituições triangulares encontra soluções precisas para sistemas lineares bem condicionados. No caso do método de pontos interiores, a matriz $A\Theta A^T$ no sistema (2.3.12) pode não ser bem condicionada e pivôs muito pequenos (ou ainda pivôs nulos ou negativos) podem ser encontrados durante a fatoração numérica. O mau condicionamento dessa matriz tem origens no fato de que, sendo $\Theta = X^{-1}Z$ uma matriz diagonal, a matriz dos coeficientes é uma soma, com pesos $\frac{z_i}{x_i}$, de produtos externos de colunas de A . Isto é,

$$A\Theta A^T = \sum_{i=1}^n \frac{z_i}{x_i} A_{.i} A_{.i}^T. \quad (2.3.13)$$

Como já mencionado, à medida que o método de pontos interiores se aproxima de uma solução, alguns dos quocientes $z_i/x_i \rightarrow \infty$, enquanto outros $z_i/x_i \rightarrow 0$. Isto implica que a norma de $A\Theta A^T$ tende para ∞ durante as últimas iterações de métodos de pontos interiores. Ainda assim, em muitos problemas, se todos os autovalores da matriz $A\Theta A^T$ crescem para ∞ , aproximadamente na mesma razão, essa matriz permanece bem condicionada [44]. Contudo, frequentemente, a matriz $A\Theta A^T$ fica mal condicionada nas iterações finais do método de pontos interiores [44].

Em geral, o mau condicionamento da matriz é detectado durante a fatoração numérica através dos pivôs encontrados. Em alguns casos, por causa de erros de arredondamentos no passo anterior, pivôs negativos podem ser encontrados, apesar da matriz $A\Theta A^T$ ser definida positiva. Quando isso acontece, a fatoração de Cholesky falha, pois não é permitido o cálculo da raiz quadrada de um número real negativo. Nesse sentido, o algoritmo precisa ser modificado para assegurar que uma boa direção de busca $(\Delta x, \Delta y, \Delta z)$ possa ser calculada na presença de erros de arredondamentos e ainda que a matriz $A\Theta A^T$ seja mal condicionada.

Para contornar o problema do mau condicionamento muitas técnicas, todas motivadas por experiências computacionais, têm sido propostas e implementadas [44]. Por exemplo, o *solver* de problemas de programação linear PCx, que é utilizado neste trabalho, substitui cada pivô pequeno pelo número 10^{128} quando isso acontece. Esta substituição torna os elementos fora da diagonal, na i -ésima coluna do fator de Cholesky L , extremamente pequenos. Também como consequência dessa ação a i -ésima componente do vetor solução fica extremamente pequena [18].

A possibilidade da matriz $A\Theta A^T$ ser muito mais densa do que a matriz A é uma outra dificuldade da abordagem do cálculo das direções de buscas via equações normais. Para isto acontecer, basta que apenas uma coluna de A seja densa. Quando isso ocorre, a fatoração de Cholesky pode ser um meio ineficiente para a solução do Sistema (2.3.12). Contudo, se A não contém muitas colunas densas, a eficiência da técnica pode ser recuperada por meio de uma modificação no algoritmo. As colunas densas são excluídas do cálculo de $A\Theta A^T$ e a fórmula de Sherman-Morrison-Woodbury é utilizada no processo. Mais detalhes sobre a técnica pode ser encontrada em [44].

Considerando as desvantagens citadas anteriormente, uma alternativa ao uso da fatoração de Cholesky na resolução do sistema de equações normais é o uso de um método iterativo, os quais são adequados para resolver sistemas lineares de grande porte e esparsos. No caso dos métodos

de pontos interiores, o método dos *gradientes conjugados* pode ser a escolha apropriada devido a sua eficiência para matrizes simétricas definidas positivas, que é o caso do sistema (2.3.12). Neste método iterativo, a operação produtos de matrizes não é requerida em suas iterações e apenas produtos de matriz por vetor são necessários, conforme pode ser visto no Algoritmo 2.2. Portanto, não há a necessidade de calcular o produto $A\Theta A^T$ explicitamente e, desta forma, a presença de colunas densas na matriz A não é um problema sério para o método.

Uma outra vantagem dos métodos iterativos para solução de sistemas lineares, é que de um modo geral, eles não sofrem com efeitos de preenchimentos na matriz. Por outro lado, alguns desses métodos, como é o caso do gradientes conjugados, são sensíveis ao condicionamento da matriz do sistema. Isto é, tais métodos não são eficientes quando aplicados em matrizes mal condicionadas. Este fato poderia inviabilizar o uso do gradientes conjugados em pontos interiores, pois como visto anteriormente, o sistema linear (2.3.12) fica muito mal condicionado à medida que o método de pontos interiores se aproxima de uma solução do problema de programação linear. Todavia, se o sistema linear mal condicionado puder ser transformado em um sistema equivalente, porém melhor condicionado, o método dos gradientes conjugados ainda pode ser aplicado com sucesso. A técnica que faz essa transformação é conhecida como *precondicionamento* e será vista com detalhes no Capítulo 3. Por enquanto, serão dadas algumas informações importantes do método dos gradientes conjugados.

Desenvolvido por Hestenes e Stiefel em 1952, o método dos gradientes conjugados é o mais famoso dos métodos de subespaço de Krylov¹ e um dos pilares da computação científica [41]. Este método resolve eficientemente sistemas de equações lineares simétricos definidos positivos, se os autovalores da matriz do sistema estiverem bem distribuídos.

Como todos os métodos iterativos, o método dos gradientes conjugados geram uma sequência de soluções aproximadas $\{x^k\}$ e, essencialmente, utiliza a matriz do sistema apenas em operações de multiplicação de matriz por vetor. A avaliação do método é estabelecida sobre o quão rapidamente os iterandos x^k convergem. Para um melhor entendimento desta análise, suponha que $\mathcal{A}$ é uma matriz simétrica definida positiva de ordem n , $b \in \mathbb{R}^n$ e $x_0 \in \mathbb{R}^n$ é um ponto inicial dado e tal que $\mathcal{A}x_0 \simeq 0$, então o Algoritmo 2.2 calcula $x \in \mathbb{R}^n$ tal que $\mathcal{A}x = b$. Observe que a cada iteração k , um único produto de matriz por vetor é realizado e há a necessidade de armazenamento de apenas quatro vetores n -dimensionais: x, r, p e w . Resultados teóricos estabelecem que, em aritmética exata, o método dos gradientes conjugados encontra a solução do sistema linear em no máximo n iterações. Contudo, quando a distribuição dos autovalores de $\mathcal{A}$ tem certas características favoráveis, o método pode encontrar a solução em um número de iterações bem menor do que n . Na verdade, sabe-se que se $\mathcal{A}$ possui r autovalores distintos, então o método de gradientes conjugados encontrará a solução do sistema em no máximo r iterações [23].

Outro resultado teórico que fornece uma caracterização útil para a análise do comportamento dos gradientes conjugados é dado pela seguinte desigualdade:

$$\|x^k - x^*\|_{\mathcal{A}} \leq 2 \left(\frac{\sqrt{\kappa_2(\mathcal{A})} - 1}{\sqrt{\kappa_2(\mathcal{A})} + 1} \right) \|x^0 - x^*\|_{\mathcal{A}}, \quad (2.3.14)$$

¹Dado um vetor v , um inteiro positivo j e uma matriz simétrica $\mathcal{A}$, o subespaço de Krylov $\mathcal{K}(\mathcal{A}, v)$ é definido como sendo o subespaço $\mathcal{K}(\mathcal{A}, v) = \text{span}\{v, \mathcal{A}v, \mathcal{A}^2v, \dots, \mathcal{A}^{j-1}v\}$. A teoria desses subespaços e os métodos iterativos baseados no mesmo podem ser consultados em [41], [23] e [39]

onde $\kappa_2(\mathcal{A}) = \|\mathcal{A}\|_2 \|\mathcal{A}^{-1}\|_2$ é o número de condição da matriz $\mathcal{A}$ na norma Euclidiana e a norma matricial $\|\cdot\|_{\mathcal{A}}$ está definida por $\|w\|_{\mathcal{A}} = \sqrt{w^T \mathcal{A} w}$ para todo $w \in \mathbb{R}^n$.

A precisão de $\{x^k\}$ é frequentemente bem melhor que a prevista pelo limitante em (2.3.14). Este limitante superior descreve bem o comportamento da convergência do método para matrizes $\mathcal{A}$ cuja distribuição dos autovalores é bastante homogênea [41]. De fato, tem-se

$$\frac{\sqrt{\kappa_2(\mathcal{A})} - 1}{\sqrt{\kappa_2(\mathcal{A})} + 1} \simeq 1 - \frac{2}{\sqrt{\kappa_2(\mathcal{A})}}$$

à medida que $\sqrt{\kappa_2(\mathcal{A})} \rightarrow \infty$. Logo, se $\kappa_2(\mathcal{A})$ é grande, mas não é muito grande, a convergência do método para a tolerância especificada é da ordem de $\sqrt{\kappa_2(\mathcal{A})}$ iterações. Contudo, este é apenas um limite superior, e a convergência pode ser mais rápida, se por exemplo, os autovalores de $\mathcal{A}$ estão agrupados [41]. Em outras palavras, podemos dizer que o método funciona bem quando $\mathcal{A}$ for bem condicionada ou possui autovalores agrupados.

O número de condição de uma matriz será recorrente ao longo deste trabalho. Sendo assim, é válido mencionar aqui que de maneira geral, o número de condição de uma matriz quadrada $\mathcal{A}$ é definido como sendo o número $\kappa(\mathcal{A}) = \|\mathcal{A}\| \|\mathcal{A}^{-1}\|$, independentemente da norma $\|\cdot\|$ utilizada. Diz-se que uma matriz $\mathcal{A}$ está mal condicionada quando $\kappa(\mathcal{A})$ é um número grande. Além disso, observa-se que $\kappa(\mathcal{A}) \geq 1$ e diz-se que uma matriz é bem condicionada quando o seu número de condição é pequeno. Como os conceitos de grande e pequeno são relativos, é importante observar que quanto mais próximo da unidade for o número de condição, melhor condicionada será a matriz.

Algoritmo 2.2 Método dos Gradientes Conjugados

```

 $x = x_0, k = 0, r = b - \mathcal{A}x_0, \rho_0 = \|r\|_2^2;$ 
enquanto ( $\sqrt{\rho_k} > \epsilon \|b\|_2$ ) e ( $k < k_{max}$ ) faça
   $k = k + 1$ 
  se  $k = 1$  então
     $p = r$ 
  senão
     $\beta_k = \rho_{k-1} / \rho_{k-2}$ 
     $p = r + \beta_k p$ 
  fim se
   $w = \mathcal{A}p$ 
   $\alpha_k = \rho_{k-1} / p^T w$ 
   $x = x + \alpha_k p$ 
   $r = r - \alpha_k w$ 
   $\rho_k = \|r\|_2^2$ 
fim enquanto

```

A simplicidade de uma iteração do método dos gradientes conjugados pode ser visualizada no Algoritmo 2.2. A cada iteração, são efetuadas apenas operações com vetores e um produto de matriz por vetor, $w = \mathcal{A}p$, que, em geral, é o passo mais caro do método.

No Capítulo 3, serão apresentadas técnicas para converter o sistema linear $\mathcal{A}x = b$ em um outro sistema equivalente $\tilde{\mathcal{A}}\tilde{x} = \tilde{b}$ com a matriz $\tilde{\mathcal{A}}$ próxima da matriz identidade ou, equivalentemente, melhor condicionada do que a matriz $\mathcal{A}$.

2.3.1.2 Sistema Aumentado

O sistema aumentado (2.3.11) é um sistema simétrico e indefinido de ordem $m + n$, portanto, bem maior que o sistema de equações normais. No entanto, apresenta algumas vantagens em relação a este último, em robustez e flexibilidade [44]. Por exemplo, colunas densas não são um problema, pois produtos do tipo $A_{.i}A_{.i}^T$ não são efetuados *a priori*, logo algoritmos de fatoração para (2.3.3) não precisam incluir nenhum artifício para tratar de colunas densas. O mau condicionamento no sistema aumentado, ao longo das iterações do método de pontos interiores, é menos agressivo do que o que ocorre nas equações normais, logo é mais fácil administrar os efeitos do mesmo. E por fim, o sistema aumentado suporta mais facilmente extensões para formulações alternativas do problema de programação linear (2.2.1). Em particular, formulações que contém variáveis livres, que são aquelas componentes de x que não possuem limitantes inferiores ou superiores, são facilmente acomodadas no sistema aumentado. Além do mais, em caso de soluções iterativas, existe uma classe mais ampla de preconditionadores disponíveis para o sistema aumentado [38, 3].

Como desvantagens, essa formulação apresenta o fato de que algoritmos e softwares para resolver sistemas simétricos indefinidos não são muito desenvolvidos ou facilmente disponíveis, como são os códigos para fatoração de Cholesky esparsa. Além do mais, obter a direção de busca $(\Delta x, \Delta y, \Delta z)$ a partir do sistema aumentado, pode ter um custo computacional muito maior do que aquele obtido de fatoração de Cholesky.

A resolução via método direto do sistema (2.3.6), pode ser calculada por meio de uma das duas técnicas apresentadas a seguir. A primeira é um método desenvolvido por Aansen [2] e calcula a fatoração

$$P \begin{bmatrix} -\Theta^{-1} & A^T \\ A & 0 \end{bmatrix} P^T = LTL^T,$$

onde L é triangular inferior com diagonal unitária e T é uma matriz tridiagonal. A matriz de permutação P é escolhida de modo que se tenha $|l_{ij}| \leq 1$. A segunda técnica possível é o método do pivoteamento diagonal desenvolvido por Bunch e Parlett [13] que calcula uma matriz de permutação P tal que

$$P \begin{bmatrix} -\Theta^{-1} & A^T \\ A & 0 \end{bmatrix} P^T = L\Lambda L^T,$$

onde Λ é uma matriz bloco diagonal indefinida formada por pivôs de tamanho 1×1 ou 2×2 . Da mesma forma que no caso anterior, a matriz P deve ser escolhida de modo a obter $|l_{ij}| \leq 1$. Maiores detalhes sobre fatorações de sistemas simétricos e indefinidos podem ser encontradas em [23].

Iterativamente, o sistema aumentado pode ser resolvido utilizando métodos de subespaços de Krylov como o MINRES. Sistemas simétricos e indefinidos, como o sistema aumentado, aparecem em uma grande variedade de aplicações em computação científica e engenharia. Um exemplo é a

resolução numérica de equações diferenciais parciais, e em particular o problema de ponto de sela. Por essa razão várias técnicas para resolução de sistemas deste tipo têm sido desenvolvidas e uma vasta seleção delas está apresentada em [9].

No próximo capítulo, trataremos de técnicas que aceleram a convergência de métodos iterativos que são utilizados para a resolução dos sistemas lineares discutidos aqui, em particular, o sistema de equações normais. Trata-se do condicionamento de sistemas lineares.

Capítulo 3

Técnicas de Precondicionamento de Sistemas Lineares

3.1 Introdução

Como mencionado no Capítulo 2, métodos iterativos podem ser utilizados com sucesso para a resolução dos sistemas lineares oriundos do método de pontos interiores para o problema de programação linear. No caso das equações normais, onde a matriz $A\Theta A^T$ é simétrica positiva definida, o método dos gradientes conjugados é a escolha natural. No entanto, sabe-se que a matriz diagonal Θ fica muito mal condicionada ao longo das iterações do método de pontos interiores, causando também o mau condicionamento da matriz $A\Theta A^T$. Por outro lado, o método dos gradientes conjugados não têm um bom desempenho quando utilizado com matrizes mal condicionadas, o que pode ser traduzido em lentidão ou até mesmo falha na convergência do método. Dessa forma, se faz necessário o uso de técnicas que permitam ao método efetuar a resolução do sistema linear, ainda que mal condicionado, de modo eficiente.

A técnica utilizada para melhorar o condicionamento de uma matriz, é chamada de *precondicionamento*. Na prática, o que se faz é substituir o sistema linear que tem uma matriz mal condicionada por outro sistema linear, mas cuja matriz dos coeficientes possui um número de condição menor do que o da anterior. Em outras palavras, o novo sistema deve ter melhores propriedades espectrais, como por exemplo, autovalores próximos da unidade. De fato, a convergência do método dos gradientes conjugados, depende da distribuição dos autovalores da matriz. Por essa razão, reconhecidamente, o precondicionamento é uma das ferramentas mais importantes no desenvolvimento de resolvidores eficientes para problemas difíceis em computação científica [8, 41, 39]. Neste capítulo, será apresentada uma breve exposição sobre as principais técnicas de precondicionamento. Devido aos objetivos deste trabalho, as fatorações incompletas serão enfatizadas. O capítulo é encerrado com a apresentação de resultados relevantes e atuais sobre o precondicionamento dos sistemas lineares originados em métodos de pontos interiores.

3.2 Precondicionamento para $\mathcal{A}x = b$

Considere o sistema linear

$$\mathcal{A}x = b, \quad (3.2.1)$$

onde $\mathcal{A} = [a_{ij}]$ é uma matriz quadrada de ordem n e os vetores x e $b \in \mathbb{R}^n$. O termo precondicionamento se refere ao processo de transformar o sistema linear (3.2.1) em um outro sistema equivalente, mas com melhores propriedades para ser resolvido por meio de um método iterativo. Um *precondicionador* é uma matriz que faz tal transformação.

Suponha que M seja uma matriz quadrada de ordem n e não singular. O sistema linear

$$M^{-1}\mathcal{A}x = M^{-1}b \quad (3.2.2)$$

tem a mesma solução de (3.2.1). Além disso, se de alguma forma, M é uma matriz próxima da matriz $\mathcal{A}$, espera-se que a matriz $M^{-1}\mathcal{A}$ tenha um número de condição menor do que o de $\mathcal{A}$ ou ainda autovalores agrupados em torno da unidade. Dessa forma, o sistema precondicionado se torna mais fácil de resolver do que o sistema (3.2.1). É válido mencionar que na prática, a matriz M^{-1} não é calculada explicitamente. Em vez disso, resolve-se um sistema de equações lineares, conforme pode ser visto no Algoritmo 3.1, onde a cada iteração k o sistema $Mz_k = r_k$ deve ser resolvido.

Algoritmo 3.1 Método dos Gradientes Conjugados Precondicionado

$x = x_0, k = 0, r_0 = b - \mathcal{A}x_0$ e um precondicionador M ;

enquanto ($r_k \neq 0$) e ($k < k_{max}$) **faça**

 Resolva $Mz_k = r_k$

$k = k + 1$

se $k = 1$ **então**

$p_1 = z_0$

senão

$\beta_k = r_{k-1}^T z_{k-1} / r_{k-2}^T z_{k-2}$

$p_k = z_{k-1} + \beta_k p_{k-1}$

fim se

$w_k = \mathcal{A}p_k$

$\alpha_k = r_{k-1}^T / p_k^T w$

$x_k = x_{k-1} + \alpha_k p_k$

$r_k = r_{k-1} - \alpha_k w$

fim enquanto

$x = x_k$

O método dos gradientes conjugados precondicionado, apresentado no Algoritmo 3.1, resolve o sistema linear (3.2.1) quando a matriz $\mathcal{A}$ é simétrica definida positiva e, no caso deste método, o precondicionador M também deve ser uma matriz simétrica definida positiva. Para uma boa eficiência do método é preciso que o sistema linear $Mz = r$ seja fácil de resolver. Assim, a escolha de um bom precondicionador M desempenha um papel crucial na convergência do método.

Sobre a escolha de um bom preconditionador M para ser utilizado com o método dos gradientes conjugados preconditionado é pertinente considerar os preconditionadores $M = \mathcal{A}$ e $M = \text{diag}(\mathcal{A})$. Teoricamente, o preconditionador $M = \mathcal{A}$ possui uma excelente propriedade, pois neste caso, após o preconditionamento, obtêm-se $M^{-1}\mathcal{A} = I_n$. Dessa forma, a matriz preconditionada teria o número de condição $\kappa(A)$ igual a 1 e o método dos gradientes conjugados convergiria em apenas uma iteração. Infelizmente, na prática a situação é outra, pois utilizar a própria matriz $\mathcal{A}$ como preconditionador significa resolver um sistema linear do tipo $\mathcal{A}z = r$, o que não traz nenhum ganho em relação a resolver o sistema original. No outro extremo, temos o preconditionador $M = \text{diag}(\mathcal{A})$, cujos elementos são exatamente aqueles da diagonal da matriz $\mathcal{A}$, em suas respectivas posições. Este preconditionador é conhecido como *escala diagonal* ou como *preconditionador de Jacobi*. Embora seja um preconditionador bem simples, de fato o mais simples que existe, o mesmo pode ainda assim produzir resultados satisfatórios, como pode ser visto em [41]. Uma análise dos dois preconditionadores anteriores dá uma ideia sobre o caminho a ser seguido com vistas à construção de preconditionadores ideais para o sistema linear $\mathcal{A}x = b$: uma combinação entre a robustez de $M = \mathcal{A}$ e o baixo custo computacional do preconditionador escala diagonal. Um balanço entre essas duas qualidades é desejável na construção de bons preconditionadores.

O preconditionamento do sistema (3.2.1) também pode ser realizado pelo lado direito fazendo $x = M^{-1}y$ e

$$\mathcal{A}M^{-1}y = b. \quad (3.2.3)$$

Outra forma usual no preconditionamento de sistemas lineares consiste na construção do preconditionador M usando fatores M_1 e M_2 tais que $M = M_1M_2$. Em seguida, o sistema original (3.2.1) é pré-multiplicado por M^{-1} e pós-multiplicado pelo fator M^{-T} , obtendo o seguinte sistema preconditionado

$$[M_1^{-1}\mathcal{A}M_2^{-T}]M_2^Tx = M_1^{-1}b. \quad (3.2.4)$$

A escolha do tipo de preconditionamento a ser utilizado depende, por exemplo, do método iterativo utilizado ou das características do problema. Técnicas de preconditionamento do tipo (3.2.4) são úteis quando a matriz $\mathcal{A}$ é simétrica, pois é preferível manter esta propriedade no preconditionamento. Neste caso, toma-se uma fatoração de M tal que $M_2 = M_1^T$ de modo que a matriz $M_1^{-1}\mathcal{A}M_1^{-T}$ no sistema preconditionado

$$[M_1^{-1}\mathcal{A}M_1^{-T}]M_1^Tx = M_1^{-1}b. \quad (3.2.5)$$

ainda é simétrica.

Como mencionado anteriormente, o método dos gradientes conjugados não calcula as matrizes preconditionadas $M^{-1}\mathcal{A}$, $M_1^{-1}\mathcal{A}M_1^{-T}$ ou $\mathcal{A}M^{-1}$ explicitamente. Com efeito, se isso fosse necessário, aumentaria consideravelmente o custo computacional do método e também poderia implicar na perda da esparsidade da matriz $\mathcal{A}$. Em vez de construir a matriz M^{-1} , o método realiza a resolução de sistemas lineares do tipo $Mz = r$.

O grande desafio, quando se trata de preconditionamento, é a construção de preconditionadores que, por um lado, efetivamente acelerem a convergência do método iterativo, mas que por outro, não necessite de um grande esforço computacional para serem construídos e nem para serem utilizados. Em outras palavras, deseja-se construir preconditionadores que tornem o sistema preconditionado mais fácil de resolver, mas que ao mesmo tempo seja barato de construir e aplicar.

Na análise desta questão, não pode ser desprezada a possibilidade de reutilização ou não do preconditionador. Pois no caso afirmativo, pode-se pensar em gastar mais tempo e esforço computacional na construção de um preconditionador eficiente, se esse esforço pode ser compensado na resolução rápida e confiável dos vários sistemas lineares. Este fato é interessante para o método de pontos interiores, versão preditor-corretor, onde há a necessidade de resolver dois sistemas lineares com a mesma matriz dos coeficientes a cada iteração.

3.2.1 Fatorações Incompletas

A maioria das técnicas de preconditionamento atuais fazem referências, de alguma forma, à fatorações incompletas. Isso se deve, em primeiro lugar, ao fato de que preconditionadores provenientes de fatorações incompletas funcionam bem para uma grande variedade de problemas, e em segundo lugar, a simplicidade com que os mesmos podem ser construídos.

Se $\mathcal{A}$ é uma matriz simétrica positiva definida, uma decomposição triangular de $\mathcal{A}$ é dada por

$$\mathcal{A} = LDL^T$$

sendo $D = \text{diag}(d_1, \dots, d_n)$, $d_i > 0$ e o fator L triangular inferior com diagonal unitária. A *fatoração incompleta* de $\mathcal{A}$ é um método pelo qual obtemos uma matriz M tal que

$$M = \tilde{L}\Delta\tilde{L}^T$$

e $\mathcal{A} = M - R$. A matriz R é chamada de *matriz resto*, $\Delta = \text{diag}(\delta_1, \dots, \delta_n)$ e $\tilde{L}$ é uma matriz triangular inferior com diagonal unitária. Em outras palavras, a matriz M é uma aproximação da matriz $\mathcal{A}$. Durante a fatoração de $\mathcal{A}$ alguns elementos são retirados do fator L para obter o fator $\tilde{L}$. Fazendo $\hat{L} = \tilde{L}\Delta^{1/2}$ obtém-se $M = \hat{L}\hat{L}^T$, a qual é denominada de *Fatoração de Cholesky Incompleta*.

Embora a fatoração de Cholesky incompleta tenha sido introduzida nos anos 50 [31, 39], foi nas décadas de 60 e 70 que a técnica começou a receber mais atenção da comunidade científica e, conseqüentemente, a ser aprimorada, especialmente para o uso com matrizes originadas em aproximações por diferenças finitas para operadores elípticos [39]. Merece destaque o trabalho de Meijerink e Van der Vorst [33] que introduz a fatoração LU incompleta, mostra que para uma classe particular de matrizes, as M -matrizes¹, a fatoração incompleta de Cholesky é tão estável quanto a fatoração de Cholesky e, propõe uma combinação desta com o método dos gradientes conjugados, cujos resultados numéricos foram bons o suficiente para que se chegasse a conclusão de que tal combinação poderia ser competitiva com os métodos diretos, ainda que estes tirassem proveito da esparsidade das matrizes. Outro mérito desse trabalho foi o de estimular a produção de novos trabalhos na área de métodos iterativos [39].

Preenchimento é uma das grandes desvantagens de métodos baseados na eliminação de Gauss, como as fatorações LU e de Cholesky. Na fatoração LU , por exemplo, a consequência desse efeito é que os fatores L e U podem se tornar bem densos, o que implica na necessidade de mais espaço de armazenamento e maior custo computacional para resolução do sistema linear. Técnicas de

¹Uma matriz $A = [a_{ij}]$ é uma M -matriz se $a_{ij} \leq 0$ para $i \neq j$, A é não singular e A^{-1} tem todos os elementos não negativos.

pivoteamento e reordenação da matriz podem ser utilizadas para reduzir o preenchimento desses fatores, porém, ainda assim, em sistemas lineares de grande porte, esses métodos demandam um alto custo computacional de tempo e armazenamento, tornando a abordagem direta desvantajosa. Contudo, métodos diretos podem ser bem aproveitados para melhorar o desempenho dos métodos iterativos, quando utilizados como base para a construção de preconditionadores. A ideia baseia-se no fato de que durante o processo da fatoração de Cholesky, por exemplo, os preenchimentos que aparecem no fator L , podem ser descartados resultando em um fator aproximado $\tilde{L}$. Assim, a matriz $M = \tilde{L}\tilde{L}^T$ é tal que $M \simeq \mathcal{A}$ e pode ser utilizada como preconditionador para $\mathcal{A}x = b$, pois $L \simeq \tilde{L}$ implica em

$$\tilde{L}^{-1}\mathcal{A}\tilde{L}^{-T} = \tilde{L}^{-1}LL^t\tilde{L}^{-T} \simeq I_n,$$

e, portanto, $\tilde{L}$ pode ser um excelente preconditionador para o sistema $\mathcal{A}x = b$.

3.2.1.1 Preenchimentos em fatorações incompletas

Permitir algum preenchimento no fator $\tilde{L}$, durante o cálculo do preconditionador $M = \tilde{L}\tilde{L}^T$ por meio de fatorações incompletas, pode reduzir significativamente o número de iterações necessárias para a convergência do método dos gradientes conjugados. Contudo, isso pode aumentar o tempo de resolução do sistema linear. Além disso, prever inicialmente quanto preenchimento permitir, como e onde eles serão permitidos são questões importantes, porém nada fáceis de realizar. A previsão *a priori* de quanto preenchimento será permitido é importante porque implica no conhecimento prévio da quantidade de espaço requerida para o armazenamento do preconditionador. Por outro lado, o critério de aceite e a posição do preenchimento podem ser considerados de modo a melhorar o desempenho do preconditionador. Por exemplo, pode-se optar por armazenar e utilizar apenas preenchimentos que contribuam significativamente para a qualidade do preconditionador [8].

Atentando para essas questões, várias estratégias foram propostas com o objetivo de aumentar a eficiência de preconditionadores construídos com base em fatorações incompletas. Basicamente duas regras estão bem estabelecidas quanto ao modo como os preenchimentos são aceitos ou descartados durante o processo de fatoração [8]. Enquanto numa delas, o descarte dos preenchimentos são realizados com base na posição dos elementos não nulos na matriz original, a outra considera o tamanho numérico do preenchimento. Um exemplo interessante do primeiro caso, é aquele onde o padrão de esparsidade da matriz A é considerado para determinar quais preenchimentos deverão ser descartados. Para ilustrar tal exemplo, considere o conjunto $n = \{1, 2, \dots, n\}$ e suponha que $\mathcal{S}$ seja um subconjunto de posições (i, j) da matriz $\mathcal{A}$ para as quais $a_{ij} \neq 0$. Um passo formal de uma fatoração incompleta que preserva o padrão de esparsidade da matriz original, é como o descrito a seguir:

$$a_{ij} \leftarrow \begin{cases} a_{ij} - a_{ik}a_{kk}^{-1}a_{kj}, & \text{se } (i, j) \in \mathcal{S}, \\ a_{ij}, & \text{caso contrário.} \end{cases}$$

para cada $k = 1, \dots, n$ e para $i, j > k$. Se $\mathcal{S}$ coincide com o conjunto de posições no qual estão todos os elementos não-nulos de $\mathcal{A}$, obtém-se a *fatoração LU incompleta sem preenchimento*. Se a matriz A é simétrica positiva definida obtém-se, de modo análogo, a *fatoração de Cholesky incompleta sem preenchimento*. Neste caso, os preconditionadores são calculados de modo a ficar com elementos

não nulos, independentemente do tamanho, apenas nas posições que correspondem a elementos não nulos na matriz $\mathcal{A}$ e o espaço requerido para armazenamento do fator é o mesmo da matriz $\mathcal{A}$. Tais preconditionadores são simples de implementar, baratos do ponto de vista computacional e melhoram significativamente a resolução de sistemas lineares oriundos de vários tipos de problemas importantes que resultam, por exemplo, em matrizes diagonalmente dominantes. Contudo, para outras classes de problemas, fatorações incompletas sem preenchimentos resultam apenas em aproximações grosseiras de $\mathcal{A}$, e então preenchimentos precisam ser considerados para a construção de preconditionadores mais sofisticados [8].

No segundo caso, um preenchimento é aceito ou descartado durante a fatoração com base em seu valor numérico. Embora ainda não se tenha nenhum entendimento conclusivo sobre a relação entre o tamanho do elemento descartado e o número de iterações necessárias para a convergência do método, em geral espera-se que o descarte de elementos de menor valor podem produzir melhores preconditionadores do que o descarte daqueles de maior valor [39].

Na técnica chamada **retirada por tolerância** (*drop tolerance*) elementos não nulos são aceitos no fator incompleto se eles são maiores que alguma tolerância limite τ dada. Por exemplo, preenchimentos podem ser aceitos se são maiores do que τ em valor absoluto. A desvantagem desta abordagem é que não é fácil escolher um valor ótimo para a tolerância τ . Em muitos casos, valores de τ escolhidos no intervalo $[0, 0001; 0, 01]$ têm obtido bons resultados [8]. Outro exemplo simples, proposto por Munksgaard, é fixar a tolerância para o elemento a_{ij} como sendo $c\sqrt{a_{ii}a_{jj}}$, onde c é uma constante [35]. Nessa abordagem, nem o padrão de esparsidade nem o número de elementos não nulos podem ser determinados a priori. Isso implica na impossibilidade de prever a quantidade de memória necessária para o armazenamento dos fatores incompletos.

Além de determinar critérios, como os citados anteriormente, para aceite ou descarte de preenchimento no fator incompleto, existe outra questão importante que deve ser considerada durante a construção de preconditionadores com base em fatorações incompletas. Trata-se de administrar a quantidade de preenchimentos a serem permitidos nos fatores incompletos. Uma possibilidade é fixar, *a priori*, o número de elementos não nulos, que se deseja obter, no fator incompleto. Esta técnica conhecida como *fixed fill-in* predetermina o padrão de elementos não nulos do fator incompleto $\tilde{L}$, que necessariamente não é o mesmo da matriz $\mathcal{A}$. Por exemplo, pode-se requerer que nenhum preenchimento ocorra, o que é equivalente à fatoração de Cholesky Incompleta sem preenchimento, ou determinar que um número p de preenchimentos seja aceito em cada coluna do fator incompleto, durante a fatoração, que é o caso da técnica *improved incomplete Cholesky factorization* proposta por Jones e Plassmann [25]. Eles propuseram uma variante da fatoração de Cholesky incompleta que não requer a seleção de uma retirada por tolerância e possui a vantagem de prever a quantidade necessária de armazenamento para o fator $\tilde{L}$. Nessa abordagem, um número fixo n_k de elementos não nulos é permitido na k -ésima linha (ou coluna) do fator $\tilde{L}$. O número n_k é na verdade a quantidade de elementos não nulos originalmente presentes na k -ésima linha (ou coluna) da parte triangular inferior da matriz $\mathcal{A}$. O padrão de esparsidade de $\mathcal{A}$ é ignorado e apenas os maiores elementos não nulos, obtidos com a fatoração, são mantidos no fator incompleto. Desta forma, o preconditionador obtido tem a mesma quantidade de armazenamento da fatoração de Cholesky incompleta sem preenchimentos, porém com melhores propriedades de convergência em muitos casos. O método tem as vantagens de não aumentar o custo da resolução do sistema linear triangular e também de ser um método do tipo *caixa preta*, ou seja, não há a necessidade de

que o usuário repasse algum parâmetro, contudo a convergência do método iterativo na resolução de problemas mais difíceis de resolver, pode ser prejudicada pelo uso do mesmo. Uma extensão natural dessa técnica seria permitir preenchimentos adicionais no fator $\tilde{L}$. Isso foi estudado em [29], onde para um número inteiro $p \geq 0$ dado, o número de elementos não nulos que serão preservados na k -ésima linha ou coluna de $\tilde{L}$ é $n_k + p$. Para $p = 0$ esta estratégia se reduz a de Jones e Plassmann. Em geral, a técnica pode melhorar fortemente o desempenho do preconditionador, porém à custa de um pouco mais de memória, embora esta ainda seja previsível. O valor ótimo de p depende do problema, mas o valor $p \simeq 0$ trabalha bem em muitos casos [9].

Saad e Vand der Vost [39] propõe uma estratégia que combina preenchimento fixo com a retirada por tolerância, conhecida como *dual threshold*. Esta técnica consiste em fixar uma tolerância τ e um número p de preenchimentos que serão permitidos em cada linha dos fatores LU incompletos. A cada passo do processo da eliminação Gaussiana, todos os preenchimentos que são menores do que τ vezes a norma-2 da linha corrente são descartados. Dos elementos que sobraram mantém, no máximo, os p maiores em magnitude. O preconditionador resultante é denotado por $ILU(\tau, p)$. Este preconditionador é muito eficiente porque se ele falha para um problema, quando se usa determinados τ e p , ele frequentemente terá sucesso com uma outra escolha desses parâmetros. Em geral, uma escolha menor de τ e/ou uma escolha maior de p funcionará. Um preconditionador correspondente pode ser obtido com a fatoração de Cholesky incompleta para matrizes simétricas positivas definidas.

3.2.1.2 Falhas na Diagonal

Sabe-se que para uma matriz não singular $\mathcal{A}$, cujas submatrizes principais são não singulares, existe uma única fatoração $\mathcal{A} = LU$, sendo L triangular inferior com diagonal unitária e U triangular superior. Na verdade, para toda matriz $\mathcal{A}$ não singular, existe uma matriz de permutação P tal que existe a fatoração $P\mathcal{A} = LU$. No caso de $\mathcal{A}$ ser uma matriz simétrica positiva definida, existe a fatoração de Cholesky de $\mathcal{A}$. Isto é, a matriz $\mathcal{A}$ possui a fatoração $\mathcal{A} = LL^T$, onde L é uma matriz triangular inferior com elementos da diagonal positivos. Porém, quando se trata de fatorações incompletas, a situação é um pouco mais complicada, pois ainda que a matriz $\mathcal{A}$ admita a fatoração LU , ou de modo equivalente a fatoração de Cholesky, a fatoração incompleta de $\mathcal{A}$ pode falhar devido a ocorrência de pivôs nulos, ou pivôs negativos no caso da fatoração de Cholesky [8]. Além do mais, instabilidades numéricas podem ocorrer se durante o processo de fatoração pivôs pequenos são encontrados. Estas são questões muito sérias em fatorações incompletas, pois podem interromper o processo de fatoração ou comprometer a qualidade dos preconditionadores, e serão vistas com mais detalhes no Capítulo 4.

3.3 Precondicionamento em Métodos de Pontos Interiores

Os primeiros preconditionadores eficientes para sistemas lineares oriundos de métodos de pontos interiores, foram desenvolvidos para problemas especialmente estruturados originados em otimizações de redes. Preconditionadores aplicáveis em problemas de programação linear gerais foram apresentados inicialmente por Gill et al. [22] e, em seguida, por Oliveira [37] e Oliveira e Sorensen [38]. Preconditionadores construídos com base em fatorações de Cholesky incompletas também

são utilizados no contexto de pontos interiores. Como observado no Capítulo 2, a cada iteração do método de pontos interiores precisamos resolver dois sistemas lineares com a mesma matriz dos coeficientes. Contudo, temos a liberdade de escolher entre resolver o sistema simétrico e indefinido

$$\begin{bmatrix} -\Theta & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} r_1 \\ r_2 \end{bmatrix}, \quad (3.3.1)$$

conhecido como sistema aumentado ou optarmos pelo sistema de equações normais

$$A\Theta^{-1}A^T\Delta y = r \quad (3.3.2)$$

que é simétrico positivo definido. Nas próximas seções, serão apresentados preconditionadores para o sistema de equações normais e para o sistema aumentado, bem como a abordagem de preconditionamento híbrida proposta em [12].

3.3.1 Preconditionadores para as Equações Normais

No Capítulo 2, foi visto que o sistema de equações normais tem matriz dos coeficientes simétrica definida positiva, e que o mesmo fica muito mal condicionado à medida que o método de pontos interiores se aproxima de uma solução. Assim, uma alternativa ao uso da fatoração de Cholesky para a resolução do mesmo é o método dos gradientes conjugados preconditionado. Na Seção 3.2.1, foi apresentado o preconditionador baseado na fatoração de Cholesky incompleta e várias estratégias para calculá-lo de modo a melhorar o seu desempenho. A versatilidade de tal preconditionador permite que o mesmo seja aplicável na resolução de sistemas lineares oriundos dos mais diversos problemas. No entanto, o mesmo pode funcionar melhor para uma determinada classe de problemas do que para outra. No contexto de pontos interiores, Oliveira e Sorensen [38] observaram que o preconditionador fatoração de Cholesky incompleta tem um bom desempenho nas iterações iniciais do método, mas pioram nas iterações finais quando o sistema se torna muito mal condicionado. Em outro estudo, desenvolvido por Bocanegra, Campos e Oliveira [12] verificou-se que o método de pontos interiores quando preconditionado com tal preconditionador falha para a maioria dos problemas testados, principalmente, por causa do critério utilizado para tratar falhas na diagonal e introduz uma abordagem de preconditionamento híbrido, onde um dos preconditionadores utilizados é uma variante da fatoração de Cholesky incompleta chamada de fatoração controlada de Cholesky [16]. Essa estratégia apresentou bons resultados e será apresentada na Seção 3.3.3. Na seção seguinte, será apresentado mais detalhes da fatoração controlada de Cholesky.

3.3.1.1 Fatoração Controlada de Cholesky - FCC

A Fatoração Controlada de Cholesky (*Controlled Cholesky Factorization*) foi proposta por Campos [15] em 1995 como uma variação da fatoração de Cholesky incompleta. Apesar de ter sido desenvolvida para ser utilizada na resolução de sistemas lineares positivos definidos de um modo em geral, foi inicialmente utilizada, com sucesso, no preconditionamento de sistemas lineares oriundos de equações diferenciais implícitas dependentes do tempo [16]. Conforme será apresentado na Seção 3.3.3, este preconditionador tem sido utilizado com sucesso também no preconditionamento

híbrido de sistemas lineares provenientes de métodos de pontos interiores. Na FCC, o número de elementos nas colunas do fator incompleto $\tilde{L}$ pode ser controlado por um parâmetro η , de modo que, $\tilde{L}$ poderá ter mais ou menos elementos que a matriz $\mathcal{A}$ fatorada. Esse parâmetro pode ser dado em função da quantidade de espaço disponível. Se a demanda por memória não for crítica, pode ser construído um preconditionador com mais elementos que $\mathcal{A}$, acelerando assim a convergência do método iterativo.

Seja a matriz $\mathcal{A} \in R^{n \times n}$ fatorada em

$$\mathcal{A} = LL^T = \tilde{L}\tilde{L}^T + R, \quad (3.3.3)$$

onde L é o fator obtido pela fatoração de Cholesky, $\tilde{L}$ é o fator obtido pela fatoração de Cholesky incompleta e R é a matriz resto. Usando $\tilde{L}$ como uma matriz preconditionadora para $\mathcal{A}$, obtém-se

$$\tilde{L}^{-1}\mathcal{A}\tilde{L}^{-T} = \tilde{L}^{-1}LL^T\tilde{L}^{-T} = (\tilde{L}^{-1}L)(\tilde{L}^{-1}L)^T. \quad (3.3.4)$$

Definindo $E = L - \tilde{L}$ e substituindo na igualdade (3.3.4) obtém-se

$$\tilde{L}^{-1}\mathcal{A}\tilde{L}^{-T} = (I + \tilde{L}^{-1}E)(I + \tilde{L}^{-1}E)^T.$$

Observe que quando $\tilde{L} \simeq L$ então $E \simeq 0$ e

$$\tilde{L}^{-1}\mathcal{A}\tilde{L}^{-T} \simeq I.$$

Este preconditionador assume que a matriz $\mathcal{A}$ está diagonalmente escalada de modo a obter diagonal unitária e assim torná-lo mais robusto. Duff e Meurant [20] mostraram que o número de iterações do método dos gradientes conjugados está diretamente relacionado com a norma de R .

De acordo com a equação (3.3.3) podemos expressar a matriz R da seguinte maneira

$$\begin{aligned} R &= LL^T - \tilde{L}\tilde{L}^T, \\ &= LL^T - L\tilde{L}^T + L\tilde{L}^T - \tilde{L}\tilde{L}^T, \\ &= L(L^T - \tilde{L}^T) + (L - \tilde{L})\tilde{L}^T, \\ &= L(L - \tilde{L})^T + (L - \tilde{L})\tilde{L}^T, \\ &= LE^T + E\tilde{L}^T. \end{aligned} \quad (3.3.5)$$

O preconditionador FCC é então construído com base na minimização da norma de Frobenius de E , visto que, quando $\|E\| \simeq 0$ então $\|R\| \simeq 0$.

Para um melhor entendimento da construção do preconditionador FCC, considere o problema

$$\min \|E\|_F^2 = \min \sum_{j=1}^n c_j \quad (3.3.6)$$

com $c_j = \sum_i^n (l_{ij} - \tilde{l}_{ij})^2$. Reescrevendo o problema de outro modo, obtém-se:

$$\min \sum_{j=1}^n c_j = \min \sum_{j=1}^n \left[\sum_{k=1}^{m_j+\eta} (l_{kj} - \tilde{l}_{kj})^2 + \sum_{k=m_j+\eta+1}^n (l_{kj})^2 \right]. \quad (3.3.7)$$

Na equação (3.3.7), m_j é o número de elementos não nulos abaixo da diagonal na j -ésima coluna da matriz $\mathcal{A}$, η é o número de elementos não nulos extras permitidos por coluna e n a ordem da matriz. Finalmente, para resolver o problema de minimizar a norma de E , é utilizada a seguinte heurística :

- Aumentamos o fator η , permitindo maior preenchimento. Desta forma, c_j decresce, pois o primeiro somatório em (3.3.7) contém mais elementos. Isto é o mesmo que ocorre quando níveis de preenchimento são usados nas fatorações incompletas.
- Escolhemos os maiores $m_j + \eta$ elementos de $\tilde{L}$, em valor absoluto, para η fixo. Desta forma, os maiores elementos estão na primeira soma e os menores na segunda, produzindo um fator $\tilde{L}$ ótimo, para uma determinada quantidade de armazenamento. Este esquema de minimização é semelhante ao esquema de retirada por tolerância.
- Uma coluna do preconditionador é calculada por vez, sendo armazenados os maiores elementos em valor absoluto. Dessa forma, uma melhor aproximação para o fator completo é obtida com a quantidade de memória disponível.

3.3.1.2 Principais características do preconditionador FCC

- **Escolha do elemento por valor.** A FCC não considera o padrão de esparsidade da matriz original. Os elementos não nulos armazenados no preconditionador podem ou não estar nas posições correspondentes aos da matriz original, pois os elementos são escolhidos por valor e não por posição.
- **Generalização da Improved Incomplete Cholesky Decomposition.** O preconditionador proposto por Jones e Plasmann [25] escolhe um número fixo de elementos não-nulos em cada linha ou coluna. Esse número é igual ao número de elementos não nulos da matriz original. Também armazena os maiores elementos em valores absoluto. Logo, podemos ver a FCC como uma generalização do método de Jones e Plassmann.
- **Incremento exponencial.** Assim como ocorre em fatorações incompletas de um modo geral, a fatoração controlada de Cholesky não está livre da presença de pivôs muito pequenos ou não positivos durante o processo da fatoração. Para contornar esse problema, é usado um incremento exponencial. Em [12], esse incremento é calculado da seguinte forma: quando um elemento da diagonal é muito pequeno ou não positivo, soma-se a ele, o valor dado por $\alpha_i = 5,0 \times 10^{-4} 2^i$ e recomeça a fatoração. Para muitos problemas de programação linear, como será visto no Capítulo 4, a fatoração poderá ser reiniciada muitas vezes, podendo até inviabilizar a resolução do mesmo. Ainda no Capítulo 4, será apresentada uma proposta para melhorar esse procedimento.
- **Preconditionador versátil.** Esta é uma importante característica do preconditionador, por ser aquela que o diferencia dos demais. Uma classe de preconditionadores pode ser construída levando em conta a disponibilidade de memória. O parâmetro η pode variar de $-n$ até n . Conforme apresentado na Tabela 3.1, no caso de $\eta = -n$ obtém-se o preconditionador

escala diagonal, isto significa que o preconditionador M é uma matriz diagonal, e por isso, nenhum espaço de armazenamento é usado visto que a matriz $\mathcal{A}$ é previamente escalada de modo a obter diagonal unitária. Já para o caso $\eta = 0$, obtém-se um fator de Cholesky aproximado $\tilde{L}$ que requer o mesmo espaço de armazenamento de $\mathcal{A}$, mas o padrão de esparsidade pode não ser o mesmo. Finalmente, para $\eta = n$, temos o fator de Cholesky completo L e, portanto, o armazenamento do preconditionador pode ser maior do que o da matriz $\mathcal{A}$. Na Seção 3.3.3, serão dados mais detalhes de como essa versatilidade pode ser aproveitada no preconditionamento de sistemas lineares em métodos de pontos interiores.

Tabela 3.1: Preenchimento com $\text{CCF}(\eta)$

η	M	Armazenamento
$-n$	$\text{diag}(\mathcal{A})^{-1/2}$	menos que $\text{nnz}(\mathcal{A})$
0	$\tilde{L}$	igual a $\text{nnz}(\mathcal{A})$
n	L	maior que $\text{nnz}(\mathcal{A})$

- **Armazenamento previsível.**

Cada uma das matrizes é armazenada em três vetores: um vetor de inteiros de dimensão $n+1$ para os ponteiros das colunas, um vetor de inteiros de dimensão n para índices das linhas e um vetor real de dimensão n para os elementos. Outro vetor, usado como espaço de trabalho, ainda é necessário para a construção de $\tilde{L}$. Como apenas $\eta + m_j$ elementos são mantidos em cada coluna j do preconditionador, então o espaço de armazenamento necessário para o preconditionador é previsível e controlável.

3.3.2 Precondicionadores para o Sistema Aumentado

Segundo Al-Jeiroundi, Gondzio e Hall [3] existem pelo menos quatro razões para calcular as direções de busca usando o sistema aumentado e resolvê-lo usando métodos iterativos de subespaços de Krylov. A primeira razão é que a matriz do sistema aumentado tem melhor condicionamento do que a matriz do sistema de equações normais. Em segundo lugar, esse sistema apresenta maior flexibilidade para explorar a esparsidade da matriz. Terceiro, permite a extensão para programação quadrática e programação não linear. E por fim, existe uma ampla seleção de preconditionadores disponíveis para tais sistemas. Este último argumento é amparado pela análise apresentada por Oliveira e Sorensen [38]. Eles mostraram que para cada preconditionador desenvolvido para o sistema de equações normais existe outro equivalente para o sistema aumentado, enquanto que uma classe inteira de diferentes preconditionadores para o sistema aumentado pode resultar no mesmo preconditionador para o sistema de equações normais. Dessa forma este é, de fato, um forte argumento para construir preconditionadores para o sistema aumentado. Embasados por esse resultado, Oliveira e Sorensen [38] propuseram um importante preconditionador para o sistema aumentado, o qual é chamado de *precondicionador separador* e será apresentado a seguir. Outros preconditionadores para o sistema aumentado foram propostos em [3, 10, 24, 17].

3.3.2.1 Precondicionador Separador

A construção do preconditionador separador se baseia na partição definida pelas equações (2.3.9) e (2.3.10) e que ocorre com os elementos da matriz Θ nas iterações finais dos métodos de pontos interiores, quando alguns elementos da matriz se tornam muito pequenos enquanto outros crescem indefinidamente. Essa partição é então, utilizada para construir uma matriz não singular B , que é o objeto chave na construção do preconditionador. Para exemplificar, suponha que esteja disponível, conforme (2.3.9) e (2.3.10), a partição $A = \begin{bmatrix} B & N \end{bmatrix}$, onde B é uma matriz quadrada de ordem m e não singular. Essa partição induz a separação da matriz Θ da seguinte forma

$$\Theta = \begin{bmatrix} \Theta_B & 0 \\ 0 & \Theta_N \end{bmatrix},$$

e permite reescrever a matriz do sistema aumentado do seguinte modo:

$$K = \begin{bmatrix} -\Theta_B & 0 & B^T \\ 0 & -\Theta_N & N^T \\ B & N & 0 \end{bmatrix}.$$

O *precondicionador separador* proposto para o sistema aumentado é a matriz

$$M^{-1} = \begin{bmatrix} \Theta_B^{-1/2} & 0 & \Theta_B^{1/2} B^T \\ 0 & \Theta_N^{-1/2} & 0 \\ I & 0 & 0 \end{bmatrix}.$$

Aplicando este preconditionador ao sistema aumentado, obtém-se:

$$M^{-1} K M^{-T} = \begin{bmatrix} I_m & B^{-1} \Theta_B^{1/2} N \Theta_N^{-1/2} & 0 \\ \Theta_N^{-1/2} N^T \Theta_B^{1/2} B^{-T} & I_{n-m} & 0 \\ 0 & 0 & -\Theta_B \end{bmatrix}.$$

Fazendo $W = \Theta_N^{-1/2} N^T B^{-T} \Theta_B^{1/2}$ pode-se reescrever:

$$M^{-1} K M^{-T} = \begin{bmatrix} I_m & W^T & 0 \\ W & -I_{n-m} & 0 \\ 0 & 0 & -I_m \end{bmatrix}.$$

O sistema anterior ainda pode ser reduzido a um sistema linear simétrico positivo definido envolvendo a matriz

$$I_m + W^T W \tag{3.3.8}$$

ou a matriz

$$I_{n-m} + W W^T, \tag{3.3.9}$$

e portanto, podem ser resolvidos por meio do método dos gradientes conjugados.

Observa-se que os autovalores das matrizes (3.3.8) e (3.3.9) são todos iguais ou maiores que a unidade. Além disso, suficientemente perto de uma solução ótima, as matrizes Θ_B^{-1} e Θ_N se

aproximam da matriz nula, o que significa que as entradas das matrizes W e W^T também tendem a zero. Dessa forma, as matrizes em (3.3.8) e (3.3.9) se aproximam da matriz identidade e, portanto, tanto os maiores autovalores dessas matrizes como os seus respectivos números de condição se aproximam do número um. Na construção desse preconditionador, buscou-se evitar o sistema de equações normais, mas o preço a pagar é encontrar a matriz B e usá-la na resolução dos sistemas lineares. Determinar a matriz B não é uma tarefa simples, contudo, calcular a fatoração $PB = LU$ é tipicamente mais barata que calcular a fatoração de Cholesky de BB^T [12]. Técnicas para calcular a matriz B são apresentadas em [38].

Uma importante característica do preconditionador separador é que o desempenho do mesmo melhora à medida que o método de pontos interiores avança em direção a uma solução ótima [38]. Nesse sentido, para melhorar o seu desempenho no preconditionamento da matriz K , é interessante que nas iterações iniciais do método de pontos interiores o sistema linear seja resolvido por um método direto como a fatoração de Cholesky ou que outro preconditionador mais adequado seja utilizado junto com o método dos gradientes conjugados. Na Seção 3.3.3, será apresentada uma estratégia que aproveita com sucesso essa característica do preconditionador.

Vale a pena mencionar outro preconditionador do tipo separador proposto por Al-Jeiroundi, Gondzio e Hall [3]. Os autores também exploraram a diferença de magnitude entre os elementos da matriz Θ e construíram um preconditionador E para o sistema aumentado da seguinte forma: inicialmente, os elementos de Θ foram ordenados em ordem não decrescente $\theta_1 \geq \theta_2 \geq \dots \geq \theta_n$, donde obteve-se $\theta_1^{-1} \leq \theta_2^{-1} \leq \dots \leq \theta_n^{-1}$. A vantagem dessa ordenação é que ela permite saber que quando se está suficientemente próximo de uma solução ótima, então os primeiros elementos θ_j^{-1} na lista correspondem as variáveis x_j que estão mais perto de ser diferente de zero no ótimo, e os últimos elementos na lista correspondem as variáveis que, provavelmente, são zero no ótimo. As primeiras m colunas linearmente independentes de A , quando permutadas de acordo com a ordem de θ_j^{-1} são selecionadas e uma matriz não singular B é construída com essas colunas. Assim, obtém-se a partição $A = \begin{bmatrix} B & N \end{bmatrix}$, onde B é não singular e as entradas θ_j^{-1} correspondentes às colunas de B são escolhidas com base nos menores elementos de Θ^{-1} . Logo, a matriz K fica com a seguinte configuração:

$$K = \begin{bmatrix} \Theta_B^{-1} & 0 & B^T \\ 0 & \Theta_N^{-1} & N^T \\ B & N & 0 \end{bmatrix}.$$

Mas, por construção, os elementos de Θ_B^{-1} são supostamente os menores elementos de Θ^{-1} e pode-se assumir que $\Theta_B^{-1} \rightarrow 0$. Logo, a matriz triangular por blocos e inversível

$$E = \begin{bmatrix} 0 & 0 & B^T \\ 0 & \Theta_N^{-1} & N^T \\ B & N & 0 \end{bmatrix}$$

é uma boa aproximação de K e, portanto, um preconditionador atrativo. Assim como o separador, o preconditionador E trabalha bem nas iterações finais de métodos de pontos interiores [3]. Porém, ao contrário do preconditionador separador de Oliveira e Sorensen, que reduz o sistema aumentado preconditionado aos sistemas positivos definidos (3.3.8) e (3.3.9) para serem resolvidos pelo método dos gradientes conjugados, o preconditionador E mantém, após o preconditionamento, as

dimensões da matriz do sistema aumentado indefinido, o qual é resolvido, também, com o método dos gradientes conjugados. Tanto em [38] como em [3] o uso dos preconditionadores propostos foi evitado nas iterações iniciais do método de pontos interiores. Em ambos os casos, nas primeiras iterações do método, foi utilizada a fatoração $A\Theta A^T = LDL^T$ para encontrar as direções de busca, e apenas quando a matriz $A\Theta A^T$ se torna extremamente mal condicionada, é que ocorre a mudança para um método iterativo. Em [3], foi utilizada uma adaptação do método dos gradientes conjugados para sistemas indefinidos, o qual foi preconditionado com o preconditionador separador E . A necessidade de aproveitar bem as características de preconditionadores como o separador deu origem à uma nova abordagem para o preconditionamento de sistemas lineares oriundos de métodos de pontos interiores. Trata da abordagem de preconditionamento híbrido, a qual será apresentada na próxima seção.

3.3.3 Preconditionadores Híbridos

A estratégia de preconditionamento híbrido consiste na combinação de dois ou mais preconditionadores distintos com intuito de melhorar a eficiência de um método iterativo na resolução de sistemas lineares. No contexto de sistemas lineares oriundos de métodos de pontos interiores, essa abordagem foi proposta inicialmente por Bocanegra, Campos e Oliveira [12] e assume a existência de duas fases durante as iterações do método de pontos interiores. Na primeira fase, o preconditionador M é construído pela Fatoração Controlada de Cholesky e na segunda fase, é construído através do preconditionador separador. Abordagens desse tipo também foi utilizada em [21] e [17] e baseiam-se no fato de que, de um modo geral, os preconditionadores oriundos de fatorações incompletas de Cholesky funcionam bem nas iterações iniciais do método de pontos interiores, quando a matriz $A\Theta A^T$ ainda é bem condicionada. Por outro lado, o preconditionador separador funciona melhor nas iterações finais do método de pontos interiores. Assim, espera-se que uma combinação inteligente dos dois reforce a convergência do método. Resultados computacionais obtidos em [12] indicaram que esta combinação de preconditionadores trabalham melhor do que a abordagem direta, fatoração de Cholesky, na resolução de várias classes de problemas de programação linear, particularmente para os casos onde o fator de Cholesky tem um número grande de preenchimento.

Apesar do bom desempenho apresentado pela abordagem, a mesma pode falhar para algumas classes de problemas de programação linear. Isso pode ocorrer, principalmente, quando o preconditionador fatoração controlada de Cholesky perde eficiência ainda nas iterações iniciais do método de pontos interiores e a mudança de fase acontece sem que o preconditionador separador esteja apto para um bom desempenho.

O problema de minimizar a lacuna existente durante a troca de fase foi tratado em [21], onde o *algoritmo de ajustamento ótimo para p coordenadas* [40] foi utilizado para acelerar a convergência do método de pontos interiores. Antes da mudança de fase, algumas iterações do algoritmo são realizadas afim de melhorar o ponto corrente e entregar ao preconditionador separador um ponto mais perto da solução. Nessa estratégia, a simplicidade e o avanço inicial rápido do algoritmo de ajustamento ótimo para p coordenadas é aproveitada do seguinte modo: inicialmente, o método de pontos interiores funciona com o método dos gradientes conjugados preconditionado com a fatoração controlada de Cholesky. Quando a troca de preconditionadores é necessária, pois o critério

para a troca foi satisfeito, algumas iterações do algoritmo são realizadas para fornecer um ponto mais próximo da solução. Daí em diante, o método de pontos interiores passa a trabalhar com o preconditionador separador, acelerando assim a resolução do problema. Resultados computacionais mostraram que o algoritmo de ajustamento ótimo para p coordenadas é eficiente nas iterações próximas ao *gap* entre os dois preconditionadores utilizados na abordagem [21].

As Figuras 3.1 e 3.2 ilustram a ideia da abordagem de preconditionamento híbrido.

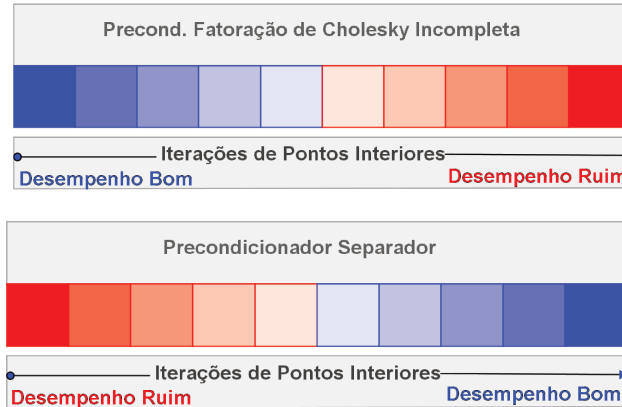


Figura 3.1: O preconditionador fatoração controlada de Cholesky trabalha bem nas iterações iniciais do método de pontos interiores enquanto a matriz do sistema linear ainda é bem condicionada. O preconditionador separador trabalha melhor nas últimas iterações do método, quando se está próximo de uma solução do problema.

A ideia é utilizar cada preconditionador em um momento adequado do preconditionamento, de modo a tirar o máximo de proveito de suas respectivas qualidades.

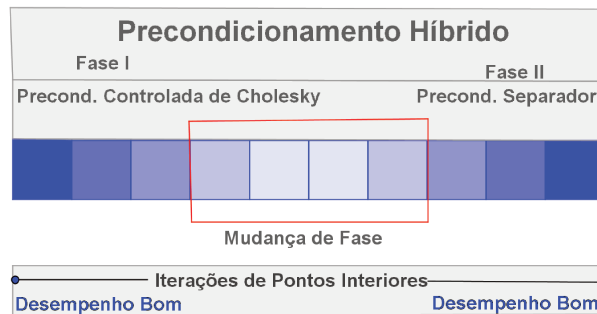


Figura 3.2: A fatoração controlada de Cholesky é usada nas primeiras iterações do método de pontos interiores. Quando esta perde a eficiência no preconditionamento, muda-se para o preconditionador separador. Uma estratégia para efetuar a mudança fase é necessária.

3.3.3.1 Mudança de fase

Quando se trata de abordagens de preconditionamento híbrido, a mudança de fase é um ponto crucial para o bom funcionamento da técnica. Atualmente, em abordagens que usam a fatoração

controlada de Cholesky, o momento exato utilizado pelo algoritmo para mudar de um preconditionador para outro é determinado por meio de heurísticas. Pelo menos duas heurísticas foram propostas para resolver o problema. A primeira delas foi proposta em [11], e funciona da seguinte maneira: a mudança de preconditionador ocorre quando o *gap* inicial ($x_0^T z_0$) para o problema de programação linear é reduzido de um fator de 10^6 ou quando o número de iterações internas do método dos gradientes conjugados atinge o valor de $m/2$, onde m é a dimensão de $A\Theta A^T$. Além disso, a heurística também controla o parâmetro η da fatoração controlada de Cholesky determinando quando preenchimentos podem ocorrer e a quantidade de novos elementos não nulos que devem ser inseridos em cada coluna do fator que será construído pela fatoração controlada de Cholesky.

Na iteração inicial, o número de entradas não nulas permitidas no preconditionador FCC é fixado pelo parâmetro η da seguinte maneira:

$$\eta_0 = \begin{cases} -\frac{|A\Theta A^T|}{m}, & \text{se } \frac{|A\Theta A^T|}{m} > 10, \\ \frac{|A\Theta A^T|}{m}, & \text{caso contrário.} \end{cases}$$

onde $|\cdot|$ denota o número de elementos não nulos na matriz. Se a matriz $A\Theta A^T$ é densa, o parâmetro η_0 induz a construção do preconditionador escala diagonal. Por outro lado, se $A\Theta A^T$ é esparsa então um preconditionador mais denso é permitido. Essa alteração é feita com base no número de iterações do método dos gradientes conjugados. À medida que o método vai perdendo eficiência, o valor de η é aumentado. Se o número de iterações realizadas supera $m/4$, o valor de η é acrescido de 10. O processo continua até atingir um valor η_{max} previamente estabelecido ou a mudança de fase ser identificada. O valor de η_{max} é baseado na quantidade de memória disponível.

Outra heurística mais eficiente e mais simples foi proposta por Velazco, Oliveira e Campos [43]. Essa se baseia apenas no número de iterações dos gradientes conjugados. Quando o número de iterações necessárias para método dos gradientes conjugados convergir é maior ou igual a $m/6$, o parâmetro η é aumentado de 10. Isto é, supondo que tal fato ocorre na iteração k , então $\eta_k = \eta_{k-1} + 10$. A mudança de fase ocorre quando η excede um valor fixado anteriormente.

Neste capítulo, foram apresentadas diversas técnicas de preconditionamento e preconditionadores utilizados no contexto de método de pontos interiores como a fatoração controlada de Cholesky e o preconditionador separador. Foi visto que a fatoração de Cholesky incompleta pode falhar na presença de pivôs pequenos na diagonal, o que pode impedir a construção do preconditionador, ou pelo menos, prejudicar o seu desempenho. No próximo capítulo, serão apresentadas propostas para tratar esse problema e melhorar o desempenho do preconditionamento de sistemas lineares oriundos de métodos de pontos interiores.

Capítulo 4

Fatoração Controlada de Cholesky Modificada

4.1 Introdução

Se $\mathcal{A}$ é uma matriz quadrada de ordem n , simétrica e positiva definida, então pode-se obter a fatoração $\mathcal{A} = LL^T$, onde L é uma matriz triangular inferior com todos os elementos da diagonal positivos. Isto é, a fatoração de Cholesky de $\mathcal{A}$ existe. No entanto, quando se trata de fatorações incompletas, pode não existir um fator $\tilde{L}$ tal que $\mathcal{A} = \tilde{L}\tilde{L}^T + R$, pois essas fatorações são suscetíveis à falhas na diagonal. Essas falhas ocorrem sempre que pivôs não positivos são encontrados durante a fatoração. Além disso, a presença de pivôs muito pequenos podem provocar instabilidade numérica na resolução do sistema triangular, comprometendo a qualidade da solução de tais sistemas [9]. A presença de pivôs nulos ou negativos na fatoração incompleta pode resultar em um preconditionador que não seja bem definido matematicamente ou numa matriz indefinida. Porém, para que o método dos gradientes conjugados funcione é necessário que tanto a matriz $\mathcal{A}$ como a matriz preconditionadora $M = \tilde{L}\tilde{L}^T$ sejam simétricas definidas positivas. Meijerink et al. [33] mostram que para uma classe de matrizes não singulares, chamada de M -matrizes, a fatoração incompleta de Cholesky sempre existe, independentemente do seu padrão de esparsidade. Esse resultado foi posteriormente estendido para uma outra classe de matrizes conhecidas como H -matrizes². Um exemplo dessas matrizes são as matrizes diagonalmente dominante [31]. Por outro lado, muitas aplicações importantes, como é o caso de métodos de pontos interiores, necessitam da resolução de sistemas lineares cujas matrizes não pertencem a nenhuma dessas duas classes.

Desde que fatorações incompletas começaram a ser usadas no preconditionamento de sistemas lineares, que falhas provocadas pela presença de pivôs não positivos ou muito pequenos são reconhecidamente um problema sério [9] e diversas alternativas têm sido propostas para contorná-lo. Neste capítulo, serão abordadas algumas técnicas comumente adotadas para corrigir falhas na diagonal durante fatorações incompletas e, em particular, da fatoração controlada de Cholesky.

De um modo geral, o uso das técnicas atuais para correção de falhas na diagonal tem a desvan-

² Uma matriz $\mathcal{A} = [a_{ij}]$ é uma H -matriz, se a matriz $H = [h_{ij}]$ tal que $h_{ij} = |a_{ij}|$ se $i = j$ e $h_{ij} = -|a_{ij}|$ se $i \neq j$, é uma M -matriz.

tagem de aumentar consideravelmente o tempo de preconditionamento e, por conseguinte, o tempo de resolução dos problemas. Com o objetivo de melhorar o desempenho da fatoração controlada de Cholesky no preconditionamento de sistemas lineares oriundos de métodos de pontos interiores, duas propostas de modificações na mesma serão apresentadas. Ambas buscam reduzir o tempo de preconditionamento dos sistemas lineares, reduzindo o número de reinícios necessários da fatoração para construir o preconditionador ou impedindo-os de acontecerem. Ambas abordagens modificam o modo como o preconditionador faz a escolha do parâmetro de correção α durante a fatoração controlada de Cholesky. No entanto, enquanto a primeira permite novas fatorações na mesma iteração do método de pontos interiores, a segunda efetua uma atualização nas colunas disponíveis do preconditionador para evitar novas fatorações.

4.2 Correção na diagonal

Como mencionado na Seção 4.1, é possível a ocorrência de falhas na diagonal durante o cálculo de fatorações incompletas como a fatoração controlada de Cholesky, por exemplo. Essas falhas podem ocorrer ainda que a matriz $\mathcal{A}$ seja simétrica definida positiva, e em geral são provocadas pela ocorrência de pivôs não positivos na diagonal. As diversas estratégias propostas para contornar esse problema se encontram classificadas, por ordem crescente de complexidade, nos seguintes grupos [9]:

Dominância Diagonal. A correção na diagonal é realizada por meio de um aumento no valor dos elementos da diagonal da matriz $\mathcal{A}$, acrescentando-se a estes um incremento positivo. Essa estratégia pode ser local ou global. Isto significa que um valor positivo pode ser acrescentado apenas na posição da diagonal da matriz $\mathcal{A}$, onde ocorreu a falha, ou pode ser acrescentado a todos os elementos da diagonal. Em qualquer um dos casos a fatoração da matriz $\mathcal{A} + \Delta$ é reiniciada após a correção, onde Δ é a matriz diagonal cujos elementos são aqueles que devem ser somado a diagonal da matriz $\mathcal{A}$, em suas respectivas posições. Comumente, $\Delta = \alpha \cdot \text{diag}(\mathcal{A})$ ou $\Delta = \alpha I_n$.

Substituição Matricial. A matriz $\mathcal{A}$ é substituída por uma outra matriz $\mathcal{B}$ próxima de $\mathcal{A}$ e para a qual existe a fatoração $\mathcal{B} = \tilde{L}\tilde{L}^T$. Uma maneira seria encontrar uma M -matriz ou uma H -matriz X tal que $\|\mathcal{A} - X\| \simeq 0$, pois se sabe que para tais matrizes existe uma fatoração incompleta [29].

Mudança de Fatoração. A matriz $\mathcal{A}$ não é modificada, mas encontra-se uma outra forma de fatoração que evite pivôs nulos ou muito pequenos.

Por serem menos complexas, as técnicas do primeiro grupo são mais comuns na literatura e um exemplo bem simples foi proposto por Kershaw [27]: um pivô pequeno ou negativo é trocado em cada passo i da fatoração, onde o mesmo aparecer, por algum valor $p_i > 0$. Esta é uma estratégia local motivada pela esperança de que se apenas poucos pivôs são instáveis a fatoração resultante ainda pode render um preconditionador satisfatório. Apesar da simplicidade da técnica em si mesma, a escolha de tal p_i não é um trabalho fácil de realizar. De fato, a escolha de um p_i muito grande pode implicar em $\|\mathcal{A} - \tilde{L}\tilde{L}^T\|_F$ muito grande e resultar na imprecisão do preconditionador. Por outro lado, a escolha de um p_i muito pequeno, pode implicar que $\|I - \tilde{L}\tilde{A}L^T\|_F$ seja um número grande e isso poderia gerar instabilidade na resolução dos sistemas triangulares com $\tilde{L}$ e $\tilde{L}^T$. Um exemplo mais elaborado dessa técnica foi proposto por Manteufel [31]: quando uma

falha ocorre na fatoração devido a presença de pivôs pequenos ou não positivos, um incremento global é aplicado na diagonal de $\mathcal{A}$ antes de tentar uma nova fatoração. Assim, a nova fatoração incompleta é calculada com a matriz

$$\tilde{\mathcal{A}} = \mathcal{A} + \alpha \text{diag}(\mathcal{A}), \alpha > 0.$$

Se para um determinado valor de α a fatoração falhar, então este valor é aumentado e, em seguida, somado novamente à diagonal da matriz e a fatoração da mesma é reiniciada. O processo é repetido até que a fatoração incompleta seja calculada com sucesso. Como no caso anterior, a escolha de um α adequado não é simples. Contudo é possível afirmar que existe um $\alpha^* > 0$ para o qual a fatoração incompleta da matriz

$$\bar{\mathcal{A}} = \mathcal{A} + \alpha^* \text{diag}(\mathcal{A})$$

existe. Por exemplo, α^* pode ser o menor valor de α para o qual a matriz $\bar{\mathcal{A}}$ seja diagonalmente dominante. De fato, matrizes diagonalmente dominantes são H -matrizes. Porém, dominância diagonal é uma condição suficiente, mas não necessária para a existência de fatorações incompletas. Desse modo é possível determinar um α tal que $0 < \alpha < \alpha^*$ e para o qual a fatoração incompleta da matriz $\bar{\mathcal{A}}$ existe [31]. Em geral, α é calculado por meio de tentativa e erro, prática que encarece a construção do preconditionador e configura a escolha e manipulação deste parâmetro como tarefas desafiadoras quando se trata de calcular fatorações incompletas. A seguir, serão apresentadas as duas abordagens usuais para o cálculo de α .

4.2.1 Incremento linear

Utilizada inicialmente por Jones e Plassmann [25], essa abordagem soma um valor fixo $\lambda > 0$ à diagonal da matriz, reinicia a fatoração e repete o processo até que a fatoração incompleta seja realizada com sucesso. Dessa forma, o parâmetro de correção α é calculado pela fórmula

$$\alpha_i = \lambda i,$$

onde λ é um número fixo e $i = 1, \dots, T_{max}$. O número T_{max} indica o número máximo permitido de reinício da fatoração. Jones e Plassmann utilizaram $\lambda = 10^{-2}$. Além de usar o primeiro valor de α aceitável, Jones e Plassmann [25] baseados em [31], calcularam um outro valor de α , num certo sentido um valor ótimo de α , para o qual o número de iterações do método dos gradientes conjugados era menor. Em geral, o valor do α “ótimo” não ultrapassou 5% do valor do primeiro α utilizado, mas em compensação, o número de iterações do gradientes conjugados, em alguns casos, foi reduzido de até 25%.

4.2.2 Incremento exponencial

Quando um elemento da diagonal é muito pequeno ou não positivo, essa abordagem soma aos elementos da diagonal de $\mathcal{A}$, o valor definido por

$$\alpha_i = \lambda 2^i$$

e recomeça a fatoração. Neste caso, λ é uma constante positiva e $i = 1, \dots, T_{max}$. Esta técnica tem a vantagem de encontrar mais rapidamente um valor adequado para α e de reiniciar a fatoração um número menor de vezes. A implementação atual da Fatoração Controlada de Cholesky usa $\lambda = 5,0 \times 10^{-4}$. A sequência de valores de α utilizada pela mesma pode ser visualizada na Figura 4.1.

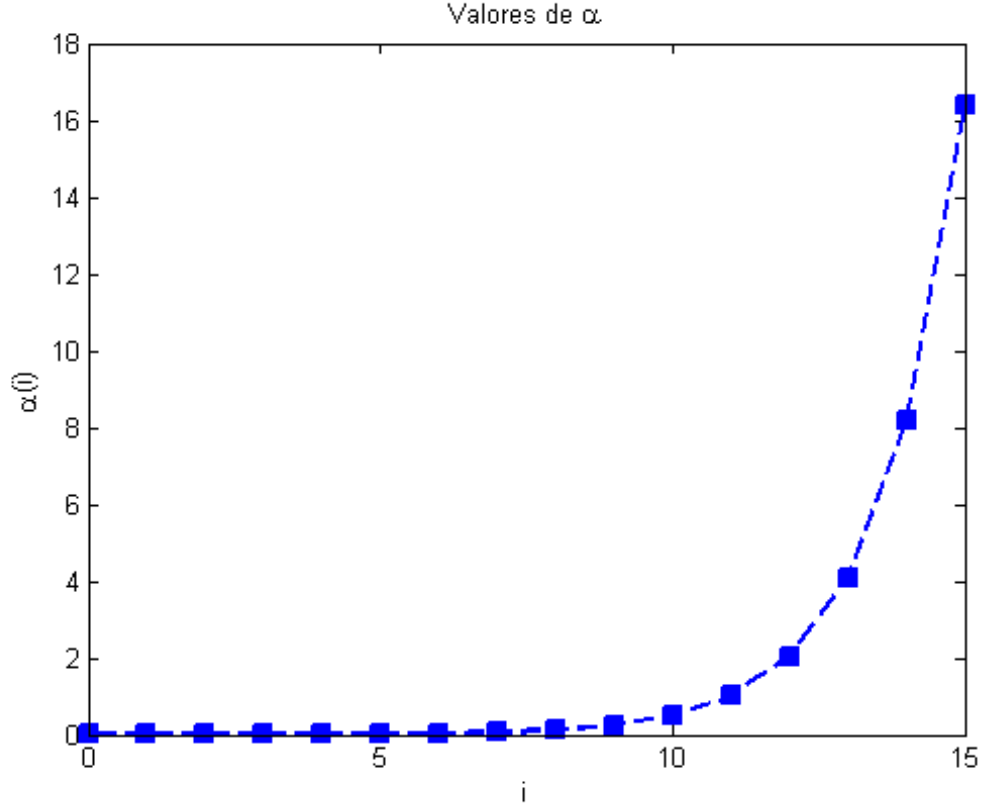


Figura 4.1: Sequência de valores de α utilizada pela fatoração controlada de Cholesky para efetuar correções na diagonal.

4.2.3 Correção na diagonal na fatoração controlada de Cholesky

Como visto no Capítulo 3, a fatoração Controlada de Cholesky é um preconditionador construído com base na fatoração de Cholesky incompleta e, por isso, não está livre de falhas causadas pela presença de pivôs inadequados na diagonal. Na abordagem híbrida para o preconditionamento em pontos interiores proposta em [12], as falhas que ocorrem durante a fatoração são corrigidas com o incremento exponencial. Para alguns problemas, o processo é repetido muitas vezes até que um valor de α adequado seja encontrado. Isso acarreta um aumento no tempo de preconditionamento e o desempenho do preconditionador fica prejudicado [12]. Um exemplo desse fenômeno é apresentado na Tabela 4.1. Ao lado de cada problema, nas linhas horizontais, está indicado o número de vezes que a fatoração de Cholesky é reiniciada e a respectiva iteração do método de

Tabela 4.1: Correção na diagonal e reinício da FCC por iteração de pontos interiores

	Número de vezes que a fatoração é reiniciada																	
Iteração:	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
els19										14	14	13	13	12	12	*		
ch25a												13	13	11	11	11	*	
scr15								13	13	12	12	12	*					
rou20								15	14	13	13	13	13	*				
qap12		15	12	12	12	*												
nug12			14	5	6	7	7	*										

pontos interiores onde tal fato ocorre. O asterisco simboliza a iteração onde ocorreu a mudança de preconditionadores. Nos problemas **rou20** e **qap12** a fatoração incompleta chega a ser reiniciada quinze vezes na mesma iteração do método de pontos interiores. Além disso, para o problema **rou20** a fatoração precisa ser recomeçada nas cinco iterações seguintes do método, totalizando setenta e um reinícios da fatoração controlada de Cholesky. Evidentemente, esse fato pode elevar consideravelmente o tempo computacional necessário para a resolução do problema. Assim, é interessante pensar em técnicas mais eficientes para o cálculo do parâmetro α a serem utilizados na correção, bem como na construção de preconditionadores aproximados para a matriz do sistema, de modo a acelerar a resolução dos problemas de programação linear. Propostas para lidar com esses problemas serão apresentadas nas próximas seções.

A principal desvantagem do uso das abordagens descritas anteriormente é o fato de ser necessária a fatoração da matriz $\mathcal{A} + \alpha \text{diag}(\mathcal{A})$, para cada valor de α , até um valor aceitável ser encontrado. Isto acontece porque o procedimento adotado quando ocorre uma falha durante a fatoração da matriz $\mathcal{A}$, é efetuar a correção, somando à diagonal de $\mathcal{A}$ um valor pequeno α e iniciar a fatoração da matriz $\mathcal{A} + \alpha \text{diag}(\mathcal{A})$. Se a fatoração incompleta falhar novamente, um novo valor de α é calculado, somado à diagonal de $\mathcal{A}$ e o processo se repete até que a fatoração seja concluída com sucesso ou até que o número máximo de reinícios permitidos T_{max} seja atingido. Essa prática tem reportado bons resultados no preconditionamento híbrido em método de pontos interiores, mas encarece a resolução dos problemas de programação linear devido ao alto custo computacional, sobretudo quando há a necessidade de recomeçar a fatoração várias vezes na mesma iteração do método. Os resultados obtidos em [25] sugerem que efetuar a correção começando com valores de α maiores também pode render bons preconditionadores ou, pelo menos, diminuir o tempo gasto com o cálculo de várias fatorações na mesma iteração do método de pontos interiores para construir o preconditionador. Por outro lado, sabe-se que apesar de um valor grande de α garantir a existência da fatoração, o mesmo pode implicar em preconditionadores de baixa qualidade [25, 31].

4.3 Uma nova abordagem para o cálculo de α na fatoração controlada de Cholesky

A abordagem proposta procura combinar dois aspectos importantes na correção de falhas em fatorações incompletas: efetuar correções com valores pequenos de α e reduzir o número de reinícios

necessários para que a fatoração seja realizada com sucesso. A primeira exigência pode ser atendida iniciando-se a correção com valores pequenos de α e aumentá-lo progressivamente, se necessário, como já é feito na fatoração controlada de Cholesky. No entanto, para evitar que novas fatorações sejam realizadas a cada valor de α , pode-se permitir uma correção com valores de α maior do que seria necessário a princípio. Assim, em vez de testar cada valor α , a partir do menor permitido, até encontrar um valor que complete a fatoração, propõe-se usar como teste um valor particular de α .

4.3.1 Estratégia para o cálculo de α

Suponha que durante o processo da fatoração $\mathcal{A} \simeq \tilde{L}\tilde{D}\tilde{L}^T$, no cálculo da coluna j das matrizes $\tilde{L}$ e $\tilde{D}$, tenha-se

$$d_{jj} \leq tol,$$

para alguma tolerância tol dada previamente.

É proposto o seguinte procedimento para a escolha de um α adequado para efetuar a correção necessária: Dada uma constante $\lambda > 0$ e $i = 1, \dots, T_{max}$, calcula-se o menor α_i , usando a fórmula,

$$\alpha_i = \lambda 2^i, \quad (4.3.1)$$

tal que

$$d_{jj} + \alpha_i > tol. \quad (4.3.2)$$

T_{max} é o número máximo de testes permitidos para achar o valor de α adequado e também serve para limitar o valor α a ser utilizado nas correções.

4.3.2 Fatoração Controlada de Cholesky Modificada Alfa

Uma vez disponível um α , calculado como em 4.3.1, pode-se pensar em estratégias para efetuar a correção da falha ocorrida durante a fatoração controlada de Cholesky. A primeira modificação proposta à fatoração controlada de Cholesky, segue a mesma idéia já utilizada pela mesma e permite o reinício da fatoração matricial, no entanto utiliza a nova maneira de calcular α . Dessa forma, após obter um valor de α , por exemplo $\alpha = \bar{\alpha}$, como descrito anteriormente, reinicia-se a fatoração da matriz $\mathcal{A} + \bar{\alpha}diag(\mathcal{A})$, que no caso da matriz escalada é simplesmente $\mathcal{A} + \bar{\alpha}I$.

A motivação para a escolha desse procedimento é a esperança de que durante a fatoração da matriz $\mathcal{A} + \bar{\alpha}I$ não ocorra mais falhas na coluna j . Isso já economizaria várias tentativas de fatorações com valores menores de α . No caso do valor de α coincidir com o valor que torna a matriz $\mathcal{A} + \bar{\alpha}I$ diagonalmente dominante, o sucesso da fatoração já está garantido. No entanto, ainda que α escolhido dessa forma permita o término da fatoração incompleta, é possível que existam valores de α menores do que $\bar{\alpha}$, tais que, a fatoração de $\mathcal{A} + \alpha I$ existe e poderia resultar em melhores condicionadores do que $\bar{\alpha}$. Sendo assim, para atenuar esse problema pode-se usar uma estratégia de *backtracking* como proposto na seguinte fórmula para reduzir o valor de α :

$$\alpha = \tau \bar{\alpha},$$

onde $\tau \in (0, 1]$ e $\bar{\alpha}$ calculado como em 4.3.1.

De modo resumido, quando ocorre uma falha na coluna j durante a fatoração controlada de Cholesky modificada, calcula-se um $\bar{\alpha}$ tal que $d_{jj} + \bar{\alpha} > \text{tol}$ e, em seguida, reduz o tamanho do mesmo multiplicando-o pelo fator τ e reinicia a fatoração da matriz $\mathcal{A} + \alpha I$, onde $\alpha = \tau \bar{\alpha}$. A escolha do fator τ deve levar em consideração os dois aspectos mencionados anteriormente, ou seja, permitir uma redução razoável do valor $\bar{\alpha}$ para que se tenha um bom preconditionador, mas essa redução não pode ser muito grande a ponto de que novas correções precisem ser efetuadas na mesma iteração. Na prática, o cálculo de um determinado valor de α usando a fórmula $\alpha = \tau \lambda 2^i$ é equivalente a modificar o valor da constante $\lambda = 5,0 \times 10^{-4}$ utilizada na fatoração controlada de Cholesky original e calcular α diretamente a partir da fórmula

$$\alpha = \lambda 2^i,$$

onde λ pode variar de acordo com o tipo de problema que se pretende resolver. A Figura 4.2 ilustra essa proposta de modificação na fatoração controlada de Cholesky.

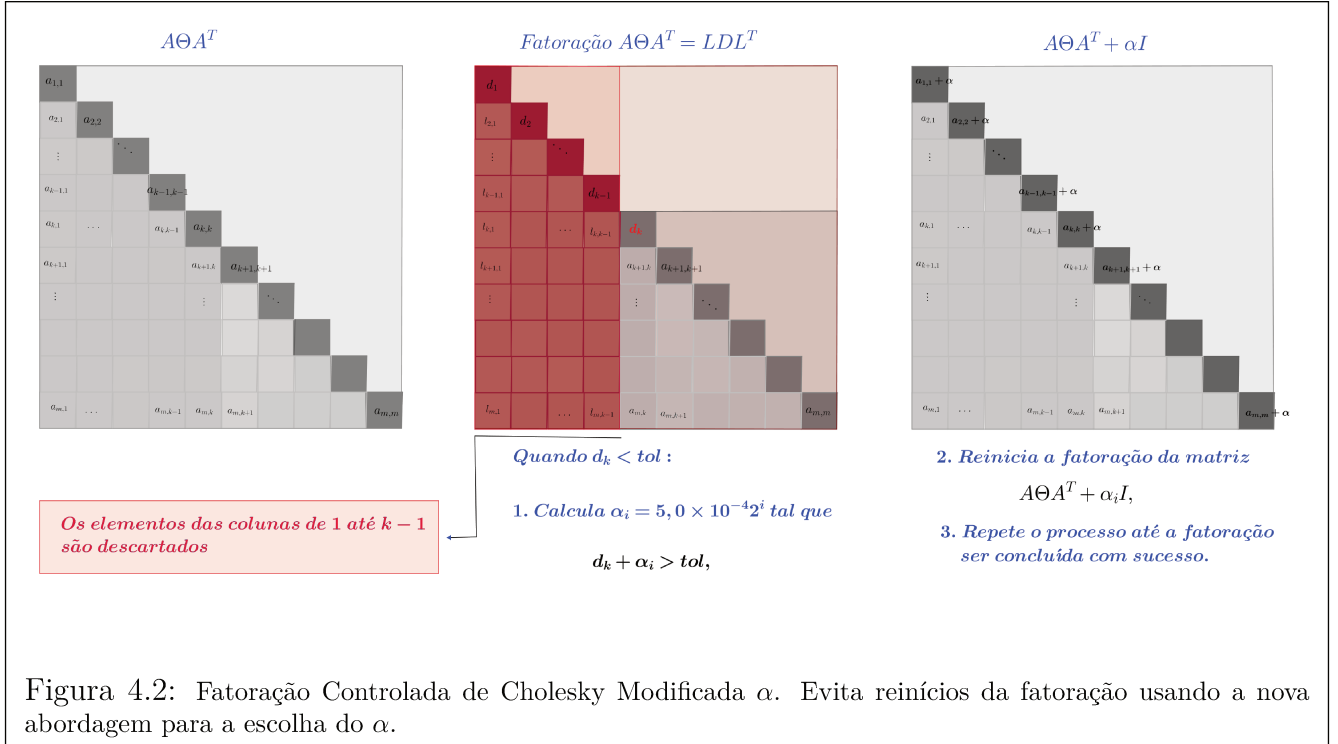


Figura 4.2: Fatoração Controlada de Cholesky Modificada α . Evita reinícios da fatoração usando a nova abordagem para a escolha do α .

Essencialmente a nova abordagem modifica a fatoração controlada de Cholesky apenas no modo como o preconditionador calcula e testa os valores de α utilizados para construir os fatores $\tilde{L}$ e $\tilde{D}$, mediante a ocorrência de falhas na diagonal. Como a abordagem utiliza a nova maneira de calcular o parâmetro α , a correção na diagonal pode ser iniciada com valores menores de α , pois o teste realizado antes do reinício da fatoração é apenas uma operação aritmética, ao contrário da fatoração controlada de Cholesky, onde os testes são fatorações matriciais, que são realizadas desde o início da tentativa de correção. E nesse caso, a tentativa de começar a correção com

valores menores, poderia implicar em um número maior de reinícios, e portanto, um gasto maior do tempo computacional. O uso de valores menores de α nas correções é importante porque mantém a qualidade do preconditionador.

Outro tipo de estratégia proposta para efetuar a correção de falhas na diagonal procura evitar que a fatoração da matriz $\mathcal{A} + \alpha I$ seja reiniciada. A estratégia lança mão do uso da atualização de preconditionadores, como proposto em [7], e serão tratados com mais detalhes na Seção 4.4.

A motivação para essa nova estratégia é proveniente do fato de que as técnicas de correção na diagonal da matriz $\mathcal{A}$ descritas anteriormente, dão origem a novas matrizes do tipo $\mathcal{A} + \alpha I$, que devem ser fatoradas a fim de obter os preconditionadores adequados. Uma alternativa à fatoração dessa matriz é o cálculo de um preconditionador aproximado para a matriz $\mathcal{A} + \alpha I$ baseado em atualizações do preconditionador já disponível para a matriz $\mathcal{A}$. Este problema aparece em outros contextos de otimização e também de resolução numérica de equações diferenciais parciais, onde sequências de matrizes do tipo $\mathcal{A} + \alpha I$ são frequentes. Meurant [34] foi o pioneiro a tratar do preconditionamento de sistemas desse tipo, ao considerar meios baratos para calcular uma fatoração de Cholesky incompleta para matrizes simétricas perturbadas $\mathcal{A} + \alpha I$, com valores de α pequenos, conhecendo-se uma fatoração incompleta da matriz $\mathcal{A}$. Além da técnica utilizada por Meurant, a Seção 4.4 apresenta uma exposição sucinta sobre o tema e está baseada, principalmente, em [7].

4.4 Preconditionadores para Sistemas Lineares diagonalmente modificados

Considere o sistema linear $\mathcal{A}x = b$ e suponha que o mesmo foi modificado somando a matriz diagonal Δ_k à matriz dos coeficientes, resultando na seguinte sequência de sistemas lineares modificados

$$(\mathcal{A} + \Delta_k)x_k = b_k, \quad k = 1, 2, 3, \dots \quad (4.4.1)$$

Cada um desses sistemas é chamado de *sistema linear diagonalmente modificado*. O problema de resolver uma sequência de tais sistemas aparece em vários campos da otimização, como métodos de região de confiança e técnicas de regularização para mínimos quadrados não lineares e, também na aplicação de métodos implícitos para equações diferenciais parciais [6]. Dessa forma, é pertinente considerar o problema de construir uma sequência de preconditionadores $\{M_k\}$ eficientes para encontrar a solução de tais sistemas lineares, através de métodos iterativos. A construção de preconditionadores eficientes e computacionalmente baratos para o método dos gradientes conjugados, para sistemas lineares como em 4.4.1, está bem apresentado em [7]. Neste caso, a matriz $\mathcal{A} \in \mathbb{R}^{n \times n}$ é simétrica positiva definida e esparsa e, $\Delta_k \in \mathbb{R}^{n \times n}$ é uma matriz diagonal positiva semidefinida.

4.4.1 Uma classe de preconditionadores atualizados para $\{\mathcal{A} + \Delta_k\}$

A classe $\text{UFP}(\mathcal{A}, \Delta_k)$ - *Updated Factorization Preconditioners* para $\{\mathcal{A} + \Delta_k\}$ - é uma classe de preconditionadores que satisfazem algumas propriedades. Na verdade, trata-se de um *framework*

para a construção e análise de um preconditionador M_k para cada matriz $\mathcal{A} + \Delta_k$, onde $\Delta_k = \text{diag}(\delta_1^k, \dots, \delta_n^k)$ e $\delta_i^k \geq 0$ para $i = 1, \dots, n$, proposto em [7]. Cada preconditionador M_k é obtido pela atualização de um preconditionador M , chamado de preconditionador *semente*, o qual é simétrico definido positivo e está disponível para $\mathcal{A}$ na forma fatorada LDL^T , onde L é triangular inferior com diagonal unitária e $D = \text{diag}(d_1, \dots, d_n)$ com $d_i > 0$ para $i = 1, \dots, n$.

Definição 4.4.1. Uma sequência de preconditionadores $\{M_k\}$ para $\{\mathcal{A} + \Delta_k\}$ pertence à classe UFP($\mathcal{A}, \Delta_k$), se

$$M_k = L_k D_k L_k^T,$$

com L_k triangular inferior e $D_k = \text{diag}(d_1^k, \dots, d_n^k)$, tal que,

- (A1) $d_i^k \geq d_i + \delta_i^k$, $i = 1, \dots, n$;
- (A2) Existe $\rho > 0$ independente de k tal que $\|D_k - D\| \leq \rho \|\Delta_k\|$;
- (A3) $\text{diag}(L_k) = \text{diag}(L) = (1, \dots, 1)^T$ e $\text{off}(L_k) = \text{off}(L)S_k$, onde $S_k = DD_k^{-1}$.

A Definição 4.4.1 é motivada pelo desejo de obter preconditionadores que podem ser aplicados ao mesmo custo que o preconditionador semente, e de alguma forma imitar o comportamento da matriz $\mathcal{A} + \Delta_k$. Por definição, M_k é simétrica definida positiva e $S_k = \text{diag}(s_1^k, \dots, s_n^k)$ é tal que $s_i^k \in (0, 1]$, $i = 1, \dots, n$, isto é, $\|S_k\| \leq 1$. Com isso, é possível provar que o condicionamento de L_k é pelo menos tão bom quanto o condicionamento de L e que o custo computacional de construir M_k é pelo menos igual ao custo de realizar um escalamento em L para construir L_k , o qual é da ordem de $O(nnz(L))$. Assim, a escolha de D_k desempenha um papel crucial no custo computacional global do preconditionador e é importante para achar um bom equilíbrio entre o custo computacional e a eficiência do preconditionador.

Para o problema de determinar preconditionadores para as matrizes $\mathcal{A} + \Delta_k$ temos as seguintes possibilidades:

1. **Reutilizar o preconditionador semente M no preconditionamento de cada uma das matrizes subsequentes.** Nesta estratégia, como não é necessário calcular novos preconditionadores, economiza-se o tempo computacional referente ao cálculo dos mesmos. No entanto, o preconditionador semente, em geral, tem menor eficiência quando utilizado no preconditionamento dos sistemas diagonalmente modificados.
2. **Recalcular o preconditionador para cada matriz $\mathcal{A} + \Delta_k$.** Esta estratégia produz preconditionadores mais eficientes para a sequência de sistemas lineares diagonalmente modificados, contudo, pode aumentar consideravelmente o tempo computacional, já que é calculado um preconditionador para cada sistema linear.
3. **Realizar atualizações de uma fatoração incompleta LDL^T de $\mathcal{A}$ para cada um dos sistemas $\mathcal{A} + \Delta_k$.** Nesta estratégia, o custo computacional, em termos de tempo computacional do cálculo dos preconditionadores, é menor do que o da segunda estratégia e os mesmos podem ser mais eficientes do que o preconditionador semente.

Sobre o desempenho desses preconditionadores, Bellavia et al.[7] observam que, na prática, o desempenho dos preconditionadores atualizados está entre o desempenho do preconditionador somente e o desempenho dos preconditionadores recalculados, em termos de iterações do método dos gradientes conjugados, mas frequentemente são mais rápidos em termos de tempo computacional.

Sobre a qualidade dos preconditionadores da classe $UFP(\mathcal{A}, \Delta_k)$, dois resultados importantes estão estabelecidos nos teoremas apresentados a seguir e cujas provas se encontram em [7]. O primeiro afirma que para valores pequenos de $\|\Delta_k\|$, M_k é uma boa aproximação de $\mathcal{A} + \Delta_k$, e o segundo estabelece que os autovalores de $M^{-1}(\mathcal{A} + \Delta_k)$ estão agrupados em *clusters* em torno da unidade.

Teorema 4.4.2. Seja $\{M_k\} \in UFP(\mathcal{A}, \Delta_k)$. Então existe um $\xi > 0$ independente de k tal que $\|\mathcal{A} + \Delta_k - M_k\| \leq \xi \|\Delta_k\|$.

Demonstração.: [7].

Teorema 4.4.3. Seja $\{M_k\} \in UFP(\mathcal{A}, \Delta_k)$. Para todo $\epsilon > 0$ existe um $\phi > 0$ independente de k tal que, se $\|\Delta_k\| < \phi$, então

$$|\lambda_i(M_k^{-1}(\mathcal{A} + \Delta_k)) - 1| < \epsilon, \quad i = 1, \dots, n.$$

Além do mais, se $\mathcal{A} + \Delta_k - M_k$ tem posto $n - r$, então r autovalores de $M^{-1}(\mathcal{A} + \Delta_k)$ são iguais a 1.

Demonstração.: [7].

Esses resultados permitem concluir que uma sequência $\{M_k\} \in UFP(\mathcal{A}, \Delta_k)$ pode, de fato, ser eficiente para sequências de sistemas lineares suavemente modificados com matrizes dos coeficientes $\mathcal{A} + \Delta_k$. Preconditionadores com melhores propriedades teóricas podem ser obtidos se a seguinte propriedade for acrescentada aos preconditionadores $M_k \in UFP(\mathcal{A}, \Delta_k)$:

(A4) Existe $\kappa > 0$ independente de k tal que $\|\mathcal{A} + \Delta_k - M_k\| \leq \kappa$.

Com esta propriedade adicional pode-se mostrar que a eficiência de $\{M_k\} \in UFP(\mathcal{A}, \Delta_k)$ aumenta à medida que $\|\Delta_k\|$ cresce. Tal fato é estabelecido no seguinte teorema:

Teorema 4.4.4. Seja $\{M_k\} \in UFP(\mathcal{A}, \Delta_k)$ satisfazendo (A4). Para todo $\epsilon > 0$ existe um $\phi > 0$ independente de k tal que, se $\|\Delta_k\| > \phi$, então

$$\frac{\|\mathcal{A} + \Delta_k - M_k\|}{\|\mathcal{A} + \Delta_k\|} < \epsilon.$$

Demonstração.: [7].

Os resultados anteriores são extensíveis à fatorações incompletas e também para matrizes positivas semidefinidas. No caso das fatorações incompletas, seja $\mathcal{A}$ uma matriz simétrica definida positiva e suponha que o preconditionador L é uma fatoração incompleta de $\mathcal{A}$, isto é, existe uma matriz R , tal que

$$\mathcal{A} = LDL^T + R.$$

Então, vale a desigualdade

$$||\mathcal{A} + \Delta_k - M_k|| \leq \xi ||\Delta_k|| + ||R||.$$

Assim, quando $||\Delta_k||$ é pequeno, a qualidade do preconditionador M_k depende do tamanho de $||R||$. Se $||R||$ é grande a capacidade de agrupamento dos autovalores em torno de 1 é perdida. Por outro lado, quando $||\Delta_k||$ é pequeno não se pode esperar que o preconditionador seja eficiente para $\mathcal{A} + \Delta_k$ se não o é para $\mathcal{A}$. A situação é diferente quando $||\Delta_k||$ é grande. De fato, o Teorema 4.4.4 ainda vale independentemente do tamanho de $||R||$. Isto está de acordo com o fato de que, à medida que todas as entradas da diagonal crescem, a matriz $\mathcal{A} + \Delta_k$ torna-se cada vez mais diagonalmente dominante e M_k tende ao preconditionador diagonal cujas entradas da diagonal crescem tão rápido quanto as entradas da diagonal de Δ_k .

4.4.1.1 Exemplo de preconditionadores $M_k \in UFP(\mathcal{A}, \Delta_k)$

A seguinte estratégia de atualização da fatoração LDL^T para matrizes diagonalmente modificadas, com $\Delta_k = \alpha_k I$, foi proposta em[6]. Considere $\mathcal{A}$ uma matriz simétrica definida positiva de ordem n para a qual está disponível uma fatoração incompleta LDL^T , com L sendo uma matriz triangular inferior com diagonal unitária e D é uma matriz diagonal com entradas positivas. A técnica consiste em construir um preconditionador M_k para a matriz $\mathcal{A} + \alpha_k I$, onde $\alpha_k > 0$, do seguinte modo:

$$M_k = (L + E_k + F_k)D(L + E_k + F_k)^T, \quad (4.4.2)$$

sendo E_k diagonal e F_k estritamente triangular inferior. A matriz $E_k = \text{diag}(e_1^k, \dots, e_n^k)$ é definida por

$$e_i^k = \sqrt{1 + \frac{\alpha_k}{d_i}} - 1, \quad 1 \leq i \leq n, \quad (4.4.3)$$

enquanto as entradas não nulas da matriz $F_k = (f_{ij}^k)$ são dadas por

$$f_{ij}^k = \lambda_j^k l_{ij}, \quad 2 \leq i \leq n, \quad 1 \leq j \leq n, \quad (4.4.4)$$

$$\lambda_j^k = \sqrt{\frac{d_j}{d_j + \alpha_k}} - 1. \quad (4.4.5)$$

Na prática, a atualização da fatoração LDL^T de $\mathcal{A}$ consiste em adicionar o termo e_i^k a i -ésima entrada unitária da diagonal de L e escalar as entradas abaixo da diagonal, na i -ésima coluna, com o escalar $\lambda_i^k + 1$. De fato, das equações (4.4.2) e (4.4.3) obtém-se

$$\tilde{L}_{ii} = L_{ii} + e_i^k = 1 + e_i^k$$

para $i = 1, \dots, n$. E das equações (4.4.2) e (4.4.4), obtém-se:

$$\tilde{L}_{ij} = L_{ij} + F_{ij} = L_{ij} + \lambda_j^k L_{ij} = (1 + \lambda_j^k) L_{ij}$$

para $2 \leq i \leq n$ e $1 \leq j \leq n$.

O padrão de esparsidade dos fatores de M_k em (4.4.2) é igual ao dos fatores da fatoração incompleta LDL^T de $\mathcal{A}$.

Na Seção 4.5, será apresentada uma proposta para construção de um preconditionador $\tilde{M}_\alpha = \tilde{L}\tilde{D}\tilde{L}^T$ para a matriz $\mathcal{A} + \alpha I$ através da atualização do preconditionador fatoração controlada de Cholesky $M = LDL^T$ disponível para a matriz $\mathcal{A}$. Nesta proposta, quando ocorre uma falha durante a construção do fator L em alguma coluna j , os elementos a_{kk} da diagonal de $\mathcal{A}$ para $k = j, \dots, n$ são incrementados de um valor positivo α , os elementos l_{ik} das colunas $k = 1, \dots, j-1$, anteriormente calculados, são atualizados adequadamente e a fatoração continua a partir da coluna $j+1$, até completar a construção de $\tilde{L}$. Mostra-se também que o preconditionador $\tilde{M}_\alpha$ assim obtido pertence à classe UFP($\mathcal{A}, \Delta_k$).

4.5 Uma proposta de Atualização para a Fatoração Controlada de Cholesky

Considere $\mathcal{A}$ uma matriz simétrica positiva definida de ordem n e o preconditionador fatoração controlada de Cholesky M , conforme apresentado no Capítulo 3, o qual é construído através de uma fatoração LDL^T incompleta da matriz $\mathcal{A}$. Do mesmo modo que em Meurant[34], a proposta de construção de um preconditionador $\tilde{M}_\alpha$ para a matriz $\mathcal{A} + \alpha I$, a partir de atualizações do preconditionador M disponível para a matriz $\mathcal{A}$, se baseia nas etapas da fatoração LDL^T da matriz $\mathcal{A} + \alpha I$. Desse modo, será útil apresentar o **Algoritmo 4.1**, o qual realiza o cálculo da fatoração $\mathcal{A} = LDL^T$ completa. De acordo com este algoritmo, os elementos d_j e l_{ij} dos fatores D

Algoritmo 4.1 Fatoração LDL^T

```

para j=1,...,n faça
   $d_j = a_{jj} - \sum_{k=1}^{j-1} l_{jk}^2 d_k$ 
  para i=j+1,...,n faça
     $l_{ij} = \frac{1}{d_j} (a_{ij} - \sum_{k=1}^{j-1} l_{ik} d_k l_{jk})$ 
  fim para
fim para

```

e L são determinados pelas seguintes fórmulas:

$$d_j = a_{jj} - \sum_{k=1}^{j-1} l_{jk}^2 d_k, \quad j = 1, \dots, n, \quad (4.5.1)$$

e

$$l_{ij} = \frac{1}{d_j} (a_{ij} - \sum_{k=1}^{j-1} l_{ik} d_k l_{jk}), \quad j = 1, \dots, n \text{ e } i = j+1, \dots, n, \quad (4.5.2)$$

respectivamente.

Como a atualização proposta está baseada na fatoração da matriz $\mathcal{A} + \alpha I$, é interessante conhecer o comportamento da mesma ao longo de uma fatoração LDL^T . Para isso, suponha que os elementos da diagonal da matriz $\mathcal{A}$ foram incrementados por um valor $\alpha > 0$, de modo a obter a matriz $\bar{\mathcal{A}}$ tal que

$$\bar{\mathcal{A}} = \mathcal{A} + \alpha I.$$

Efetuada o cálculo da decomposição $\bar{A} = \bar{L}\bar{D}\bar{L}^T$ de acordo com o **Algoritmo 4.1**, tem-se:

$$\bar{d}_j = \bar{a}_{jj} - \sum_{k=1}^{j-1} \bar{l}_{jk}^2 \bar{d}_k \quad j = 1, \dots, n \quad (4.5.3)$$

e

$$\bar{l}_{ij} = \frac{1}{\bar{d}_j} (\bar{a}_{ij} - \sum_{k=1}^{j-1} \bar{l}_{ik} \bar{d}_k \bar{l}_{jk}), \quad j = 1, \dots, n \quad \text{e} \quad i = j+1, \dots, n. \quad (4.5.4)$$

É importante observar que os elementos da matriz $\bar{\mathcal{A}}$ que estão fora da diagonal principal são exatamente os mesmos elementos da matriz $\mathcal{A}$ em suas respectivas posições, enquanto os elementos da diagonal principal são os elementos da matriz $\mathcal{A}$, em suas respectivas posições, acrescidos do número real α . Assim, as equações (4.5.3) e (4.5.4) podem ser reescritas, respectivamente, da seguinte maneira:

$$\bar{d}_j = a_{jj} + \alpha - \sum_{k=1}^{j-1} \bar{l}_{jk}^2 \bar{d}_k, \quad j = 1, \dots, n. \quad (4.5.5)$$

e

$$\bar{l}_{ij} = \frac{1}{\bar{d}_j} (a_{ij} - \sum_{k=1}^{j-1} \bar{l}_{ik} \bar{d}_k \bar{l}_{jk}), \quad j = 1, \dots, n \quad \text{e} \quad i = j+1, \dots, n. \quad (4.5.6)$$

Agora, para entender o comportamento dos fatores $\bar{L}$ e $\bar{D}$ ao longo da fatoração $\bar{L}\bar{D}\bar{L}^T$ de $\bar{A}$ e construir aproximações para esses fatores, considere o cálculo, por colunas, dos elementos das matrizes $\bar{D}$ e $\bar{L}$. Os elementos da primeira coluna das matrizes $\bar{D}$ e $\bar{L}$ podem ser obtidos substituindo $j = 1$ nas equações (4.5.5) e (4.5.6). Para o elemento $\bar{d}_1$, tem-se:

$$\bar{d}_1 = a_{11} + \alpha = d_1 + \alpha,$$

pois $d_1 = a_{11}$ na fatoração da matriz $\mathcal{A}$, e os elementos abaixo da diagonal principal da primeira coluna de $\bar{L}$ são dados por:

$$\bar{l}_{i1} = \frac{1}{\bar{d}_1} a_{i1}.$$

Para cada $i = 2, \dots, n$ os elementos $\bar{l}_{i1}$ podem ser reescritos da seguinte maneira:

$$\bar{l}_{i1} = \frac{1}{\bar{d}_1} a_{i1} = \frac{1}{\bar{d}_1} a_{i1} \frac{d_1}{d_1} = \frac{a_{i1}}{d_1} \frac{d_1}{\bar{d}_1}$$

Por outro lado, $\frac{a_{i1}}{d_1} = l_{i1}$. Logo, conclui-se que

$$\bar{l}_{i1} = l_{i1} \frac{d_1}{\bar{d}_1},$$

para $i = 2, \dots, n$.

Uma vez que uma fórmula para obtenção dos elementos da primeira coluna de $\bar{L}$ está disponível, seria interessante obter valores semelhantes para os elementos $\bar{l}_{ij}$ da segunda e demais colunas de

$\bar{L}$. Para isso será necessário considerar algumas aproximações. Observe que para valores pequenos de α tem-se $\bar{d}_1 \simeq d_1$ e $\frac{d_1}{\bar{d}_1} \simeq 1$.

Os elementos da segunda coluna dos fatores $\bar{D}$ e $\bar{L}$ podem ser calculados fazendo-se $j = 2$ nas equações (4.5.5) e (4.5.6), respectivamente. Assim, usando o valor de $\bar{l}_{21}$ calculado anteriormente e considerando a aproximação $\frac{d_1}{\bar{d}_1} \simeq 1$, obtém-se:

$$\begin{aligned}\bar{d}_2 &= a_{22} + \alpha - \bar{l}_{21}^2 \bar{d}_1 \\ &= a_{22} + \alpha - l_{21}^2 \frac{d_1^2}{\bar{d}_1^2} \bar{d}_1 \\ &= a_{22} + \alpha - l_{21}^2 d_1 \frac{d_1}{\bar{d}_1} \\ &\simeq a_{22} + \alpha - l_{21}^2 d_1 \\ &\simeq d_2 + \alpha.\end{aligned}$$

Para o cálculo dos elementos de $\bar{L}$ que estão abaixo da diagonal, na segunda coluna, serão usados também os elementos disponíveis na primeira coluna, do seguinte modo:

$$\bar{l}_{i2} = \frac{1}{\bar{d}_2} (a_{i2} - \bar{l}_{i1} \bar{l}_{21} \bar{d}_1) \quad (4.5.7)$$

$$\begin{aligned}&= \frac{1}{\bar{d}_2} \left(a_{i2} - l_{i1} \frac{d_1}{\bar{d}_1} l_{21} \frac{d_1}{\bar{d}_1} \bar{d}_1 \right) \\ &= \frac{1}{\bar{d}_2} \left(a_{i2} - l_{i1} l_{21} d_1 \frac{d_1}{\bar{d}_1} \right) \\ &\simeq \frac{1}{\bar{d}_2} (a_{i2} - l_{i1} l_{21} d_1). \quad (4.5.8)\end{aligned}$$

Logo, pode-se considerar a aproximação

$$\bar{l}_{i2} \simeq \frac{1}{\bar{d}_2} (a_{i2} - l_{i1} l_{21} d_1).$$

Multiplicando o lado direito da equação anterior por $\frac{d_2}{\bar{d}_2}$ e arrumando os fatores, obtém-se:

$$\bar{l}_{i2} \simeq \frac{1}{\bar{d}_2} \left(\frac{a_{i2} - l_{i1} l_{21} d_1}{d_2} \right) d_2.$$

Mas, por (4.5.2) tem-se

$$\frac{a_{i2} - l_{i1} l_{21} d_1}{d_2} = l_{i2}.$$

Portanto, para um valor de α pequeno, pode-se considerar a aproximação

$$\bar{l}_{i2} \simeq l_{i2} \frac{d_2}{\bar{d}_2}.$$

Seguindo esse raciocínio, aproximações para elementos $\bar{d}_j$ e $\bar{l}_{ij}$ quaisquer serão calculadas, considerando-se valores pequenos de α . Suponha que todas as $j - 1$ colunas de $\bar{L}$ e $\bar{D}$ já tenham sido calculadas e que $\frac{d_k}{\bar{d}_k} \simeq 1$ para $k = 1, \dots, j - 1$. Assim, o cálculo de elementos aproximados para a coluna j pode ser realizado da seguinte maneira:

$$\begin{aligned}\bar{d}_j &= a_{jj} + \alpha - \sum_{k=1}^{j-1} \bar{l}_{jk}^2 \bar{d}_k \\ &\simeq a_{jj} + \alpha - \sum_{k=1}^{j-1} l_{jk}^2 \frac{d_k^2}{\bar{d}_k^2} \bar{d}_k \\ &\simeq a_{jj} + \alpha - \sum_{k=1}^{j-1} l_{jk}^2 d_k \frac{d_k}{\bar{d}_k} \\ &\simeq a_{jj} + \alpha - \sum_{k=1}^{j-1} l_{jk}^2 d_k \\ &\simeq d_j + \alpha\end{aligned}$$

e

$$\begin{aligned}\bar{l}_{ij} &= \frac{1}{\bar{d}_j} \left(a_{ij} - \sum_{k=1}^{j-1} \bar{l}_{ik} \bar{l}_{jk} \bar{d}_k \right) \\ &\simeq \frac{1}{\bar{d}_j} \left(a_{ij} - \sum_{k=1}^{j-1} l_{ik} \frac{d_k}{\bar{d}_k} l_{jk} \frac{d_k}{\bar{d}_k} \bar{d}_k \right) \\ &\simeq \frac{1}{\bar{d}_j} \left(a_{ij} - \sum_{k=1}^{j-1} l_{ik} l_{jk} d_k \frac{d_k}{\bar{d}_k} \right) \\ &\simeq \frac{1}{\bar{d}_j} \left(a_{ij} - \sum_{k=1}^{j-1} l_{ik} l_{jk} d_k \right).\end{aligned}$$

Por fim, multiplicando e dividindo a equação anterior por d_j e arrumando os fatores, obtém-se a aproximação requerida:

$$\bar{l}_{ij} \simeq l_{ij} \frac{d_j}{\bar{d}_j}. \quad (4.5.9)$$

Desse modo, aproximações $\tilde{L}$ e $\tilde{D}$ para os fatores $\bar{L}$ e $\bar{D}$, respectivamente, da fatoração LDL^T de $\bar{A} = A + \alpha I$ são obtidas através das fórmulas,

$$\tilde{d}_j = d_j + \alpha \quad (4.5.10)$$

para $j = 1, 2, \dots, n$; e

$$\tilde{l}_{ij} = l_{ij} \frac{d_j}{\tilde{d}_j} \quad (4.5.11)$$

para $j = 1, 2, \dots, n$ e $i = j+1, \dots, n$. Por fim, para que o fator aproximado $\tilde{L}$ preserve as características do preconditionador original, os elementos da diagonal de $\tilde{L}$ são definidos com sendo a unidade. Isto é,

$$\tilde{l}_{jj} = 1, \quad j = 1, \dots, n. \quad (4.5.12)$$

Essa aproximação tem a qualidade de que a primeira coluna da matriz $\tilde{L}$ é calculada de modo exato, como na fatoração LDL^T da matriz $\mathcal{A} + \alpha I$. As demais aproximações dependem da magnitude do número α . Quanto ao uso desses fatores aproximados como preconditionadores, pode-se esperar que o mesmo funcione bem para correções que ocorram nas primeiras colunas da matriz e para valores de α pequeno.

4.5.1 Características do Precondicionador Atualizado

A atualização do preconditionador construído com base na fatoração controlada de Cholesky pretende resolver, de maneira eficiente, as falhas que ocorrem na diagonal durante a fatoração de Cholesky incompleta, reduzindo o tempo gasto no preconditionamento. No entanto, como trata-se de uma aproximação do fator de Cholesky incompleto, está sujeito a uma nova falha na diagonal e, conseqüentemente, a uma nova correção na mesma, o que inevitavelmente o distanciará ainda mais do fator de Cholesky. Ainda assim, espera-se que o mesmo tenha um bom desempenho no preconditionamento dos sistemas lineares. A seguir, serão analisadas algumas características do preconditionador atualizado, principalmente, aquelas herdadas do preconditionador original.

- **Padrão de Esparsidade.** O padrão de esparsidade do fator de Cholesky incompleto determinado pela fatoração controlada de Cholesky é mantido pelo novo preconditionador, pois o processo de atualização do fator antigo não cria novos elementos não nulos.
- **Memória requerida.** O processo de atualização do preconditionador não cria novos elementos não nulos. Também não cria novas estruturas matriciais. Deste modo, não há alteração na quantidade de memória requerida para o armazenamento do preconditionador.
- **Relação entre os valores dos elementos do fator L .** Uma característica importante da fatoração controlada de Cholesky é a forma como é feito o controle do preenchimento no fator incompleto: são escolhidos os maiores elementos em módulo. O preconditionador atualizado mantém essa relação. Isto é, após a atualização de cada coluna anteriormente calculada do preconditionador, os elementos destas mantêm a mesma ordenação de valores do preconditionador original. De fato, considere L o fator de Cholesky incompleto calculado pela fatoração controlada de Cholesky. Suponha que durante o processo de fatoração ocorreu uma falha na diagonal na coluna $j + 1$ e que, portanto, os elementos da diagonal da matriz foram incrementados de um valor α . Sem perda de generalidade, analisaremos a atualização realizada na coluna j . Sejam l_{ij} e l_{kj} elementos quaisquer na coluna j antes da atualização e tais que $|l_{ij}| \geq |l_{kj}|$. Após o processo de atualização obtemos elementos $\tilde{l}_{ij}$ e $\tilde{l}_{kj}$ tais que

$$\tilde{l}_{ij} = l_{ij} \frac{d_j}{\tilde{d}_j} \text{ e } \tilde{l}_{kj} = l_{kj} \frac{d_j}{\tilde{d}_j}.$$

Como $\left| \frac{d_j}{\tilde{d}_j} \right| = \left| \frac{d_j}{d_j + \alpha} \right| \geq 0$, temos:

$$|l_{ij}| \geq |l_{kj}| \Rightarrow |l_{ij}| \left| \frac{d_j}{\tilde{d}_j} \right| \geq |l_{kj}| \left| \frac{d_j}{\tilde{d}_j} \right| \Rightarrow \left| l_{ij} \frac{d_j}{\tilde{d}_j} \right| \geq \left| l_{kj} \frac{d_j}{\tilde{d}_j} \right| \Rightarrow |\tilde{l}_{ij}| \geq |\tilde{l}_{kj}|.$$

- **Custo computacional por correção na diagonal.** Ao contrário da fatoração controlada de Cholesky, que na presença de falhas na diagonal pode ter que efetuar várias fatorações até obter o preconditionador, a construção do preconditionador atualizado é obtida sem a necessidade de calcular novas fatorações. O valor de α é determinado uma única vez com base no valor de d_j e de modo que $d_j + \alpha > 0$. Em seguida, é feito o incremento na diagonal e a fatoração continua a partir da coluna j , precedida de uma atualização das colunas anteriormente calculadas. Esse processo é realizado apenas uma vez por falha. Dessa forma, a atualização do preconditionador pode ser mais barata, computacionalmente falando, do que reiniciar a fatoração a cada correção realizada na diagonal. Além disso, como não há a necessidade de reiniciar a fatoração, maiores valores de α podem ser permitidos e assim uma maior amplitude de correções na diagonal.
- **Pertence a classe de preconditionadores $UFP(\mathcal{A}, \Delta_\alpha)$ para $\{\mathcal{A} + \alpha I\}$.**

Essa característica é importante e será apresentada através da Proposição 4.5.1. Este é um resultado central neste trabalho porque uma vez que o preconditionador $\tilde{M}$ pertence à classe $UFP(\mathcal{A}, \Delta_\alpha)$, toda a teoria desenvolvida para essa classe de preconditionadores se aplica ao mesmo. Em particular, valem os Teoremas 4.4.2 e 4.4.3. Desta maneira, pode-se concluir que, para valores pequenos de α , o preconditionador $\tilde{M}$ é uma boa aproximação para a matriz $\mathcal{A} + \alpha I$ e os autovalores da matriz $\tilde{M}^{-1}(\mathcal{A} + \alpha I)$ se agrupam em torno da unidade. Essas duas qualidades são desejáveis em preconditionadores de um modo geral.

Proposição 4.5.1. Suponha que a fatoração $\mathcal{A} = LDL^T$ esteja disponível. Considere $\mathcal{A} + \Delta_\alpha$ com $\Delta_\alpha = \text{diag}(\alpha, \dots, \alpha)$, $\alpha > 0$. O preconditionador $\tilde{M}_\alpha$ definido pelas equações (4.5.10), (4.5.11) e (4.5.12) pertence à classe $UFP(\mathcal{A}, \Delta_\alpha)$.

Demonstração. Primeiro é preciso mostrar que existem fatores L_α e D_α tais que $\tilde{M}_\alpha = L_\alpha D_\alpha L_\alpha^T$. Nesse sentido, considere a matriz

$$I + \text{off}(L)D(D + \alpha I)^{-1}.$$

Note que os elementos da diagonal dessa matriz são todos iguais a 1 e que os elementos fora da diagonal são definidos por $l_{ij} \frac{d_j}{d_j + \alpha}$. Considere a matriz

$$P_\alpha = L_\alpha D_\alpha L_\alpha^T,$$

onde

$$L_\alpha = I + \text{off}(L)D(D + \alpha I)^{-1} \text{ e } D_\alpha = D + \alpha I.$$

Note que

$$(d_j)_\alpha = d_j + \alpha, \text{ para } j = 1, \dots, n$$

e

$$(l_{ij})_\alpha = l_{ij} \frac{d_j}{d_j + \alpha}, \text{ para } j = 1, \dots, n \text{ e } i = j + 1, \dots, n.$$

Dessa forma, $L_\alpha = \tilde{L}$ e $D_\alpha = \tilde{D}$. Assim,

$$\tilde{M}_\alpha = P_\alpha = L_\alpha D_\alpha L_\alpha^T.$$

Agora resta verificar que as propriedades **(A1)**, **(A2)** e **(A3)** na **Definição** 4.4.1 são satisfeitas por $\tilde{M}_\alpha$. De fato, denotando $\Delta_\alpha = \alpha I$, obtém-se $\tilde{D} = D + \Delta_\alpha$ e como $\tilde{d}_j = d_j + \alpha$ para $j = 1, \dots, n$, então a propriedade **(A1)** vale. Agora para mostrar que **(A2)** também vale, basta considerar a igualdade abaixo

$$\|\tilde{D} - D\|_2 = \|\alpha I\| = \alpha = \|\Delta_\alpha\|_2,$$

e tomar $\rho = 1$ na **Definição** 4.4.1 para qualquer valor de α dado.

Finalmente, lembrando que $S_\alpha = DD_\alpha^{-1} = D\tilde{D}^{-1}$ e como $\tilde{l}_{ij} = l_{ij} \frac{d_j}{\tilde{d}_j} = l_{ij} d_j \tilde{d}_j^{-1}$, para $j = 1, \dots, n$ e $i = j + 1, \dots, n$, tem-se

$$\text{off}(\tilde{L}) = \text{off}(L)D\tilde{D}^{-1} = \text{off}(L)S_\alpha.$$

Além disso, como $\tilde{l}_{jj} = 1$ para $j = 1, \dots, n$, então $\text{diag}(\tilde{L}) = (1, \dots, 1)^T = \text{diag}(L)$ e, portanto, **(A3)** vale.

4.5.2 Fatoração Controlada de Cholesky Modificada Beta

Nessa nova proposta de modificação à fatoração controlada de Cholesky, pretende-se evitar o reinício da fatoração da matriz $\mathcal{A} + \alpha I$ quando, durante a construção do preconditionador, ocorre uma falha na diagonal e uma correção é efetuada com um determinado valor α . Para esse fim, será utilizada a idéia de construção de um preconditionador $\tilde{M}$ para a matriz diagonalmente modificada $\mathcal{A} + \alpha I$, a partir da atualização do preconditionador M disponível para a matriz $\mathcal{A}$. No entanto, no contexto de preconditionamento de sistemas lineares provenientes de métodos de pontos interiores, usando a fatoração controlada de Cholesky, os fatores L e D da matriz $\mathcal{A}$ não estão completamente disponíveis, quando uma falha ocorre na diagonal. Sendo assim, a atualização do preconditionador se dará parcialmente, apenas nos elementos calculados antes da ocorrência da falha.

A fatoração controlada de Cholesky é realizada por colunas, logo, no momento que ocorre uma falha durante a fatoração da matriz $\mathcal{A}$, digamos no cálculo da coluna j , estão disponíveis $j - 1$ colunas do preconditionador M . A proposta é aproveitar esses elementos e atualizá-los para construir parte do preconditionador $\tilde{M}$ para a matriz $\mathcal{A} + \alpha I$. Para isso serão utilizadas as fórmulas 4.5.10 e 4.5.11 obtidas na seção anterior. Após o cálculo de um valor adequado de α e da atualização dos elementos das colunas disponíveis, a fatoração da matriz diagonalmente modificada continua normalmente a partir da coluna j .

A seguir será apresentada uma sequência dos principais passos realizados por essa abordagem para a construção do preconditionador, quando falhas ocorrem durante a fatoração controlada de Cholesky.

Quando ocorre uma falha na coluna j durante o processo da fatoração controlada de Cholesky, digamos $d_j \leq tol$, o cálculo do preconditionador $\tilde{M}$ é realizado efetuando as seguintes etapas:

1. Calcule um valor de α tal que $d_j + \alpha > tol$. Faça $\tilde{d}_j = d_j + \alpha$;
2. Atualize os elementos das colunas $k = 1, \dots, j - 1$ de L usando as seguintes fórmulas:

$$d_k = d_k + \alpha, \quad k = 1, \dots, j - 1$$

$$l_{ik} = l_{ik} \frac{d_k}{\tilde{d}_k} = l_{ik} \frac{d_k}{d_k + \alpha}, \quad i = k + 1, \dots, m;$$

3. Finalize o cálculo dos elementos da coluna j com $\tilde{d}_j = d_j + \alpha$;
4. Atualize os elementos da diagonal de $\mathcal{A}$, $a_{kk} = a_{kk} + \alpha$, para $k = j + 1, \dots, m$;
5. Continue o processo de fatoração para as colunas $k = j + 1, \dots, n$.

As etapas da fatoração controlada de Cholesky modificada beta está ilustrada na Figura 4.3.

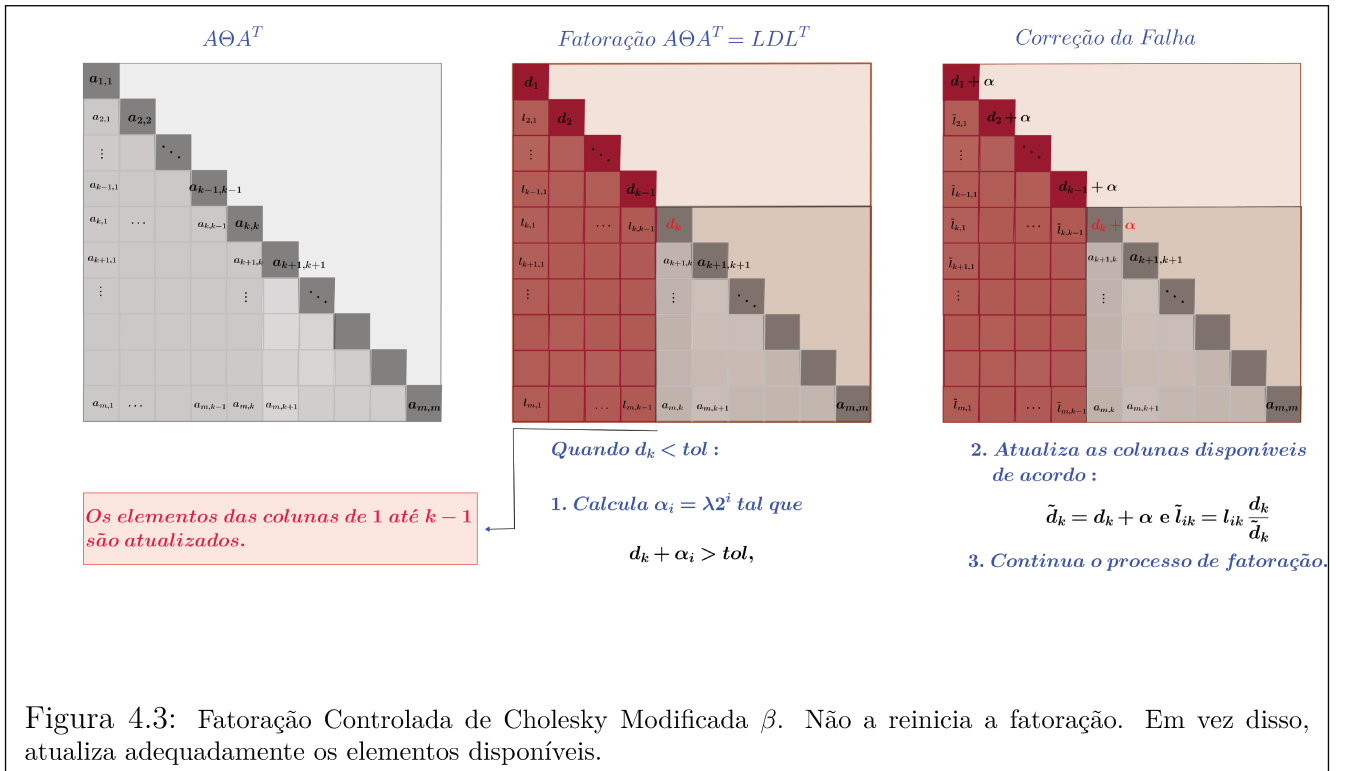


Figura 4.3: Fatoração Controlada de Cholesky Modificada β . Não a reinicia a fatoração. Em vez disso, atualiza adequadamente os elementos disponíveis.

Neste capítulo, foram apresentadas duas propostas para melhorar o desempenho da fatoração controlada de Cholesky no preconditionamento de sistemas lineares oriundos dos métodos de pontos

Tabela 4.2: Correção de falhas na fatoração LDL^T : FCC Original e FCC Modificada

	FCC Original	FCC Modificada	
		FCCα	FCCβ
Procedimento adotado durante a fatoração LDL^T quando ocorre $d_j < tol$:	Calcula α_i e reinicia a fatoração da matriz $\mathcal{A} + \alpha_i I$. Se a fatoração da nova matriz falhar, o processo é repetido até a fatoração ser completada ou i atingir o seu valor máximo i_{max} .	Calcula α_i até que $d_j + \alpha_i > tol$. Em seguida reinicia a fatoração da matriz $\mathcal{A} + \tau \alpha_i I$.	Calcula α_i até que $d_j + \alpha_i > tol$. Em seguida atualiza as colunas $k = 1, \dots, j - 1$ usando fórmulas adequadas e depois prossegue com a fatoração.
Fórmula utilizada para o cálculo do parâmetro α:	$\alpha_i = \lambda 2^i$ onde $\lambda = 5,0 \times 10^{-4}$ (Menos Flexível)	$\alpha_i = \lambda 2^i$ onde $\lambda = 5,0 \times 10^{-4}$ (Mais Flexível)	$\alpha_i = \lambda 2^i$ onde $\lambda = 5,0 \times 10^{-4}$ (Mais Flexível).
Valor máximo permitido para α_i:	Para cada valor de α_i calculado, uma nova fatoração matricial é realizada. Logo, quanto maior for valor de i , maior será o custo computacional para construir o preconditionador.	Apesar de permitir reinícios da fatoração, valores maiores de α podem ser usados na construção do preconditionador sem aumentar muito o custo computacional do mesmo. Assim, a economia de tempo na construção do preconditionador pode compensar um menor desempenho do mesmo.	Permite o uso de valores maiores de α , apesar de funcionar melhor para valores pequenos de α . No entanto, a perda de eficiência do preconditionador também pode ser compensada com o baixo custo de construção do mesmo.

interiores. Ambas estão relacionadas às falhas que ocorrem na diagonal durante o processo de fatoração incompleta e tem o objetivo de reduzir o tempo do preconditionamento dos sistemas lineares envolvidos. A primeira proposta apresentada modifica a fatoração controlada de Cholesky apenas na forma como o parâmetro de correção α é calculado, permite novas fatorações durante a mesma iteração do método de pontos interiores e será denotada por FCC α . Por sua vez, a segunda modificação apresentada também usa a nova estratégia para o cálculo da parâmetro α , no entanto, não permite novas fatorações e, na ocorrência de falhas na fatoração, constrói o preconditionador atualizando os elementos disponíveis antes da falha, por meio de fórmulas pré-determinadas. Não ocorrendo falhas durante a fatoração de Cholesky incompleta, tanto a FCC α como a FCC β se resume à fatoração controlada de Cholesky original. A Tabela 4.2 mostra um quadro comparativo entre a FCC e as duas versões modificadas, FCC α e FCC β . Testes numéricos para avaliar o desempenho de ambas serão apresentados no próximo capítulo.

Capítulo 5

Resultados Numéricos

5.1 Introdução

Neste capítulo, serão apresentados experimentos computacionais realizados com as modificações propostas ao preconditionador fatoração controlada de Cholesky. Os testes numéricos foram realizados no contexto do preconditionamento de sistemas lineares oriundos de métodos de pontos interiores e utilizando a versão preconditionada do *solver* PCx [18]. Sabe-se que preconditionadores provenientes de fatorações de Cholesky incompletas tem um bom desempenho nas iterações iniciais do método de pontos interiores, mas o mesmo não ocorre nas iterações finais do método [38], quando a matriz $A\Theta A^T$ fica muito mal condicionada. Por essa razão os testes foram realizados com a versão preconditionada do PCx utilizando a abordagem híbrida proposta em [12], a qual utiliza o preconditionador fatoração controlada de Cholesky nas iterações iniciais e o preconditionador separador nas últimas iterações do método. O código da fatoração controlada de Cholesky, implementado em FORTRAN por Campos [15], foi modificado de maneira a incorporar as modificações propostas neste estudo e possibilitar a realização dos experimentos. Assim, foram obtidas duas versões da fatoração controlada de Cholesky, as quais estão sendo chamadas de FCC modificada alfa ($FCC\alpha$) e FCC modificada beta ($FCC\beta$). A versão original da fatoração controlada de Cholesky será chamada de FCC original ($FCC\text{ orig}$), ou simplesmente FCC.

Nos experimentos numéricos, além das modificações $FCC\alpha$ e $FCC\beta$ propostas, também foram consideradas duas outras modificações realizadas na FCC original. Tratam-se de correções baseadas na estratégia de correção local de Kershaw [27], mencionada na Seção 4.2, e do preconditionador atualizado proposto por Bellavia et al.[6], o qual foi apresentado na Seção 4.4.1. Os dois novos preconditionadores serão chamados de FCC modificada gama ($FCC\gamma$) e FCC modificada delta ($FCC\delta$), respectivamente.

A comparação entre o desempenho das modificações $FCC\alpha$, $FCC\beta$, $FCC\gamma$ e $FCC\delta$ em relação a FCC original está apresentada em diversas tabelas com as estatísticas obtidas em termos de tempo de resolução dos problemas e do número de iterações do método dos gradientes conjugados. Além de tabelas, a comparação foi realizada usando os perfis de desempenho proposto por Dolan e Moré [19], o qual será apresentado brevemente na Seção 5.1.2.

5.1.1 PCx

O PCx é um eficiente código para resolver problemas de programação linear e sua implementação é baseada no método de pontos interiores preditor-corretor de Mehrotra. O código PCx, assim como todas as suas rotinas, estão implementados em linguagem C. A exceção é a fatoração controlada de Cholesky, que está implementada em FORTRAN. Múltiplas correções, geralmente, não são permitidas em abordagens iterativas e esta opção foi desativada do código. De resto, todos os demais parâmetros padrões do PCx foram adotados. Em todos os experimentos foi utilizado o método dos gradientes conjugados com critério de convergência estabelecido pela norma residual Euclidiana $\|r_k\|$. Para resolver os dois sistemas lineares de equações normais 2.3.12 na primeira fase do condicionamento híbrido, o critério de parada foi fixado como $\|r_k\| < 10^{-4}$. Quando o *gap* de otimalidade é menor do que 10^{-5} ou a mudança de fase é detectada então considera-se $\|r_k\| < 10^{-8}$. O limite de iterações é a dimensão m do sistema linear. A heurística adotada para a mudança de fases foi a proposta em [43] e também utilizada em [21], onde o valor inicial do parâmetro η é definido como sendo Mel ou $-Mel$, como apresentado na Seção 3.3.3. Porém, para os problemas **nug12** e **qap12** foi utilizado o valor $Mel + 100$ e para o problema **ste36a**, o valor $-(Mel + 100)$. Os testes computacionais foram realizados em ambiente Linux, em duas máquinas: a máquina I equipada com processador intel corei5 de 2.67 GHz e 4Gb de memória RAM e a máquina II equipada com processador intel corei7 de 3.60 GHz e 16Gb de memória RAM instalada, sendo utilizável 15,9GB.

5.1.2 Perfil de desempenho

Sejam $\mathcal{P}$ um conjunto de problemas testes com n_p problemas e $\mathcal{S}$ um conjunto de *solvers*, que aqui serão chamados simplesmente de métodos. O perfil de desempenho (*performance profile*) é uma ferramenta utilizada para avaliar e comparar o desempenho do conjunto de métodos $\mathcal{S}$ na resolução dos problemas do conjunto $\mathcal{P}$.

Seja $t_{p,s} \geq 0$ uma estatística correspondente a solução de um problema teste $p \in \mathcal{P}$ por um método $s \in \mathcal{S}$. Esta estatística pode ser, por exemplo, o tempo computacional de resolução do problema ou o número de iterações do método. Suponha que quanto menor for o valor $t_{p,s}$, melhor será considerado o método s . Além disso, seja t_p o valor mínimo atingido na resolução do problema teste p por um método qualquer do conjunto $\mathcal{S}$ em análise, e seja r_M um valor dado tal que $r_M \geq \frac{t_{p,s}}{t_p}$ para todo $s \in \mathcal{S}$. O desempenho do método s na resolução do problema p é comparado com o melhor desempenho t_p obtido por qualquer outro método na resolução deste problema. Isto é, considera-se o quociente de desempenho

$$r_{p,s} = \frac{t_{p,s}}{t_p}.$$

O valor r_M escolhido é tal que $r_M \geq r_{p,s}$ para todo $s \in \mathcal{S}$ e $r_{p,s} = r_M$ se, e somente se, o método s não resolve o problema p . Mostra-se que a escolha de r_M não afeta a avaliação do desempenho dos métodos.

O perfil de desempenho do método s é definido como

$$\pi(r) = \frac{\text{número de problemas } p \in \mathcal{P} \text{ tais que } r_{p,s} \leq r}{n_p}, \quad r \geq 1,$$

Em outras palavras, $\pi(r)$ é a fração de problemas para o qual $t_{p,s}$ está dentro de um fator r do valor mínimo t_p . Em $r = 1$, o perfil de desempenho indica a porcentagem de problemas para os quais o método s é o melhor método, enquanto a porcentagem de problemas que são resolvidos pelo método s é dada por

$$\lim_{r \rightarrow r_M^-} \pi(r).$$

Finalmente, note que quando os valores dos quocientes de desempenho $\frac{t_{p,s}}{t_p}$ têm diferentes magnitudes, pode ser conveniente usar um perfil de desempenho em escala logarítmica, isto é,

$$\pi_{\log}(r) = \frac{\text{número de problemas } p \in \mathcal{P} \text{ tais que } \log(r_{p,s}) \leq r}{n_p}, \quad r \geq 0.$$

Neste trabalho, foi utilizado o sistema de logaritmos na base 2.

5.2 Problemas Testes

Os problemas utilizados para avaliar o desempenho da fatoração controlada de Cholesky modificada, em suas versões α e β , são apresentados nas Tabelas 5.1, 5.2 e 5.3. Foram considerados problemas de programação linear para os quais falhas no pivô durante a fatoração controlada de Cholesky foram detectadas, quando os parâmetros usuais do PCx, da FCC original e da heurística de mudança de fases proposta em [43] foram utilizados.

Os problemas apresentados na Tabela 5.1 são provenientes da coleção KENNINGTON, disponíveis em [1], os da Tabela 5.2 são da coleção QAPLIB [14] e os da Tabela 5.3 são da coleção NETLIB [1]. Além dessas informações, as tabelas citadas apresentam o número de linhas, o número de colunas e o número de elementos não nulos da matriz das restrições A dos respectivos problemas de programação linear. Os dados se referem aos problemas pré-processados. Todos os problemas da coleção KENNINGTON foram resolvidos utilizando-se a máquina II.

5.3 Resultados Computacionais

Os experimentos realizados com as modificações propostas a fatoração controlada de Cholesky incorporaram a nova abordagem para o cálculo do parâmetro de correção α . Dessa forma, quando, durante a fatoração $A\Theta A^T = LDL^T$, tem-se $d_{jj} < tol$, para alguma tolerância tol dada, calcula-se α tal que,

$$d_{jj} + \alpha > tol.$$

O parâmetro α é calculado pela fórmula

$$\alpha = \lambda 2^i,$$

Tabela 5.1: Problemas Testes 1 - KENNINGTON

Problema	Linhas	Colunas	Não Nulos A
cre-a	2989	6692	16939
cre-b	5328	36382	112233
cre-c	2370	5412	13576
cre-d	4094	28601	86704
ex05	832	7805	46248
ex09	1821	18184	104942
osa-07	1081	25030	89316
ken-07	1388	2613	5847
ken11	9964	16740	38157
ken13	22365	36561	82191
ken18	78538	128434	297886
pds-02	2598	7339	14896
pds-06	9145	28472	60075
pds-10	16558	48763	103725
pds-20	32276	106180	226494
pds-40	64265	214385	457538
pds-60	96503	332862	709178
pds-80	126109	430800	916852
pds-100	156243	514577	1096002

Tabela 5.2: Problemas Testes 2 - QAPLIB

Problema	Linhas	Colunas	Não Nulos A
els19	4350	13186	50882
chr22b	5587	10417	36520
chr25a	8149	15325	53725
scr15	2234	6210	24060
scr20	5079	15980	61780
rou20	7359	37640	152980
nug05	148	225	1050
nug05-3rd	1208	1425	5523
nug06	280	486	1680
nug07	474	931	3318
nug08	742	1632	5936
nug12	3192	8856	33528
nug15	6330	22275	85470
qap8	742	1632	5936
qap12	2794	8856	33528
ste36a	27683	15980	512640

Tabela 5.3: Problemas Testes 3 - NETLIB

Problema	Linhas	Colunas	Não Nulos A
25fv47	788	1843	10538
adlittle	55	137	417
agg2	514	750	4558
agg3	514	750	4574
bandm	240	395	1894
blend	71	111	477
bnl2	1964	4008	14037
boeing1	331	697	3104
boeing2	126	265	979
bore3d	81	138	549
capri	241	436	1528
d6cube	403	5443	34233
degen2	442	757	4167
degen3	1501	2604	25425
e226	198	429	2515
etamacro	334	669	1995
finnis	438	935	2332
forplan	121	447	4415
israel	174	316	2443
kb2	43	68	313
maros	655	1437	6634
ganges	1113	1510	6537
nesm	654	2922	13244
nl	6665	14680	45211
pilotnov	848	2117	11922
scfxm3	915	1704	7381
scsd8-2b-64	5130	35910	112770
scsd8-2c-64	5130	35910	112770
scsd8-2r-432	8650	60550	190210
sctap2	1033	2443	7052
sctap3	1408	3268	9383
sierra	1202	2705	7771
stocfor2	1980	2868	8090
stocfor3	15362	22228	62960
woodw	708	5364	19809
bl	5729	12462	33963

onde na FCC original $\lambda = 5,0 \times 10^{-4}$, $i = 15$ é o número máximo de correções permitidas a cada iteração do método de pontos interiores e tol é igual a 10^{-8} .

Inicialmente, o desempenho da $FCC\alpha$ e da $FCC\beta$, no condicionamento híbrido em métodos

de pontos interiores, foi comparado com os resultados provenientes da FCC original. Nesses experimentos, todos os parâmetros da FCC original foram mantidos. Sendo assim, tanto a $FCC\alpha$ como a $FCC\beta$ foram testadas com $\lambda = 5,0 \times 10^{-4}$ e com o número máximo de correções permitidas igual a 15. Em seguida, testes foram realizados com $FCC\gamma$ e $FCC\delta$. Valores menores de λ também foram considerados.

5.3.1 FCC modificada α

A $FCC\alpha$ trata falhas na diagonal calculando um parâmetro $\alpha > 0$ e, em seguida, reinicia a fatoração controlada de Cholesky da matriz

$$A\Theta A^T + \alpha I.$$

Os resultados computacionais apresentados compararam o desempenho do PCx preconditionado com a abordagem híbrida, onde os preconditionadores FCC original ou a $FCC\alpha$ foram utilizados nas iterações iniciais do método de pontos interiores.

A Tabela 5.4 apresenta a média dos tempos de processamento obtidos na resolução dos problemas da coleção KENNINGTON.

Tabela 5.4: Problemas KENNINGTON ($FCC\alpha$)

Problema	Iterações		Tempo de Processamento	
	FCCorig	$FCC\alpha$	FCCorig	$FCC\alpha$
cre-a	27	27	5,59	5,52
cre-b	43	43	30,92	30,74
cre-c	27	27	4,18	4,08
cre-d	42	42	22,49	22,33
ex05	39	39	4,80	4,78
ex09	x	x	x	x
osa-07	24	24	1,38	1,38
ken-07	15	15	0,07	0,07
ken11	23	23	7,76	7,64
ken13	29	29	66,70	65,85
ken18	41	41	715,20	710,70
pds-02	25	25	0,42	0,42
pds-06	39	39	6,28	6,23
pds-10	47	47	13,37	13,32
pds-20	60	60	146,81	146,48
pds-40	78	78	279,72	278,48
pds-60	84	84	767,21	772,77
pds-80	83	83	1071,39	1072,24
pds-100	85	85	1830,72	1823,93

Estes problemas, em particular os do tipo **pds**, são interessantes para o estudo desta abordagem porque são exemplos de problemas para os quais a FCC tem um bom desempenho e o critério

de troca de fases na abordagem híbrida não é atingido [12]. Assim podemos avaliar melhor o desempenho do método ao longo da resolução de tais problemas sem a influência do segundo preconditionador. É importante observar que os problemas **pds**, são oriundos de modelos *multi-commodity* e geram matrizes $A\Theta A^T$ muito esparsas [12].

Nas Tabelas 5.4, 5.5 e 5.6 as duas primeiras colunas indicam o número de iterações do método de pontos interiores e as outras duas indicam a média do tempo de processamento necessário para encontrar a solução de cada problema. O símbolo x indica que o problema não foi resolvido.

A Tabela 5.4 mostra que os melhores resultados, em termo de tempo computacional, para os problemas da coleção KENNINGTON, foram obtidos quando a $FCC\alpha$ foi utilizada. Esse resultado está de acordo com a hipótese de que a técnica poderia reduzir o tempo computacional da resolução dos problemas de programação linear. Com efeito, o número de iterações do método de gradientes conjugados permaneceu o mesmo para a maioria dos problemas desta coleção, conforme está apresentado na Tabela 5.8.

Tabela 5.5: Problemas QAPLIB ($FCC\alpha$)

Problema	Iterações		Tempo de Processamento	
	FCCorig	FCC α	FCCorig	FCC α
els19	31	31	73,56	73,19
chr22b	29	29	23,60	23,64
chr25a	28	28	50,29	50,36
scr15	24	24	10,30	10,35
scr20	x	x	x	x
rou20	24	24	1504,26	1502,48
nug05	x	x	x	x
nug05-3rd	x	x	x	x
nug06	6	6	0,09	0,09
nug07	11	11	0,36	0,36
nug08	9	9	1,01	1,01
nug12	20	20	266,57	260,23
nug15	27	27	1376,18	1375,77
qap8	10	10	1,18	1,18
qap12	20	20	337,51	337,17
ste36a	37	37	13903,60	13886,90

Os resultados obtidos para os problemas das coleções QAPLIB e NETLIB estão apresentados nas Tabelas 5.5 e 5.6, respectivamente, e indicam que o uso da $FCC\alpha$ também reduz o tempo de resolução dos problemas de programação linear quando as duas fases do preconditionamento híbrido são realizadas.

Conforme pode ser visto na Tabela 5.7, houve uma redução no tempo total de resolução para 53,52% dos problemas testados, contra 11,27% que obtiveram um tempo de resolução maior quando comparados com o tempo computacional apresentando pela FCC original. Os demais problemas apresentaram o mesmo tempo de resolução para os dois preconditionadores. É importante

Tabela 5.6: Problemas NETLIB ($FCC\alpha$)

Problema	Iterações		Tempo de Processamento	
	FCCorig	$FCC\alpha$	FCCorig	$FCC\alpha$
25fv47	31	31	3,19	3,18
adlittle	12	12	0,01	0,01
agg2	34	34	0,82	0,82
agg3	x	x	x	x
bandm	17	17	0,10	0,11
blend	10	10	0,01	0,01
bnl2	51	51	5,10	5,09
boeing1	20	20	0,27	0,26
boeing2	14	14	0,04	0,04
bore3d	16	16	0,02	0,02
capri	20	20	0,07	0,07
d6cube	19	19	0,84	0,84
degen2	12	12	0,30	0,29
degen3	16	16	6,75	6,74
e226	32	32	0,31	0,31
etamacro	30	30	0,24	0,23
finnis	26	26	0,18	0,17
forplan	24	24	0,13	0,12
israel	21	21	0,25	0,25
kb2	13	13	0,01	0,01
maros	23	23	1,87	1,86
ganges	18	18	0,94	0,93
nesm	31	31	2,74	2,72
nl	41	41	41,07	41,04
pilotnov	18	18	0,84	0,83
scfxm3	20	20	1,18	1,17
scsd8-2b-64	7	7	2,09	2,08
scsd8-2c-64	7	7	2,06	2,05
scsd8-2r-432	18	18	16,90	16,87
sctap2	14	14	0,08	0,09
sctap3	16	16	0,15	0,15
sierra	21	21	0,14	0,14
stocfor2	21	21	1,82	1,81
stocfor3	32	32	109,78	109,81
woodw	31	31	0,63	0,63
bl	38	38	27,06	27,05

observar, que apesar da $FCC\alpha$ ter reduzido o tempo de resolução total da maioria dos problemas de

programação linear, tal redução é inferior a 1% em relação ao tempo utilizado pela FCC original.

Tabela 5.7: Desempenho da FCC α em relação FCC original

Coleção	Melhor Tempo	Mesmo Tempo	Pior Tempo
NETLIB	19	14	3
QAPLIB	6	7	3
KENNINGTON	13	4	2
%	53,52	35,21	11,27

O número de iterações do método dos gradientes conjugados preconditionado, quando a FCC α foi utilizada, para alguns problemas testes pode ser visto na Tabela 5.8. Vale notar que para alguns problemas, como os problemas do tipo **pds**, **ken18** e **qap12** o número de iterações realizadas pelo método dos gradientes conjugados usando a FCC α é igual ao número de iterações obtido quando a FCC original é utilizada. Já para os problemas **scr15** e **rou20** houve uma redução neste número. Este resultado indica que a nova maneira de construir o preconditionador, em geral, não afeta a

Tabela 5.8: Iterações do Gradiente Conjugados (FCC α).

Problema	FCCorig	FCC α	Problema	FCCorig	FCC α
pds-02	1261	1261	els19	17600	18108
pds-06	7121	7121	chr22b	22953	24196
pds-10	6969	6969	chr25a	26919	34606
pds-20	57016	57016	scr15	10621	10366
pds-40	42353	42353	rou20	46403	44859
pds-60	78211	78211	ken11	14302	15718
pds-80	80368	80368	ken18	146057	146057
pds-100	125581	125581	qap12	50726	50726

qualidade do mesmo, porém, ao evitar que algumas fatorações sejam realizadas, tem a vantagem de reduzir o tempo computacional de resolução do problema. Por outro lado, para os problemas **els19** e **ken11**, houve um aumento no número de iterações do método dos gradientes conjugados, mas ainda assim o tempo de resolução desses problemas foi menor. Já para os problemas **chr22b** e **chr25a** o aumento no número de iterações do método dos gradientes conjugados foi acompanhado de aumento no tempo de resolução dos mesmos. Esses resultados estão de acordo com as tabelas 5.4, 5.5 e 5.6 nas quais, em geral, o tempo de processamento apresentado pela FCC α é menor do que o apresentado pela FCC original. É importante ressaltar que a FCC α não impede que novas fatorações sejam iniciadas, logo quando isso ocorre, também pode aumentar o tempo de resolução dos problemas de programação linear.

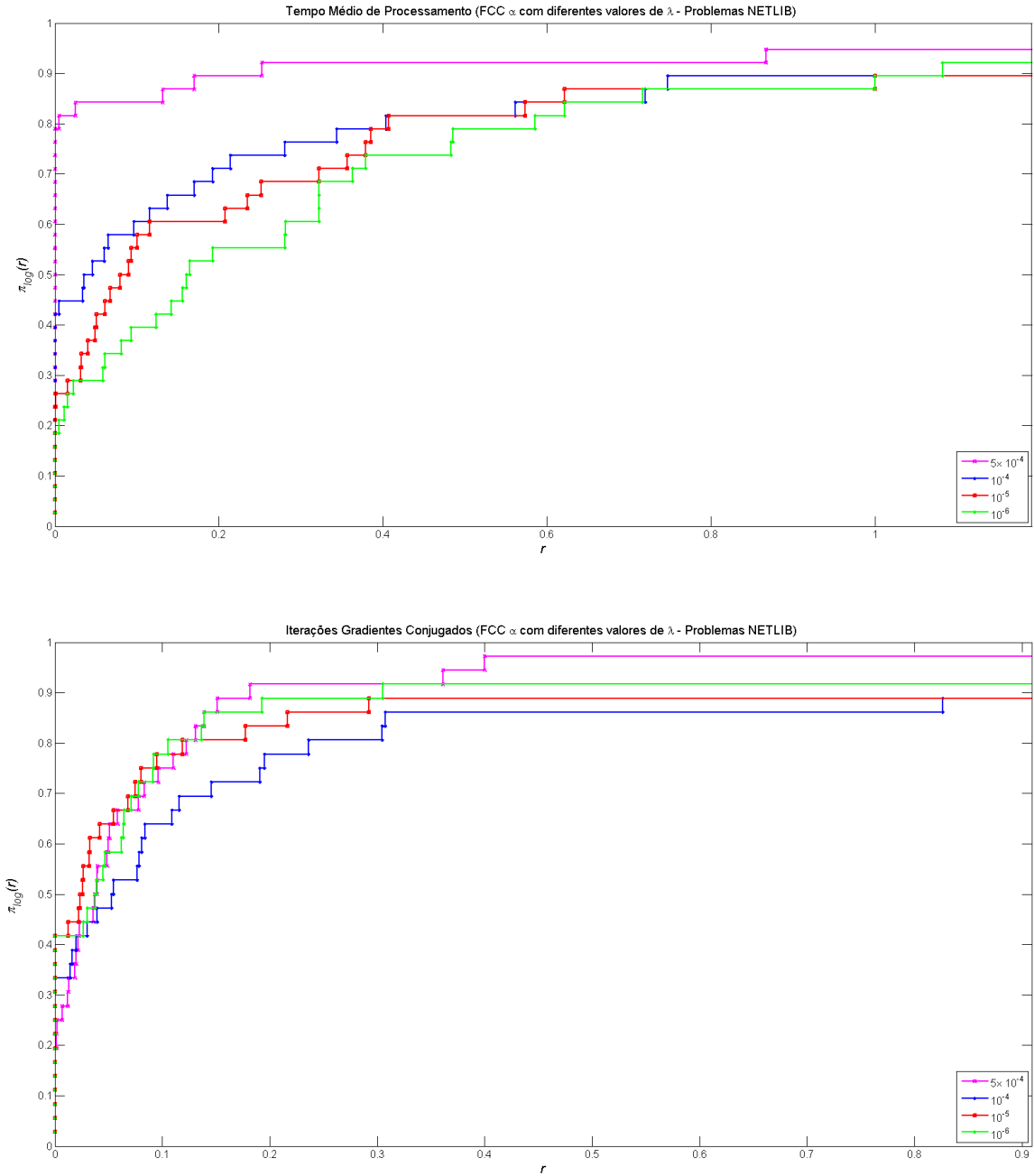


Figura 5.1: Perfil de desempenho da FCC α com diferentes valores de λ . Valores menores reduzem o número de iterações do método dos gradientes conjugados, mas aumentam o tempo médio de resolução dos problemas.

A Figura 5.1 apresenta a curva de desempenho do PCx quando a FCC α é utilizada com os seguintes valores de λ : $5,0 \times 10^{-4}$, 10^{-4} , 10^{-5} e 10^{-6} . A primeira curva leva em consideração a média do tempo de processamento dos problemas e a segunda o número de iterações do método dos gradientes conjugados. Os demais parâmetros permaneceram os mesmos da FCC original. Observa-se que apesar das curvas de desempenho do método dos gradientes conjugados, em termos do número de iterações, serem próximas para praticamente todos os valores de λ , o desempenho do PCx, em termos de tempo computacional, é melhor quando a FCC α é utilizada com os valores $\lambda = 5,0 \times 10^{-4}$, que é o valor usado pela FCC original, e $\lambda = 10^{-4}$. No entanto, é interessante observar o bom desempenho do método dos gradientes conjugados quando a FCC α é utilizada com os valores $\lambda = 10^{-5}$ e $\lambda = 10^{-6}$. Ou seja, como era esperado, perturbações menores na diagonal da matriz $A\Theta A^T$ podem render preconditionadores melhores.

5.3.2 FCC Modificada sem o reinício de fatorações

Nesta seção, serão apresentados os resultados obtidos com as modificações realizadas no preconditionador fatoração controlada de Cholesky que evitam o reinício da fatoração. Foram realizados testes com a FCC β , a FCC γ e a FCC δ . A primeira modificação trata-se da atualização do preconditionador proposta em 4.5.2, a segunda apenas efetua a correção adequada no pivô deficiente e continua o processo de fatoração normalmente e a última é uma adaptação do preconditionador atualizado apresentado em 4.4.1.1.

5.3.2.1 FCC modificada β

A comparação entre os resultados obtidos com a FCC β e os resultados obtidos com a FCC original podem ser visualizada nas tabelas 5.9, 5.10 e 5.11. Os resultados para os problemas da coleção KENNINGTON estão apresentados nas tabelas 5.9. Os resultados obtidos para os problemas **pds** mostram que a FCC β não tem um bom desempenho para esse tipo de problema. Por exemplo, para resolver o problema **pds-40**, a FCC β necessita aproximadamente 10,2 vezes mais tempo do que a FCC original, como pode ser conferido na Tabela 5.9.

Contudo, vale a pena observar que, entre os problemas do tipo **pds**, o número de iterações do método de pontos interiores é maior apenas no problema **pds-02**, e que há uma redução no número de iterações para os problemas **pds-06**, **pds-20** e **pds-80**. O aumento significativo no tempo de resolução desses problemas pode estar associado a mudança de fase no preconditionamento, que para esse tipo de problemas, não ocorre com a FCC original mas ocorre quando a FCC β é utilizada.

A Tabela 5.10 apresenta os resultados obtidos para os problemas da coleção QAPLIB. É importante observar que os problemas **scr20** e **nug05-3rd** não foram resolvidos pela FCC original, o problema **nug15** não foi resolvido com o uso da FCC β e o problema **nug05** não foi resolvido por nenhuma das duas fatorações em questão, quando as mesmas foram utilizadas com os parâmetros usuais do PCx e da fatoração controlada de Cholesky.

Os resultados mostram que para essa classe de problemas, a FCC β pode ser mais vantajosa que a FCC original. Problemas do tipo QAP são modelos para problemas de atribuição quadrática (*Quadratic Assignment Problem*) e são um dos grandes desafios de otimização combinatória [24]. A matriz de equações normais oriundas desses problemas não é muito esparsa e a fatoração de

Tabela 5.9: Problemas KENNINGTON ($FCC\beta$)

Problema	Iterações		Tempo	
	FCCorig	FCC β	FCCorig	FCC β
cre-a	27	27	5,59	3,12
cre-b	43	44	30,92	34,91
cre-c	27	26	4,18	2,12
cre-d	42	43	22,49	24,49
ex05	39	65	4,80	9,04
ex09	x	54	x	47,17
osa-07	24	24	1,38	1,25
ken-07	15	16	0,07	0,11
ken11	23	23	7,76	11,52
ken13	29	29	66,70	63,92
ken18	41	41	715,20	571,22
pds-02	25	29	0,42	1,34
pds-06	39	38	6,28	29,14
pds-10	47	47	13,37	124,23
pds-20	60	59	146,81	649,30
pds-40	78	78	279,72	2845,05
pds-60	84	84	767,21	8742,34
pds-80	83	79	1071,39	11995,94
pds-100	85	85	1830,72	23432,78

Tabela 5.10: Problemas QAPLIB ($FCC\beta$)

Problema	Iterações		Tempo	
	FCCorig	FCC β	FCCorig	FCC β
els19	31	31	73,56	77,97
chr22b	29	29	23,60	26,81
chr25a	29	29	50,29	51,16
scr15	24	24	10,30	10,41
scr20	x	20	x	84,64
rou20	24	24	1504,26	1320,63
nug5	x	x	x	x
nug5-3rd	x	6	x	1,12
nug6	6	6	0,09	0,06
nug7	11	11	0,36	0,38
nug8	9	9	1,01	0,86
nug12	20	21	266,57	102,21
nug15	27	x	1376,18	x
qap8	10	10	1,18	1,07
qap12	20	26	337,51	90,57
ste36a	37	37	13903,60	13884,25

Tabela 5.11: Problemas NETLIB ($FCC\beta$)

Problema	Iterações		Tempo	
	FCCorig	FCC β	FCCorig	FCC β
25fv47	31	38	3,17	6,45
adlittle	12	12	0,01	0,01
agg2	34	28	0,82	0,44
agg3	x	x	x	x
bandm	17	29	0,10	0,36
blend	10	11	0,01	0,01
bnl2	51	37	5,08	6,09
boeing1	20	20	0,26	0,18
boeing2	14	17	0,04	0,04
bore3d	16	17	0,01	0,02
capri	20	23	0,07	0,07
d6cube	19	21	0,84	2,25
degen2	12	12	0,29	0,24
degen3	16	16	6,77	6,56
e226	32	23	0,31	0,19
etamacro	30	29	0,24	0,14
finnis	26	28	0,17	0,18
forplan	24	31	0,13	0,24
israel	21	28	0,26	0,15
kb2	13	16	0,01	0,01
maros	23	27	1,86	2,98
ganges	18	17	0,93	0,62
nesm	31	31	2,72	2,99
nl	41	47	41,04	115,54
pilotnov	18	x	0,83	x
scfxm3	20	23	1,17	2,34
scsd8-2b-64	7	7	2,08	2,08
scsd8-2c-64	7	7	2,05	2,05
scsd8-2r-432	18	18	16,87	16,87
sctap2	14	14	0,08	0,08
sctap3	16	15	0,15	0,15
sierra	21	25	0,14	0,79
stocfor2	21	21	1,81	2,22
stocfor3	32	32	109,68	124,57
woodw	31	32	0,63	0,70
bl	38	38	27,06	58,26

Cholesky tem um grande número de entradas não nulas, por isso, a resolução do sistema linear por meio de um método direto é menos efetiva do que quando resolvida com o método dos gradientes conjugados preconditionado. Além disso, pode-se tirar vantagens da versatilidade da fatoração controlada de Cholesky e usar um parâmetro negativo η de modo a construir preconditionadores

com um baixo custo para essa classe de problemas. No entanto, em geral, muitas falhas na diagonal acontecem durante o processo de fatoração, o que encarece a resolução dos problemas.

Conforme apresentado na Tabela 5.10, a $FCC\beta$ obteve melhor tempo computacional em 10 dos 14 problemas resolvidos da coleção QAPLIB. Esses resultados são promissores e indicam que o preconditionador pode ter um desempenho satisfatório na resolução de problemas dessa classe. Nota-se ainda que o número de iterações do PCx aumentou apenas nos problemas, **nug12** e **qap12**, mas ainda assim, nesses dois casos, houve uma redução significativa no tempo de resolução dos mesmos.

Os resultados obtidos com a resolução dos problemas da coleção NETLIB estão apresentados na Tabela 5.11. Estes resultados indicam que com os parâmetros da FCC original, a $FCC\beta$ tem desempenho inferior ao preconditionador original. Porém, o problema **israel**, da coleção NETLIB, é um importante exemplo de que a atualização de preconditionadores pode ser mais eficiente que a reconstrução de novos preconditionadores, pois para este problema o número de iterações do método de pontos interiores aumentou em cerca de 33%, enquanto o tempo de processamento foi reduzido em aproximadamente 42%.

5.3.2.2 FCC modificada γ

Os resultados obtidos com a $FCC\gamma$ estão apresentados nas Tabelas 5.12, 5.13 e 5.14.

Tabela 5.12: Problemas KENNINGTON ($FCC\gamma$)

Problema	Iterações		Tempo	
	FCCorig	$FCC\gamma$	FCCorig	$FCC\gamma$
cre-a	27	27	5,59	3,05
cre-b	43	44	30,92	34,63
cre-c	27	26	4,18	2,08
cre-d	42	43	22,49	24,19
ex05	39	65	4,80	9,00
ex09	x	54	x	46,87
osa-07	24	24	1,38	1,25
ken-07	15	15	0,07	0,07
ken11	23	23	7,76	6,53
ken13	29	29	66,70	43,75
ken18	41	35	715,20	334,56
pds-02	25	29	0,42	1,26
pds-06	39	38	6,28	28,89
pds-10	47	48	13,37	124,23
pds-20	60	61	146,81	524,93
pds-40	78	77	279,72	2609,73
pds-60	84	84	767,21	8745,93
pds-80	83	79	1071,39	12898,51
pds-100	85	86	1830,72	23523,00

Como pode ser visto na Tabela 5.12, o uso da $FCC\gamma$ reduziu o tempo de resolução dos problemas

ken11, **ken13** e **ken18** da coleção KENNINGTON em cerca de 51% em média. Contudo, os resultados obtidos para problemas do tipo **pds** foram semelhantes aos obtidos com a $FCC\beta$, o que indica que a $FCC\gamma$ também não trabalha bem para este tipo de problema. No entanto, excetuando-se os problemas **pds**, a $FCC\gamma$ apresenta um bom desempenho para os demais problemas dessa classe, conforme mostra o perfil de desempenho apresentado na Figura 5.3.

A Tabela 5.13 apresenta os resultados para os problemas da coleção QAPLIB. Para essa classe de problemas, a média dos tempos computacionais obtidos com a $FCC\gamma$, embora um pouco maior, são parecidos com os resultados obtidos pela $FCC\beta$. Além disso, com os parâmetros da FCC original, o problema **ste36a** não foi resolvido pela $FCC\gamma$.

Tabela 5.13: Problemas QAPLIB ($FCC\gamma$)

Problema	Iterações		Tempo	
	FCCorig	$FCC\gamma$	FCCorig	$FCC\gamma$
els19	31	31	73,56	77,96
chr22b	29	29	23,60	27,76
chr25a	29	28	50,29	56,65
scr15	24	24	10,30	10,38
scr20	x	20	x	84,67
rou20	24	24	1504,26	1320,89
nug5	x	x	x	x
nug5-3rd	x	x	x	x
nug6	6	6	0,09	0,06
nug7	11	11	0,36	0,40
nug8	9	10	1,01	1,07
nug12	20	21	266,57	103,07
nug15	27	x	1376,18	x
qap8	10	10	1,18	1,07
qap12	20	26	337,51	90,55
ste36a	37	x	13903,60	x

Conforme apresentado na Tabela 5.14 e na Figura 5.4, o desempenho da $FCC\gamma$ para os problemas da coleção NETLIB também é inferior ao desempenho da FCC original, mas está bem próximo do desempenho da $FCC\beta$, porém essa última leva uma pequena vantagem no número de problemas resolvidos com melhores tempos computacionais.

5.3.2.3 FCC modificada δ

Em geral, o desempenho da $FCC\delta$ foi inferior aos das outras modificações realizadas na fatoração controlada de Cholesky na resolução das três classes de problemas. Por isso, optou-se por não apresentar tabelas contendo os resultados obtidos com a mesma. No entanto, o seu perfil de desempenho está apresentado nas Figuras 5.2, 5.3 e 5.4. Vale à pena destacar ainda que para 15 problemas da coleção NETLIB, o número de iterações do método de pontos interiores é menor do que o obtido quando a $FCC\beta$ e a $FCC\gamma$ foram utilizadas. Além disso, em 5 desses problemas, o

Tabela 5.14: Problemas NETLIB ($FCC\gamma$)

Problema	Iterações		Tempo	
	FCCorig	FCC γ	FCCorig	FCC γ
25fv47	31	38	3,17	6,45
adlittle	12	12	0,01	0,01
agg2	34	28	0,82	0,72
agg3	x	x	x	x
bandm	17	29	0,10	0,37
blend	10	11	0,01	0,01
bnl2	51	37	5,08	6,09
boeing1	20	20	0,26	0,16
boeing2	14	17	0,04	0,04
bore3d	16	17	0,01	0,02
capri	20	22	0,07	0,08
d6cube	19	21	0,84	2,25
degen2	12	12	0,29	0,24
degen3	16	16	6,77	6,54
e226	32	23	0,31	0,18
etamacro	30	29	0,24	0,14
finnis	26	28	0,17	0,19
forplan	24	31	0,13	0,24
israel	21	28	0,26	0,16
kb2	13	16	0,01	0,01
maros	23	27	1,86	2,98
ganges	18	20	0,93	0,57
nesm	31	31	2,72	3,00
nl	41	44	41,04	121,59
pilotnov	18	x	0,83	x
scfxm3	20	23	1,17	2,51
scsd8-2b-64	7	7	2,08	2,08
scsd8-2c-64	7	7	2,05	2,05
scsd8-2r-432	18	18	16,87	16,87
sctap2	14	14	0,08	0,08
sctap3	16	15	0,15	bf0,14
sierra	21	27	0,14	0,80
stocfor2	21	21	1,81	2,22
stocfor3	32	32	109,68	124,19
woodw	31	32	0,63	0,70
bl	38	38	27,06	64,08

número de iteração do método de pontos interiores foi menor do que o obtido com o uso da FCC original.

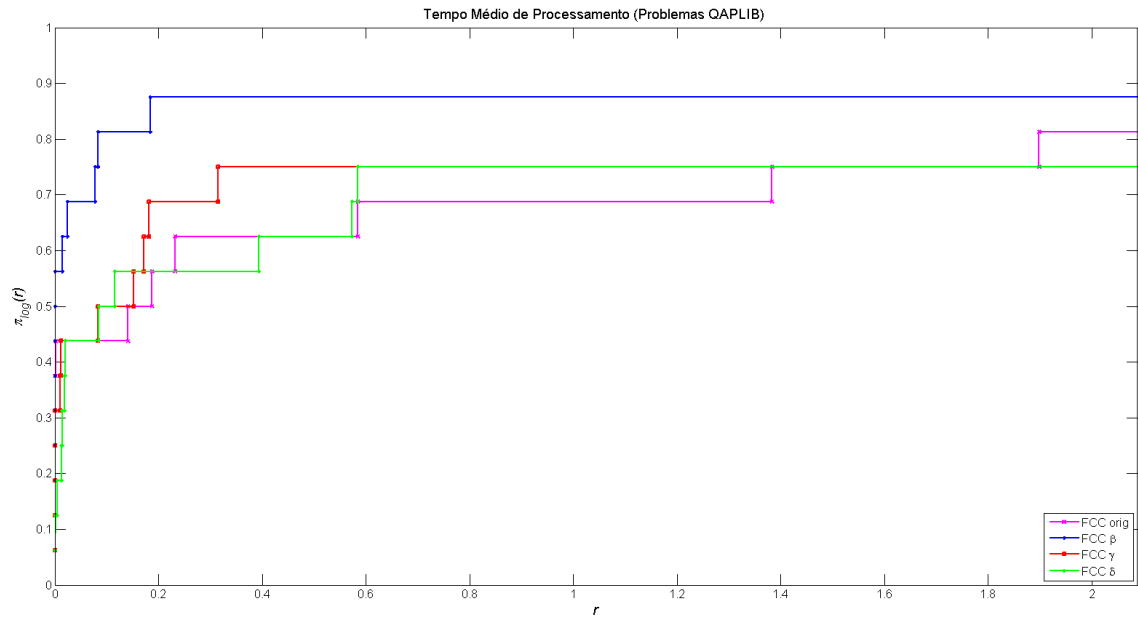


Figura 5.2: Perfil de desempenho para os problemas QAPLIB. A FCC β tem melhor desempenho.

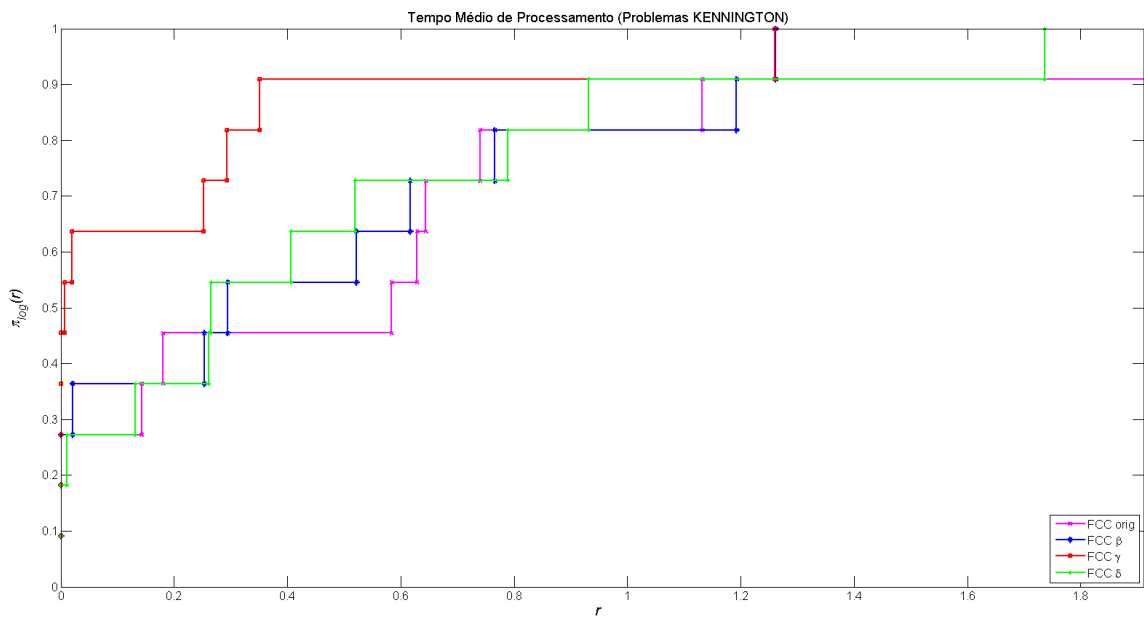


Figura 5.3: Perfil de desempenho para os problemas KENNINGTON. A FCC γ tem melhores resultados.

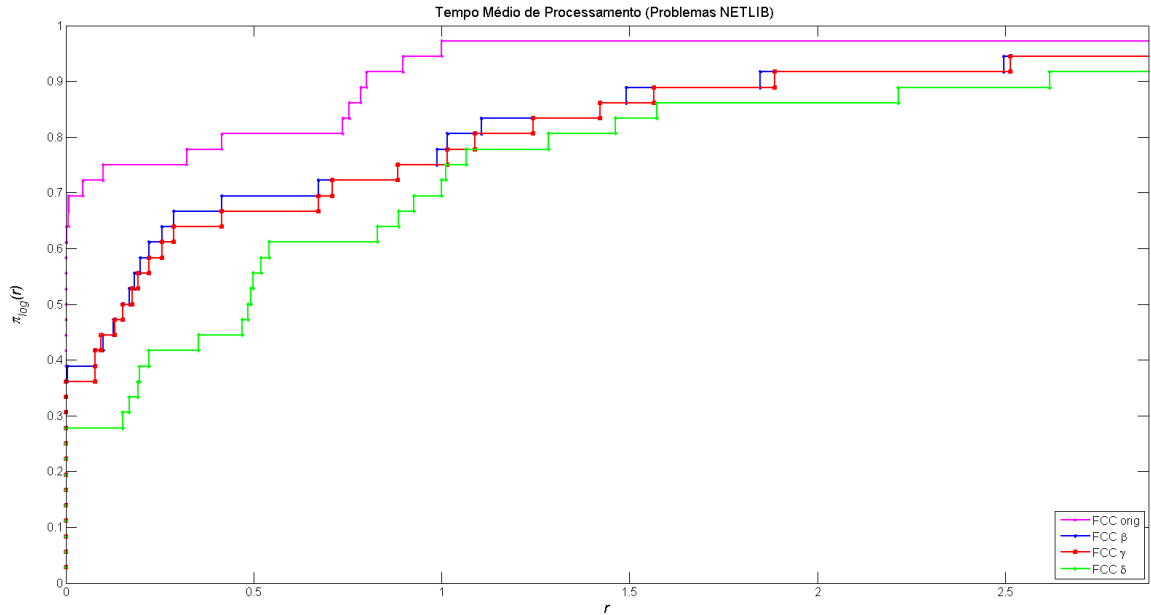


Figura 5.4: Perfil de desempenho para problemas NETLIB. A FCC original tem melhor desempenho.

5.4 Algoritmo de Ajustamento Ótimo para p Coordenadas

O algoritmo de ajustamento ótimo para p coordenadas é um método que resolve problemas de programação linear priorizando p coordenadas a cada iteração [40]. Isto é, ele encontra o valor ótimo para p variáveis e ajusta as demais coordenadas de acordo com esse valor, onde p é limitado pelo tamanho do problema. Suas principais vantagens são simplicidade e rápido avanço inicial. Em cada iteração é realizado apenas um produto de matriz por vetor e um sistema linear positivo definido, com dimensões menores que o tamanho do problema, é resolvido. Por outro lado, sua convergência é muito lenta.

O avanço inicial rápido do algoritmo foi utilizado em [21] para melhorar a abordagem de preconditionamento híbrido em método de pontos interiores. Seguindo essa linha, experimentos utilizando o algoritmo também foram realizados para avaliar o desempenho dos preconditionadores $FCC\alpha$ e $FCC\beta$. Para esse experimento foram considerados apenas problemas da coleção QAPLIB. Os parâmetros utilizados no algoritmo de ajustamento ótimo foram os mesmos utilizados em [21].

Os resultados obtidos mostraram que os ganhos obtidos com a abordagem foram mantidos pela $FCC\beta$ em sete dos treze problemas testados.

5.5 Potencialidades da $FCC\beta$

A fatoração controlada de Cholesky modificada nas versões β e γ possuem a vantagem de evitar o reinício do processo de fatoração matricial, quando pivôs muito pequenos ou negativos aparecem. Essa é uma característica importante porque pode reduzir o tempo computacional necessário para a construção do preconditionador. No entanto, o preço a pagar é a obtenção de um preconditionador de menor qualidade. Porém, espera-se obter melhores preconditionadores à medida que valores menores de α sejam utilizados nas correções. Assim, obter preconditionadores razoáveis a um baixo custo computacional pode ser uma estratégia vantajosa na resolução de diversos problemas de programação linear. E esse é o objetivo dos preconditionadores $FCC\beta$ e $FCC\gamma$.

Tabela 5.15: Iterações do método dos GC com o parâmetro $\lambda = 10^{-6}$

Problema	Iterações Pontos Interiores		Iterações GC	
	$FCC\gamma$	$FCC\beta$	$FCC\gamma$	$FCC\beta$
pds-02	26	26	4926	4540
pds-06	38	40	27994	32273
pds-10	47	47	52217	28293
pds-20	61	60	78835	84138
pds-40	78	77	84831	198254
pds-60	84	84	441938	263036
pds-80	82	83	340105	195639
pds-100	86	83	590806	340125

Com o objetivo de avaliar o desempenho da $FCC\beta$ na presença de correções realizadas na diagonal com valores pequenos de α , foram realizados testes numéricos com a $FCC\beta$ e a $FCC\gamma$ utilizando parâmetros diferentes dos utilizados pela FCC original. Foram considerados os valores 10^{-4} , 10^{-5} e 10^{-6} para λ . A tolerância no número máximo de correções permitidas foi aumentada para 50. Inicialmente, testes foram realizados com os problemas do tipo **pds**, pois para estes problemas a $FCC\beta$ com os parâmetros originais não obteve um bom desempenho.

Tabela 5.16: Tempo médio de resolução com o parâmetro $\lambda = 10^{-6}$

Problema	Iterações Pontos Interiores		Tempo de execução	
	$FCC\gamma$	$FCC\beta$	$FCC\gamma$	$FCC\beta$
pds-02	26	26	1,10	0,73
pds-06	38	40	26,94	18,21
pds-10	47	47	96,17	30,74
pds-20	61	60	482,56	292,62
pds-40	78	77	1815,70	1047,95
pds-60	84	84	4314,64	2539,47
pds-80	82	83	11790,40	3048,70
pds-100	86	83	9574,11	5983,10

A Tabela 5.15 mostra os resultados obtidos para tais problemas, quando $\lambda = 10^{-6}$ foi utilizado.

Nota-se que com a $FCC\beta$, o PCx realiza, em média, apenas 61,18% das iterações do método dos gradientes conjugados realizadas quando a $FCC\gamma$ é utilizada. Além disso, o tempo total de resolução dos problemas **pds** foi menor com a $FCC\beta$ do que com a $FCC\gamma$. Logo, para essa classe de problemas, o preconditionador $FCC\beta$ tem melhor desempenho do que o preconditionador $FCC\gamma$, ainda que em termos de tempo computacional, este possa ser mais caro de construir.

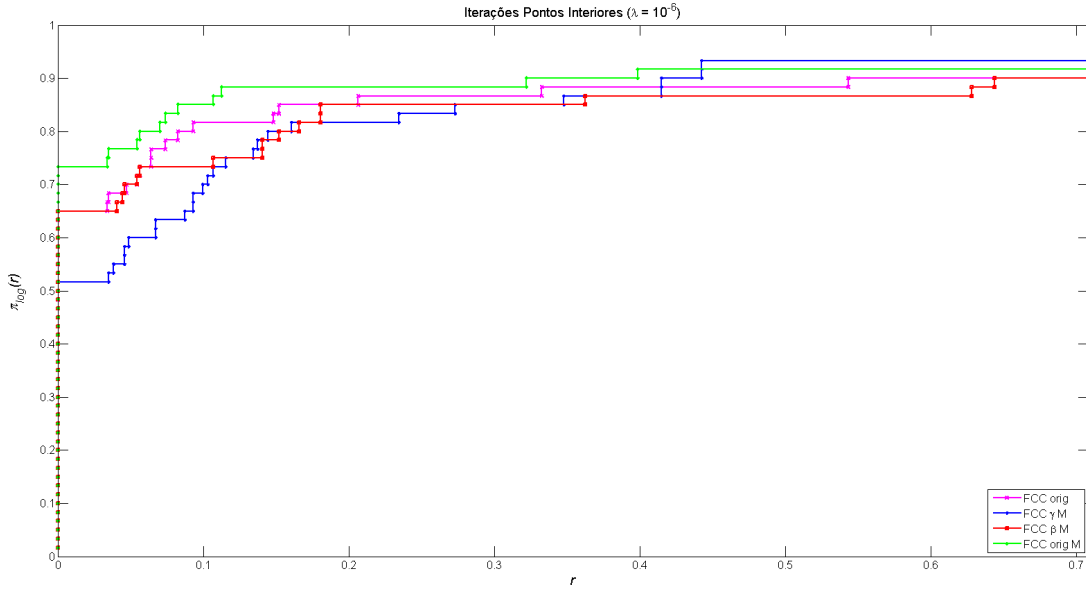


Figura 5.5: Desempenho em termos de iterações do método de pontos interiores: $FCC\beta$ M e FCCorig têm desempenhos parecidos dentro do fator $r = 0,1$.

A mudança nos parâmetros utilizados pela FCC original reduziu consideravelmente o tempo computacional utilizado pelo PCx quando a $FCC\beta$ foi utilizada na resolução dos problemas **pds**. Conforme pode ser visto na Tabela 5.16, o tempo necessário para a resolução do problema **pds-40** foi reduzido de 10,17 vezes para 3,75 vezes, quando comparado com o tempo gasto quando a FCC original é utilizada. Já o tempo necessário para resolver o problema **pds-20** foi reduzido de 4,42 vezes para aproximadamente 2 vezes.

Experimentos com $\lambda = 10^{-6}$ também foram realizados com os demais problemas testes apresentados nas tabelas 5.1 (KENNINGTON), 5.2 (QAPLIB) e 5.3 (NETLIB), com exceção dos problemas **nug15** e **ste36a**. As figuras 5.5, 5.6 e 5.7 apresentam as curvas de desempenho obtidas para esses problemas, considerando o número de iterações do método de pontos interiores, o número de iterações dos métodos dos gradientes conjugados e o tempo computacional gasto na resolução de tais problemas. Nessas figuras, o símbolo M ao lado do nome, do respectivo preconditionador, indica que o mesmo está usando um novo valor para o parâmetro λ . A Figura 5.5 indica que, considerando o número de iterações do método de pontos interiores, o perfil de desempenho dos preconditionadores são parecidos.

Quando o número de iterações do método dos gradientes conjugados é considerado, conforme

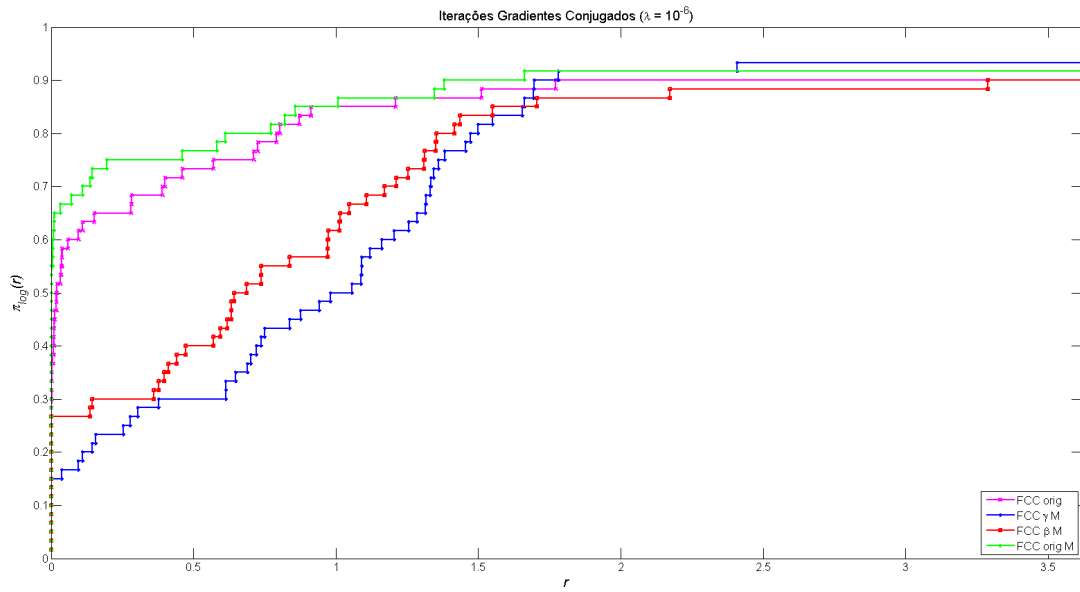


Figura 5.6: FCC β M tem melhor desempenho que FCC γ M, com $\lambda = 10^{-6}$, dentro do fator $r = 1,5$.

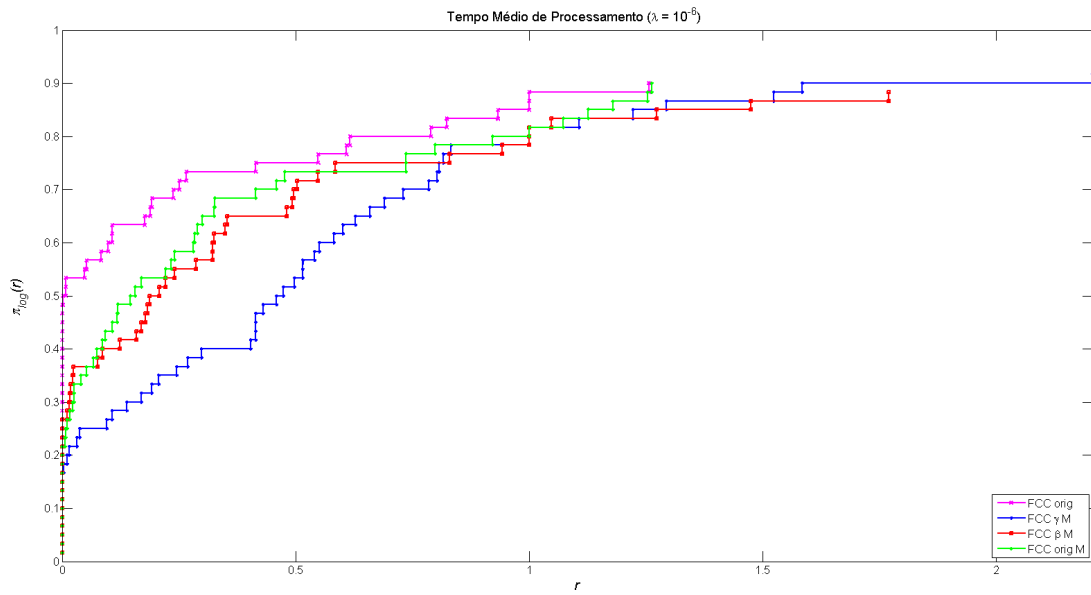


Figura 5.7: Em termos de tempo computacional, e dentro do fator $r = 1$, o desempenho da FCC β M é melhor do que o da FCC γ M, mas está abaixo do desempenho da FCC original.

mostra a Figura 5.6, a FCC original, tanto com o valor original de λ , quanto com o novo valor, obtém perfis de desempenho superiores aos demais, como já era esperado. No entanto, com relação ao tempo computacional utilizado na resolução dos problemas, FCC β M e FCC original M apresentam desempenhos muito próximos, conforme apresentado na Figura 5.7. Dessa forma, pode-se concluir que a estratégia de atualização do preconditionador fatoração controlada de Cholesky pode reduzir o tempo de resolução de problemas de programação linear através do PCx preconditionado com estratégias de preconditionamento híbridos.

5.6 FCC modificada $\times$ FCC original

As curvas de desempenho, considerando o tempo utilizado na resolução dos problemas testes, com cada um dos preconditionadores FCC α , FCC β e FCC original utilizando os parâmetros originais da fatoração controlada de Cholesky, estão apresentadas na Figura 5.8. Neste experimento, os problemas **pds** não foram considerados, visto que o desempenho da FCC β com os parâmetros originais foram bastantes discrepantes em relação aos demais problemas. Observa-se que para algum fator de tempo r cada uma das curvas apresenta um desempenho melhor que as demais.

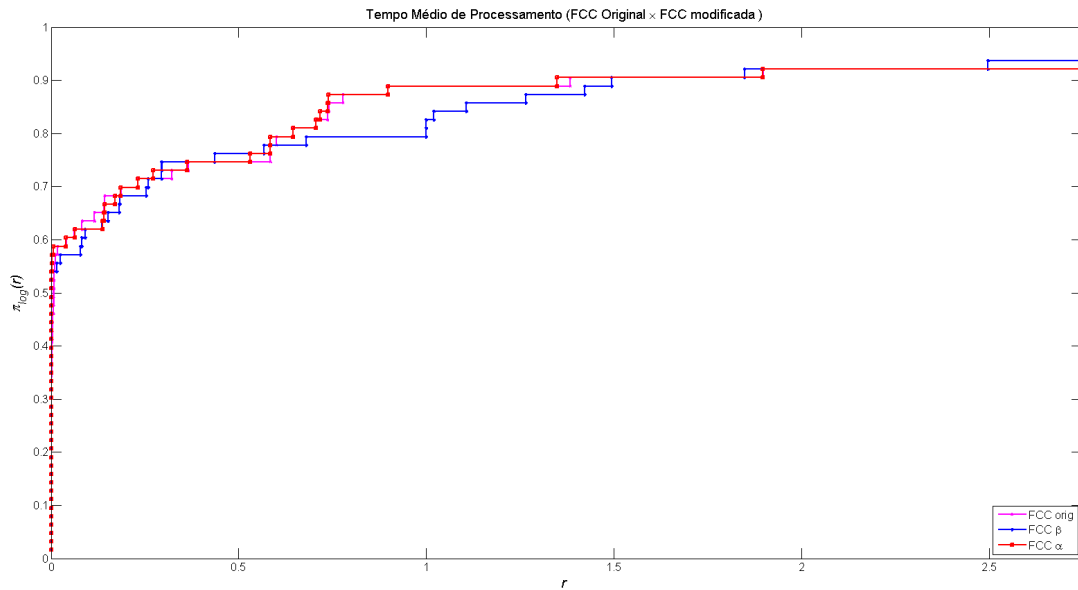


Figura 5.8: A FCC original, FCC α e FCC β possuem perfis de desempenho bem parecidos. Porém, para o fator de tempo $r = 2,5$, a FCC β tem uma probabilidade levemente maior de resolver mais problemas.

Os resultados obtidos pela FCC α mostram que a nova estratégia para o cálculo do parâmetro de correção α melhora o desempenho da fatoração controlada de Cholesky no preconditionamento de sistemas lineares oriundos de métodos de pontos interiores. A técnica também se mostrou adequada para a construção de preconditionadores atualizados como a FCC β .

Capítulo 6

Conclusões e Trabalhos Futuros

6.1 Conclusões

Duas abordagens foram propostas para reduzir o tempo computacional necessário para a resolução de problemas de programação linear, via método de pontos interiores, mediante modificações no preconditionador fatoração controlada de Cholesky. Tais propostas basearam-se no fato de que a fatoração controlada de Cholesky é suscetível à falhas durante o processo de fatoração, devido a presença de elementos não positivos e/ou muito pequenos na diagonal. Quando uma falha ocorre, a diagonal da matriz $A\Theta A^T$ é incrementada de um número α e a fatoração da nova matriz é reiniciada.

As modificações propostas consideraram duas situações. Uma buscou reduzir o número de vezes que a fatoração controlada de Cholesky pode ser reiniciada até concluir a construção do preconditionador e a outra considerou evitar tais reinícios. Para isso, foi considerada uma nova maneira de calcular o parâmetro α . Aceita-se para efetuar a correção o menor valor de α que somado ao pivô deficiente o torna aceitável.

A primeira proposta tenta evitar o reinício da fatoração efetuando escolhas maiores de α para realizar as correções na diagonal. Nessa abordagem, em vez de começar as correções com valores muito pequenos de α , permitiu-se iniciar as correções com um valor de α que tornasse o pivô deficiente aceitável.

Por sua vez, a segunda situação considerou dois tipos de correções. Uma que efetua a correção apenas no pivô no qual ocorreu a falha e continua com o processo de fatoração, e outra que propôs atualizar os elementos das colunas disponíveis do preconditionador fatoração controlada de Cholesky, em vez de realizar uma nova fatoração. Dessa forma, essa abordagem utiliza os elementos disponíveis do fator L para obter aproximações para os elementos de $\tilde{L}$.

Ambas as técnicas tiveram o seu desempenho analisado no preconditionamento de sistemas lineares oriundos de métodos de pontos interiores. Foi adotado o método de preconditionamento híbrido, composto pelo preconditionador fatoração controlada de Cholesky e o preconditionador separador.

Experimentos numéricos foram realizados com vários problemas de programação linear e resultados promissores foram obtidos para ambas as propostas, indicando que o uso das mesmas pode reduzir o tempo de resolução de algumas classes de problemas de programação linear. Em particu-

lar, para problemas da coleção QAPLIB obteve-se evidências de que atualizar o preconditionador fatoração controlada de Cholesky pode ser mais eficiente que calcular uma nova fatoração desde o início.

As modificações propostas também foram combinadas com o algoritmo de ajustamento ótimo para p coordenadas. Experimentos numéricos realizados com problemas da coleção QAPLIB mostraram que a abordagem também melhora a resolução de problemas de programação linear.

6.2 Trabalhos Futuros

A fatoração controlada de Cholesky é um preconditionador universal e, portanto, pode ser aplicado a vários tipos de problemas. Assim, as abordagens apresentadas aqui podem ser testadas em outros contextos de preconditionamento, como por exemplo, a resolução numérica de certos tipos de equações diferenciais.

Na presença de falhas durante a construção do preconditionador fatoração controlada de Cholesky, a substituição do mesmo por um preconditionador obtido com uma combinação das colunas já disponíveis no momento da falha e um preconditionador diagonal pode ser investigada.

Novas fórmulas de atualização do preconditionador fatoração controlada de Cholesky, que mantenham boas propriedades do preconditionador atualizado, mesmo na presença de perturbações maiores na diagonal, poderão ser investigadas.

O desempenho do gradiente conjugado depende da distribuição dos autovalores da matriz do sistema linear. Assim, pode-se considerar usar o espectro de cada matriz preconditionada, como por exemplo, o maior e o menor valor de Ritz, a média e o desvio padrão desses elementos para comparar as versões modificadas entre elas e com a fatoração controlada de Cholesky.

Referências Bibliográficas

- [1] Netlib lp repository: Netlib collection LP test sets. <http://www.netlib.org/lp/data/>. Acessado: 04-04-2014.
- [2] J.O. Aasen. On the reduction of a symmetric matrix to tridiagonal form. *BIT*, 11:233–242, 1971.
- [3] G. Al-Jeiroudi, J. Gondzio, and J. Hall. Preconditioning indefinite systems in interior point methods for large scale linear optimization. *Optimization Methods & Software*, 23:345–363, 2008.
- [4] M.S. Bazaraa, J.J. Jarvis, and H.D. Sherali. *Linear Programming and Network Flows*. Wiley, 2010.
- [5] M.S. Bazaraa, H.D. Sherali, and C.M. Shetty. *Nonlinear Programming: Theory and Algorithms*. Wiley, 2013.
- [6] S. Bellavia, V. de Simone, D. di Serafino, and B. Morini. Efficient preconditioner updates for shifted linear systems. *SIAM J. Sci. Comput.*, 33(4):1785 – 1809, 2011.
- [7] S. Bellavia, V. de Simone, D. di Serafino, and B. Morini. A preconditioning framework for sequences of diagonally modified linear systems arising in optimization. *SIAM J. Numerical Analysis*, 50(6):3280–3302, 2012.
- [8] M. Benzi. Preconditioning techniques for large linear systems: A survey. *Journal of Computational Physics*, 182(2):418 – 477, 2002.
- [9] M. Benzi, G.H. Golub, and J. Liesen. Numerical solution of saddle point problems. *Acta Numerica*, 14:1–137, 2005.
- [10] L. Bergamaschi, J. Gondzio, M. Venturin, and G. Zilli. Inexact constraint preconditioners for linear system arising interior point methods. *Computational Optimization and Applications*, 36(3):137–147, 2007.
- [11] S. Bocanegra. *Algoritmos de Newton-Krylov preconditionados para métodos de pontos interiores*. PhD thesis, Instituto de Ciências Exatas - Universidade Federal de Minas Gerais, 2005. <http://hdl.handle.net/1843/RVMR-6JTN72>.

- [12] S. Bocanegra, F.F. Campos, and A.R.L. Oliveira. Using a hybrid preconditioner for solving large-scale linear systems arising from interior point methods. *Computational Optimization and Applications*, 36:149–164, 2007.
- [13] J.R. Bunch and B.N. Parlett. Direct methods for solving symmetric indefinite systems of linear equations. *SIAM J. Numer. Anal.*, 8:639–655, 1971.
- [14] R.E. Burkard, S. Karisch, and F. Rendl. Qaplib - A quadratic assignment problem library. *European Journal of Operational Research*, 55(1):115 – 119, 1991.
- [15] F.F. Campos. *Analysis of conjugate gradients—type methods for solving linear equations*. PhD thesis, Oxford University Computing Laboratory, 1995. http://solo.bodleian.ox.ac.uk/primo_library/libweb/action/dlDisplay.do?vid=OXVU1&docId=oxfaleph015311280.
- [16] F.F. Campos and N.R.C. Birkett. An efficient solver for multi-right hand side linear systems based on the $\text{cccg}(\eta)$ method with applications to implicit time-dependent partial differential equations. *SIAM J. Sci. Comput.*, 19(1):126 – 138, 1998.
- [17] J.S. Chai and Kim-Chuan Toh. Preconditioning and iterative solution of symmetric indefinite linear systems arising from interior point methods for linear programming. *Comp. Opt. and Appl.*, 36(2-3):221–247, 2007.
- [18] J. Czyzyk, S. Mehrotra, and S. J. Wright. PCx user guide, 1997.
- [19] Elizabeth D. Dolan and Jorge J. Moré. Benchmarking optimization software with performance profiles. *Mathematical Programming*, 91(2):201–213, 2002.
- [20] I.S. Duff and G.A. Meurant. The effect of ordering on preconditioned conjugate gradients. *BIT*, 29(4):635–657, 1989.
- [21] C.T.L.S Ghidini, A.R.L. Oliveira, J. Silva, and M.I. Velazco. Combining a hybrid preconditioner and a optimal adjustment algorithm to accelerate the convergence of interior point methods. *Linear Algebra and Applications*, 436(5):1267–1284, 2012.
- [22] P.E. Gill, W. Murray, D.B. Ponceleón, and M.A. Saunders. Preconditioners for indefinite systems arising in optimization. *SIAM J. Matrix Anal. Appl.*, 13(1):292–311, 1992.
- [23] G.H. Golub and F. Van Loan. *Matrix Computation*. The Johns Hopkins University Press, 3 edition, 1996.
- [24] J. Gondzio. Interior point methods 25 years later. *European Journal of Operational Research*, 218(3):587 – 601, 2012.
- [25] M.T. Jones and P. E. Plassmann. An improved incomplete Cholesky factorization. *ACM Transactions on Mathematical Software*, 21(1):5–17, 1995.
- [26] N. Karmarkar. A new polynomial-time algorithm for linear programming. *Combinatorica*, 4(4):373–395, 1984.

- [27] D.S. Kershaw. The incomplete Cholesky—conjugate gradient method for the iterative solution of systems of linear equations. *Journal of Computational Physics*, 26(1):43 – 65, 1978.
- [28] L.G. Khachiyan. Polynomial algorithms in linear programming. *USSR Computational Mathematics and Mathematical Physics*, 20(1):53 – 72, 1980.
- [29] C-Jen Lin and J.J. Moré. Incomplete Cholesky factorizations with limited memory. *SIAM J. SCI. COMPUT.*, 21(1):24–45, 1999.
- [30] D. Luenberger and Y. Ye. *Linear and Nonlinear Programming*. Springer, 3 edition, 2008.
- [31] T.A. Manteuffel. An incomplete factorization technique for positive definite linear systems. *Mathematics of Computation*, 34(150):473–497, 1980.
- [32] S. Mehrotra. On the implementation of a primal-dual interior point method. *SIAM Journal on Optimization*, 2(4):575–601, 1992.
- [33] J.A. Meijerink and H.A. van der Vorst. An iterative solution method for linear systems of which the coefficient matrix is a symmetric m -matrix. *Mathematics of Computation*, 31(137):148–162, 1977.
- [34] G. Meurant. On the incomplete Cholesky decomposition of a class of perturbed matrices. *SIAM J. Sci. Comput.*, pages 419–429, 2001.
- [35] N. Munksgaard. Solving sparse symmetric sets of linear equations by preconditioned conjugate gradients. *ACM Trans. Math. Softw.*, 6(2):206–219, 1980.
- [36] J. Nocedal and S.J. Wright. *Numerical Optimization*. Springer, 2 edition, 2006.
- [37] A.R.L. Oliveira. *A New Class of Preconditioners for Large-Scale Linear Systems from Interior Point Methods for Linear Programming*. PhD thesis, Department of Computational and Applied Mathematics, Rice University, 1997. <http://hdl.handle.net/1911/19149>.
- [38] A.R.L. Oliveira and D.C. Sorensen. A new class of preconditioners for large-scale linear systems from interior point methods for linear programming. *Linear Algebra and its Applications*, 394(0):1 – 24, 2005.
- [39] Y. Saad and H.A. van der Vorst. Iterative solution of linear systems in the 20th century. *Journal of Computational and Applied Mathematics*, 123(1–2):1 – 33, 2000.
- [40] J. Silva. *Uma família de algoritmos para programação linear baseada no algoritmo de Von Neumann*. PhD thesis, Instituto de Matemática, Estatística e Ciências da Computação-Universidade de Campinas, 2009. <http://www.bibliotecadigital.unicamp.br/document/?code=000443131&fd=y>.
- [41] L.N. Trefethen and D.I. Bau. *Numerical Linear Algebra*. Society for Industrial and Applied Mathematics, 1997.

- [42] R.J. Vanderbei. *Linear Programming: Foundations and Extensions*. International Series in Operations Research & Management Science. Springer, 1996.
- [43] M.I. Velazco, A. R.L. Oliveira, and F.F. Campos. A note on hybrid preconditioners for large-scale normal equations arising from interior-point methods. *Optimization Methods & Software*, 25(2):321–332, 2010.
- [44] S.J. Wright. *Primal-Dual Interior-Point Methods*. Miscellaneous Bks. Society for Industrial and Applied Mathematics, 1997.