



UNIVERSIDADE ESTADUAL DE
CAMPINAS

Instituto de Matemática, Estatística e
Computação Científica

FÁBIO RODRIGUES SILVA

**Experimentos numéricos com um sistema linear
alternativo para o pré-condicionador separador
aplicado a métodos de pontos interiores**

Campinas

2016

Fábio Rodrigues Silva

**Experimentos numéricos com um sistema linear
alternativo para o pré-condicionador separador aplicado a
métodos de pontos interiores**

Dissertação 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 Mestre em Matemática Aplicada.

Orientador: Aurelio Ribeiro Leite de Oliveira

Este exemplar corresponde à versão final da Dissertação defendida pelo aluno Fábio Rodrigues Silva e orientada pelo Prof. Dr. Aurelio Ribeiro Leite de Oliveira.

Campinas

2016

Agência(s) de fomento e nº(s) de processo(s): CAPES

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

Si38e Silva, Fábio Rodrigues, 1990-
Experimentos numéricos com um sistema linear alternativo para o pré-condicionador separador aplicado a métodos de pontos interiores / Fábio Rodrigues Silva. – Campinas, SP : [s.n.], 2016.

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

1. Programação linear. 2. Métodos de pontos interiores. 3. Pré-condicionadores. 4. Sistemas lineares. 5. Métodos do gradiente conjugado. 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: Numerical experiments with an alternative linear system for the splitting preconditioner applied to interior point methods

Palavras-chave em inglês:

Linear programming

Interior-point methods

Preconditioners

Linear systems

Conjugate gradient methods

Área de concentração: Matemática Aplicada

Titulação: Mestre em Matemática Aplicada

Banca examinadora:

Aurelio Ribeiro Leite de Oliveira [Orientador]

Daniela Renata Cantane

Marta Ines Velazco Fontova

Data de defesa: 29-02-2016

Programa de Pós-Graduação: Matemática Aplicada

Dissertação de Mestrado defendida em 29 de fevereiro de 2016 e aprovada

Pela Banca Examinadora composta pelos Profs. Drs.

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

Prof.(a). Dr(a). DANIELA RENATA CANTANE

Prof.(a). Dr(a). MARTA INES VELAZCO FONTOVA

A Ata da defesa com as respectivas assinaturas dos membros encontra-se no processo de vida acadêmica do aluno.

À minha família...

Agradecimentos

Ao professor Dr. Aurelio Oliveira, pela enorme paciência e atenção e imprescindível orientação no desenvolvimento deste trabalho.

À professora Dra. Marta Velazco, pela valiosa contribuição na solução dos entraves apresentados pelo código computacional.

Aos amigos Maria Cristina Tiemi, Paula Kikuchi, Daniela Oliveira, Kelly Cadena, Ariquele Rios, Elisama Araujo, Jônathas Douglas, Petra Bartmeyer, José Vanterler, Fernanda Bia, Lilian Berti, Diego Gomes, Ana Carolina Vieira, Chryslaine Pereira, Tatiana Rocha, Luiz Sobral e tantos outros que conviveram e convivem comigo no instituto e fora dele, proporcionando momentos de alegria, aprendizado e muita descontração, que são tão importantes para superar as flutuações de ânimo às quais enfrentamos dia após dia.

Às amigas que tenho de Ituiutaba-MG, que promovem os melhores momentos de diversão, em especial ao Adriano Lemos, Thaís Carvalho, Juliana Araújo, Ricardo Oliveira, Lorena Ribeiro, Antunes Neto, Natasha Pim, Isabella Muniz e muitos outros. O carinho que vocês têm por mim é aconchegante e sempre me sinto bem em estar com vocês!

À minha irmã Fernanda Anielly Silva, nada consegue ter sentido sem você! Aos sobrinhos Gustavo e Gabriela, que são duas dádivas maravilhosas, amores da minha vida, razão das lágrimas de saudade de casa e que sempre levarei à frente de meus planos.

Agradeço aos meus pais Francisco Rodrigues da Silva e Maria Aparecida Vitor da Conceição, por todo o apoio em minhas decisões, suporte emocional nos momentos de necessidade e por continuar acreditando em minhas habilidades e competências, me incentivando a sempre ir cada vez mais longe. É por vocês que eu sonho alto.

Aos professores da FACIP-UFU, que sempre acreditaram no meu potencial e incentivaram a alçar voos mais altos na formação acadêmica, em especial à Professora Dra. Tânia Carvalho, eterna mãezona do PET Matemática Pontal, amiga e conselheira. Agradeço, também, aos professores João Carlos Moreira, José Laércio Doricio e Milena Brandão pela amizade e experiência acadêmica.

Aos irmãos do PET Matemática Pontal: a família é sempre unida até mesmo quando não estamos próximos. Agradecimento especial à Jeane Silveira, Enio Marques, Bruna Luiza, Angélica Lourenço, José Henrique e Magna Paulina. A gente sofre mas a gente se diverte.

À Dirce Alves, que além de me acolher em sua residência, me dá preciosos conselhos sobre a vida de modo geral.

À Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) pelo suporte financeiro desta pesquisa, junto ao Programa de Iniciação Científica e Mestrado (PICME) da Olimpíada Brasileira de Matemática das Escolas Públicas (OBMEP). Poderia não ter conseguido chegar tão longe se não fosse a OBMEP e o PICME.

A todos que direta ou indiretamente contribuíram para o desenvolvimento desta dissertação.

A Deus, por TUDO!

“Não basta ensinar ao homem uma especialidade. Porque se tornará assim uma máquina utilizável, mas não uma personalidade. É necessário que se adquira um senso prático daquilo que vale a pena ser empreendido, daquilo que é belo, do que é moralmente correto. A não ser assim, ele se assemelhará, com seus conhecimentos profissionais, mais a um cão ensinado do que uma criatura harmoniosamente desenvolvida. Deve aprender a compreender as motivações dos homens, suas quimeras e suas angústias para determinar com exatidão seu lugar exato em relação a seus próximos e à comunidade. Os excessos do sistema de competição e de especialização prematura, sob o falacioso pretexto da eficácia, assassinam o espírito, impossibilitam qualquer vida cultural e chegam a suprimir os progressos nas ciências do futuro. É preciso, enfim, tendo em vista a realização de uma educação perfeita, desenvolver o espírito crítico na inteligência do jovem.” (Einstein)

Resumo

Neste trabalho, realizamos experimentos numéricos com um sistema linear alternativo para o pré-condicionador separador aplicado ao método iterativo de Gradientes Conjugados pré-condicionado, no contexto da solução iterativa de problemas de programação linear via método de pontos interiores. Consideramos o método preditor-corretor proposto por Mehrotra, que obtém a solução do problema de programação linear aplicando o método de Newton nas condições de otimalidade de Karunsh-Kuhn-Tucker perturbadas. O método de Gradientes Conjugados pré-condicionado é usado para obter a solução dos sistemas lineares. Em trabalhos anteriores, uma estratégia híbrida de pré-condicionamento foi proposta e experimentos computacionais revelaram que seu desempenho é superior ao uso de métodos diretos na solução dos sistemas lineares para algumas classes de problemas de programação linear. Seu funcionamento ocorre em duas fases: a fase 1 utiliza um tipo de pré-condicionador de fatoração incompleta de Cholesky com preenchimento adaptativo em função da memória disponível no computador; a fase 2 utiliza um pré-condicionador baseado na fatoração LU, especializado para as últimas iterações dos métodos de pontos interiores. Nestes trabalhos, heurísticas para promover a troca de fases foram apresentadas e novas estratégias para o reordenamento das colunas da matriz de coeficientes foram exibidas. Na fase 2, a matriz pré-condicionada é indefinida e pode ser reduzida para um sistema linear definido positivo de uma das dimensões do complemento de Schur. Estes trabalhos basearam-se em resultados numéricos sobre a matriz quadrada de dimensão igual ao número de linhas da matriz de restrições. Realizamos experimentos computacionais com a matriz quadrada de dimensão igual à diferença entre o número de colunas e linhas da matriz de restrições, objetivando aferir o desempenho das versões desenvolvidas para este fim. Foram propostas quatro estratégias de recuperação da solução do sistema aumentado pré-condicionado. Exibimos os resultados em função de quatro métricas de desempenho: o número de iterações de pontos interiores e de gradientes conjugados, tempo total de solução e número de vezes que a fase 2 calcula uma nova matriz não singular. Comparamos as versões propostas com a versão híbrida apresentada em trabalhos anteriores, resolvendo um conjunto de 35 problemas teste selecionados de diferentes bibliotecas de problemas de programação matemática. Apenas uma das versões desenvolvidas neste trabalho resolveu todos problemas propostos. Os resultados mostram que, de maneira geral, as versões desenvolvidas apresentam melhor comportamento computacional em relação à versão híbrida, no que se refere às métricas de desempenho consideradas, para todos os problemas que foram resolvidos com sucesso.

Palavras-chave: Programação linear. Métodos de pontos interiores. Pré-condicionadores. Sistemas lineares. Métodos do gradiente conjugado.

Abstract

In this work, we carried out numerical experiments with alternative linear systems for the splitting preconditioner applied to preconditioned Conjugate Gradient iterative method, in the context of iterative solution of linear programming problems by interior point methods. We consider the Mehrotra's predictor-corrector method, that search the solution of the linear programming problem by applying Newton's method in the perturbed Karunsh-Kuhn-Tucker's optimality conditions. The preconditioned Conjugate Gradients method is used for the solution of the linear systems. In previous works, a hybrid preconditioning strategy was proposed and computational experiments show its superior performance against direct methods in solving linear systems for some linear programming problems classes. Such approach takes place in two phases: phase 1 uses a kind of incomplete Cholesky preconditioner with an adaptive fill-in as a function of the available memory in the computer; phase 2 uses a preconditioner based on LU factorization, specialized for the latest iterations of interior point methods. In these works, heuristics to promote the exchange of phases were presented and new strategies for the reordering of the columns of the coefficient matrix were presented. In phase 2, the preconditioned matrix is indefinite and can be reduced to a positive definite linear system of one of the sizes of Schur complement. These studies were based on numerical results on the square matrix of dimension equals to the number of constraints matrix lines. We perform computational experiments with a square matrix of size equals to the difference between the number of columns and rows of constraints matrix, aiming to benchmark the versions developed for this purpose. It was proposed four strategies for the preconditioned augmented system recovery solution. We show the results in terms of four performance metrics: the number of interior point and conjugate gradient iterations, solution time and number of times that the phase 2 computes a new non-singular matrix. We compare the proposed versions with the hybrid version presented in previous works, solving a set of 35 selected test problems from different mathematical programming problems libraries. Only one of the developed versions in this work solve all the chosen problems. The results show that, in general, the developed versions present better computational performance over the hybrid version, regarding the performance metrics considered, for all the successful solveds problems.

Keywords: Linear programming. Interior-point methods. Preconditioning. Linear systems. Conjugate gradient methods.

Lista de ilustrações

Figura 1 – Soluções iteradas do Algoritmo Preditor-Corretor no espaço (x, s) . . .	36
Figura 2 – Técnicas adotadas pela versão original do <i>solver</i> PCx	58
Figura 3 – Técnicas adotadas pela versão modificada do <i>solver</i> PCx, obtida pelo autor	59
Figura 4 – Técnicas adotadas pelo <i>solver</i> PCx na versão implementada pelo autor	60
Figura 5 – Perfil de desempenho para iterações do método de pontos interiores . .	95
Figura 6 – Perfil de desempenho para tempo total de solução	96
Figura 7 – Perfil de desempenho para iterações de Gradientes Conjugados pré-condicionado	98
Figura 8 – Perfil de desempenho para cálculos de uma nova matriz B	99

Lista de tabelas

Tabela 1	–	Dados gerais dos problemas abordados.	85
Tabela 2	–	Resultados gerais para as versões PCx_MOD, PCx_F3 e PCx_F4. . .	88
Tabela 3	–	Resultados gerais para as versões PCx_MOD, PCx_A3 e PCx_A4. . .	89
Tabela 4	–	Total de iterações do método de Gradientes Conjugados pré-condicionado para as versões PCx_MOD, PCx_F3 e PCx_F4.	90
Tabela 5	–	Total de iterações do método de Gradientes Conjugados pré-condicionado para as versões PCx_MOD, PCx_A3 e PCx_A4.	91
Tabela 6	–	Dados de atualização da matriz B após a mudança de pré-condicionadores para as versões PCx_MOD, PCx_F3 e PCx_F4.	92
Tabela 7	–	Dados de atualização da matriz B após a mudança de pré-condicionadores para as versões PCx_MOD, PCx_A3 e PCx_A4.	93

Lista de símbolos

$\mathbf{M}_{m \times n}(\mathbb{R})$	Espaço vetorial das matrizes de ordem $m \times n$ com entradas reais
\mathbb{R}_+^n	Conjunto dos vetores x pertencentes a \mathbb{R}^n que satisfazem $x \geq 0$
\mathbb{R}_{++}^n	Conjunto dos vetores x pertencentes a \mathbb{R}^n que satisfazem $x > 0$
$\ \cdot\ _p$	Norma-p vetorial
$\ \cdot\ _p$	Norma-p matricial
$\ \cdot\ _F$	Norma matricial de Frobenius
\mathbf{I}_n	Matriz identidade de ordem $n \times n$
$\text{tr}(\mathbf{A})$	Traço da matriz \mathbf{A}

Lista de algoritmos

Algoritmo 1 – Método de pontos interiores primal-dual	34
Algoritmo 2 – Método Preditor-Corretor de Mizuno-Todd-Ye	37
Algoritmo 3 – Método Preditor-Corretor de Mehrotra	40
Algoritmo 4 – Método da Máxima Descida	111
Algoritmo 5 – Método dos Gradientes Conjugados	112
Algoritmo 6 – Método dos Gradientes Conjugados Pré-condicionado	113

Sumário

Introdução	17
1 Programação Linear	20
1.1 Conceitos preliminares	20
1.2 Convexidade	24
1.3 Dualidade e condições de otimalidade	25
1.4 Método de pontos interiores primais-duais	32
1.4.1 Métodos seguidores de caminho: o método Preditor-Corretor	33
1.4.2 Ponto inicial infactível: o método Preditor-Corretor de Mehrotra	37
1.4.2.1 Uma variante do algoritmo de Mehrotra	39
1.4.3 Calculando as direções de busca	39
2 Sistema Aumentado	41
2.1 Preliminares	41
2.2 Precondicionamento de sistemas lineares	42
2.3 Propriedades do sistema aumentado	44
2.4 Pré-condicionadores no contexto do sistema aumentado e do complemento de Schur	46
2.5 Uma classe de pré-condicionadores para o método de pontos interiores	48
2.5.1 Construção de um pré-condicionador	48
2.5.2 Propriedades do sistema pré-condicionado	51
2.5.3 Redução a sistemas definidos positivos	52
2.5.4 Escolhendo o conjunto de colunas	55
3 Aspectos Práticos	58
3.1 Fatoração Incompleta de Cholesky	60
3.1.1 Fatoração Controlada de Cholesky	62
3.2 Mudança de Fases	64
3.3 Considerações sobre o algoritmo utilizado	64
3.3.1 Formulação adotada	65
3.3.2 A técnica de pré-processamento	67
3.3.3 Técnica de <i>scaling</i>	68
3.3.4 Escolha do ponto inicial	69
3.3.5 Critério de parada	70
3.4 Considerações sobre métodos iterativos	72
3.4.1 Recuperação da solução: estratégia inicial	72
3.4.2 Recuperação da solução: primeira estratégia nova	74
3.4.3 Recuperação da solução: segunda estratégia nova	77
3.4.4 Recuperação da solução: terceira estratégia nova	80

3.4.5	Recuperação da solução: quarta estratégia nova	81
3.5	Considerações sobre a fatoração LU	81
3.5.1	Obtendo colunas linearmente independentes	81
3.5.2	Dependência e independência simbólica	82
4	Experimentos Computacionais e Análise dos Resultados	84
4.1	Apresentação dos resultados	86
4.2	Análise dos dados	94
5	Considerações Finais	101
 REFERÊNCIAS		103
 Apêndices		107
APÊNDICE A Método de Newton		108
APÊNDICE B Método dos Gradientes Conjugados Pré-condicionado		110
B.1	Método da Máxima Descida	110
B.2	Método dos Gradientes Conjugados	111
B.3	Método dos Gradientes Conjugados aplicado a um sistema linear pré- condicionado	112

Introdução

A programação linear é um campo de destaque da teoria de otimização desde os anos 1940. Muitas técnicas da teoria de otimização recaem em formulações lineares para os problemas de aplicações práticas oriundas dos diversos setores da sociedade, tais como Militar, Administração, Engenharias, Economia e Finanças, Transporte, Produção, dentre muitas outras. As formulações e métodos de solução tiveram início formal a partir da Segunda Guerra Mundial,

[...] quando então a estatística era utilizada para determinar quais as táticas militares que funcionavam. A lógica pragmática das operações militares é que especificava as agendas de pesquisas, e nesse contexto o que importava era a eficácia do método, pois as soluções tinham de ser alcançadas em tempo hábil. (CHAVES, 2011, p. 12).

Um dos precursores desta linha, e cujo algoritmo desenvolvido ainda é largamente aplicado à solução dos problemas de programação linear, é George B. Dantzig. Seu algoritmo, denominado Algoritmo Simplex, foi desenvolvido em 1947 e, embora seja bastante eficiente, confiável e largamente aplicado, é um algoritmo de tempo não polinomial, sendo de tempo exponencial no pior caso (um algoritmo é dito ser de tempo polinomial quando existe um polinômio relacionado à dimensão do problema a ser resolvido e que limita o número de operações necessárias para a resolver o problema). (KLEE; MINTY, 1972; DANTZIG; THAPA, 1997).

De acordo com Gondzio (2012, p. 587), o primeiro algoritmo de tempo polinomial para programação linear foi desenvolvido por Khachiyan (1980), que constrói uma série de elipsoides inscritos na região factível do problema de programação linear e os centros desses elipsoides formam uma sequência de pontos que convergem para uma solução ótima do problema. Anos depois, Karmarkar (1984) propôs um algoritmo projetivo para resolver o problema de programação linear em tempo polinomial, o qual pode ser visto como um refinamento do método proposto por Khachiyan. Em seu trabalho, Karmarkar introduziu a noção de uma função potencial para garantir uma constante redução da distância à otimalidade a cada iteração. Embora cada iteração do método de Karmarkar seja computacionalmente cara, tendo em vista a requisição de uma operação de projeção que muda a cada iteração, ele atinge a otimalidade em poucas iterações, característica que tornou seu método bastante atrativo. Estes são considerados os precursores dos métodos de pontos interiores. Tais métodos buscam a otimalidade do problema por construir uma série de pontos que percorrem o interior da região factível, diferente do que acontece no método Simplex onde a sequência de pontos é tomada nos vértices da região.

Diversas contribuições foram feitas na teoria de pontos interiores, tais como o uso de funções de barreira logarítmica, a interpretação de que os métodos seguem um caminho

central que converge para a solução do problema, o uso de pontos iniciais e estratégias infactíveis, entre outros estudos. No início dos anos 1990, já haviam evidências suficientes que comprovassem o desempenho superior dos métodos de pontos interiores em programação linear de grande porte. [Mehrotra \(1992\)](#) propôs um algoritmo de pontos interiores que teve destaque nos anos seguintes, devido às suas propriedades computacionais atraentes. Seu algoritmo resolve dois sistemas lineares com a mesma matriz de coeficientes para construir uma direção de busca, além de usar uma heurística para estimar o comprimento de passo, de tal modo que a distância à otimalidade seja reduzida a cada iteração, mas com convergência quadrática. O sucesso de sua implementação atraiu diversos pesquisadores a desenvolver técnicas que melhorem o desempenho computacional do método, de modo a obter menores custo e tempo computacional a cada iteração ([GONDZIO, 2012](#)).

Os sistemas lineares envolvidos no algoritmo de Mehrotra podem ser resolvidos por abordagens diretas ou iterativas. [Oliveira e Sorensen \(2005\)](#) estudaram a solução iterativa dos sistemas lineares utilizando o método iterativo de Gradientes Conjugados, pré-condicionado por um tipo de matriz que pertence a uma classe desenvolvida para ser especializada no tratamento destes sistemas lineares, que ficam cada vez mais mal condicionados à medida em que a solução do problema de programação linear se aproxima, mostrando ser uma abordagem melhor que a fatoração de Cholesky (uma das fatorações mais usadas atualmente para abordagem direta). Eles mostraram, ainda, que dos sistemas lineares do algoritmo de Mehrotra (que podem ser reduzidos ao sistema aumentado e, ainda, ao complemento de Schur), o sistema do complemento de Schur pré-condicionado pode ser visto como um caso particular contido em uma classe mais geral de pré-condicionadores para o sistema aumentado, fato este que justificou a construção de uma classe de pré-condicionadores para o sistema aumentado. Além disso, como a matriz pré-condicionada é diagonal por blocos eles mostraram que, na estratégia de recuperação de solução, um dos blocos da matriz pré-condicionada pode ser reduzido a um sistema definido positivo quadrado, de dimensão igual ao número de linhas (ou à diferença entre o número de colunas e linhas) da matriz de restrições do problema de programação linear. Uma destas matrizes definidas positivas (especificamente a de dimensão igual ao número de linhas da matriz de restrições) pode ser obtida por um pré-condicionador para o complemento de Schur.

[Bocanegra, Campos e Oliveira \(2007\)](#) propuseram uma abordagem híbrida de pré-condicionamento dos sistemas lineares oriundos do método de pontos interiores proposto por Mehrotra. A estratégia desenvolvida considera que o processo de solução acontece em duas fases, sendo que na fase 1 é adotado um tipo de pré-condicionador de fatoração incompleta, enquanto que a fase 2 adota o pré-condicionador proposto por [Oliveira e Sorensen \(2005\)](#). Foi apresentada, ainda, uma heurística que promove a troca de pré-condicionadores, heurística esta que foi melhorada no trabalho de [Velazco, Oliveira e Campos \(2010\)](#).

A proposta central desta dissertação é realizar experimentos numéricos com o sistema linear definido positivo de dimensão igual à diferença entre o número de colunas e linhas da matriz de restrições, oriundo da redução a sistemas definidos positivos descrita por [Oliveira e Sorensen \(2005, p. 9\)](#). Os trabalhos de [Oliveira e Sorensen \(2005\)](#), de [Bocanegra, Campos e Oliveira \(2007\)](#) e de [Velazco, Oliveira e Campos \(2010\)](#) basearam-se em resultados numéricos sobre a matriz quadrada de dimensão igual ao número de linhas da matriz de restrições. Isto justifica o interesse pela realização dos experimentos com um sistema linear alternativo.

A seguir, esta dissertação está organizada em 5 capítulos. No primeiro, podemos encontrar uma revisão dos conceitos e principais resultados de programação linear que oferecem suporte ao assunto central desta dissertação. No segundo capítulo, apresentamos as características do sistema aumentado e descrevemos a classe apresentada por [Oliveira e Sorensen \(2005\)](#). O terceiro capítulo traz questões práticas acerca do código utilizado e estratégias de recuperação da solução do sistema linear pré-condicionado. O quarto capítulo apresenta os resultados computacionais obtidos das implementações e o quinto capítulo exprime as considerações finais deste trabalho. Adicionalmente, 2 apêndices foram escritos, de modo a oferecer mais apoio teórico aos métodos mencionados no decorrer do texto.

1 Programação Linear

A área de abrangência desta dissertação é a Programação Matemática, especificamente a Programação Linear. Em [Dantzig e Thapa \(1997, p. 1\)](#), Programação Matemática é definida como sendo um ramo da matemática, que trata de técnicas para maximizar ou minimizar uma função objetivo, sujeito a restrições lineares, não lineares e/ou inteiras sobre as variáveis. Abaixo, podemos ver uma das formas que se apresentam problemas de programação matemática

$$\begin{aligned} &\text{minimizar} && f(x) \\ &\text{sujeito a} && x \in \mathcal{T}, \end{aligned} \tag{1.1}$$

onde x é denominada *variável de decisão*, f é a *função objetivo* e \mathcal{T} representa o conjunto de *restrições* do problema. Desta forma, em Programação Linear, a função objetivo é linear nas variáveis e as restrições podem ser igualdades ou desigualdades lineares.

Neste capítulo, abordamos alguns dos conceitos básicos em programação linear que oferecem suporte ao assunto central desta dissertação. Apresentamos os principais resultados da teoria de dualidade, de modo a relacionar as formulações primal e dual de problemas de programação linear. Exibimos as condições de otimalidade, que devem ser satisfeitas para garantir existência de solução. Finalmente, um método de pontos interiores primal-dual é apresentado para resolver problemas de programação linear.

1.1 Conceitos preliminares

Reescreveremos o problema (1.1) usando restrições e função objetivo lineares. O espaço vetorial que temos interesse é o dos números reais (\mathbb{R}) e, por questões práticas de modelagem, impomos que as variáveis sejam não nulas. Sendo assim, usamos a notação matricial e escrevemos

$$\begin{aligned} &\min && c^\top x \\ &\text{s. a} && x \in \mathcal{X}, \end{aligned} \tag{1.2}$$

onde $\mathcal{X} := \{x \in \mathbb{R}^n; \mathbf{A}x \geq b, x \geq 0\}$, $c \in \mathbb{R}^n$, $\mathbf{A} \in \mathbb{M}_{m \times n}(\mathbb{R})$, $b \in \mathbb{R}^m$, $n > m$ e $\text{posto}(\mathbf{A}) = m$. A menos da imposição $x \in \mathbb{R}_+^n$, a matriz \mathbf{A} possui toda a informação do conjunto de restrições lineares do problema. [Roos, Terlaky e Vial \(2005, p. 18\)](#) definem a forma dada em (1.2) como sendo a *forma canônica* do problema de programação linear.

Exemplo 1.1. *Considere o problema de programação linear a seguir*

$$\begin{aligned} &\max && c^\top x \\ &\text{s. a} && \mathbf{A}x \leq b \\ &&& x \geq 0. \end{aligned} \tag{1.3}$$

Esta formulação é diferente da formulação canônica. Para transformá-lo para a forma canônica usamos o fato que $\max c^\top x = \min -c^\top x$. Desta forma, reescrevemos o problema (1.3) como

$$\begin{aligned} \min \quad & -c^\top x \\ \text{s. a} \quad & -\mathbf{A}x \geq -b \\ & x \geq 0. \end{aligned}$$

Podemos representar o problema de programação linear (1.2) em sua forma padrão que é dada por

$$\begin{aligned} \min \quad & c^\top x \\ \text{s. a} \quad & \mathbf{A}x = b \\ & x \geq 0, \end{aligned} \tag{1.4}$$

que difere de (1.2) apenas na igualdade, em vez de desigualdade no conjunto de restrições.

Exemplo 1.2. Considere o problema de programação linear dado abaixo, onde

$\mathbf{A}_1, \mathbf{A}_2, \mathbf{A}_3 \in \mathbb{M}_{m \times n}(\mathbb{R})$, $x_1, x_2, x_3 \in \mathbb{R}^n$ e $b_1, b_2, b_3 \in \mathbb{R}^m$:

$$\begin{aligned} \min \quad & c_1^\top x_1 + c_2^\top x_2 + c_3^\top x_3 \\ \text{s. a} \quad & \mathbf{A}_1 x_1 \leq b_1 \\ & \mathbf{A}_2 x_2 \geq b_2 \\ & \mathbf{A}_3 x_3 = b_3 \\ & x_1, x_2, x_3 \geq 0. \end{aligned}$$

Vamos transformá-lo em sua forma padrão adicionando à formulação variáveis de folga e excesso não negativas $y_1, y_2 \in \mathbb{R}^m$ para atingir as igualdades desejadas. Então, temos

$$\begin{aligned} \min \quad & c_1^\top x_1 + c_2^\top x_2 + c_3^\top x_3 + 0^\top y_1 + 0^\top y_2 \\ \text{s. a} \quad & \mathbf{A}_1 x_1 + \mathbf{I} y_1 = b_1 \\ & \mathbf{A}_2 x_2 - \mathbf{I} y_2 = b_2 \\ & \mathbf{A}_3 x_3 = b_3 \\ & x_1, x_2, x_3 \geq 0 \\ & y_1, y_2 \geq 0, \end{aligned}$$

que é equivalente a (1.4) tomando $c = [c_1^\top, c_2^\top, c_3^\top, 0^\top, 0^\top]^\top \in \mathbb{R}^{3n+2m}$,

$b = [b_1^\top, b_2^\top, b_3^\top]^\top \in \mathbb{R}^{3m}$, $x = [x_1^\top, x_2^\top, x_3^\top, y_1^\top, y_2^\top]^\top \in \mathbb{R}_+^{3n+2m}$ e

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_1 & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_2 & \mathbf{0} & \mathbf{0} & -\mathbf{I} \\ \mathbf{0} & \mathbf{0} & \mathbf{A}_3 & \mathbf{0} & \mathbf{0} \end{bmatrix} \in \mathbb{M}_{3m \times (3n+2m)}(\mathbb{R}).$$

Podemos encontrar problemas onde o vetor de variáveis de decisão em (1.4) é limitado inferior e superiormente, digamos $l \leq x \leq u$, onde $l, u \in \mathbb{R}^n$. Subtraindo l em cada termo das desigualdades anteriores, teremos $0 \leq \tilde{x} \leq \tilde{u}$ onde $x = \tilde{x} + l$ e $u = \tilde{u} + l$. Nesse

caso, $\mathbf{A}x = b$ se torna $\mathbf{A}\tilde{x} = \tilde{b}$, onde $\tilde{b} = b - \mathbf{A}l$ e $c^\top x$ se torna $c^\top \tilde{x} + c^\top l$. O problema pode, então, ser reescrito como

$$\begin{aligned} \min \quad & c^\top (\tilde{x} + l) \\ \text{s. a} \quad & \mathbf{A}\tilde{x} = \tilde{b} \\ & 0 \leq \tilde{x} \leq \tilde{u}. \end{aligned}$$

Exemplo 1.3. Considere o problema de programação linear dado abaixo, onde $\mathbf{A}_1, \mathbf{A}_2, \mathbf{A}_3 \in \mathbb{M}_{m \times n}(\mathbb{R})$, $x_1, x_2, x_3, u_1, u_2, u_3 \in \mathbb{R}^n$ e $b_1, b_2, b_3 \in \mathbb{R}^m$:

$$\begin{aligned} \min \quad & c_1^\top x_1 + c_2^\top x_2 + c_3^\top x_3 \\ \text{s. a} \quad & \mathbf{A}_1 x_1 \leq b_1 \\ & \mathbf{A}_2 x_2 \geq b_2 \\ & \mathbf{A}_3 x_3 = b_3 \\ & 0 \leq x_1 \leq u_1 \\ & 0 \leq x_2 \leq u_2 \\ & 0 \leq x_3 \leq u_3. \end{aligned}$$

Vamos transformá-lo em sua forma padrão. Cada uma das restrições do tipo $0 \leq x \leq u$ pode ser vista como $\mathbf{I}x \leq u$ e $x \geq 0$, que pode ser reescrita como $\mathbf{I}x + \mathbf{I}w = u$, onde $x, w \in \mathbb{R}_+^n$. Portanto, teremos

$$\begin{aligned} \min \quad & c_1^\top x_1 + c_2^\top x_2 + c_3^\top x_3 + 0^\top y_1 + 0^\top y_2 + 0^\top w_1 + 0^\top w_2 + 0^\top w_3 \\ \text{s. a} \quad & \mathbf{A}_1 x_1 + \mathbf{I}y_1 = b_1 \\ & \mathbf{A}_2 x_2 - \mathbf{I}y_2 = b_2 \\ & \mathbf{A}_3 x_3 = b_3 \\ & \mathbf{I}x_1 + \mathbf{I}w_1 = u_1 \\ & \mathbf{I}x_2 + \mathbf{I}w_2 = u_2 \\ & \mathbf{I}x_3 + \mathbf{I}w_3 = u_3 \\ & x_1, x_2, x_3, w_1, w_2, w_3 \geq 0 \\ & y_1, y_2 \geq 0, \end{aligned}$$

que é equivalente a (1.4) tomando $c = [c_1^\top, c_2^\top, c_3^\top, 0^\top, 0^\top, 0^\top, 0^\top, 0^\top]^\top \in \mathbb{R}^{3n+2m+3n}$, $b = [b_1^\top, b_2^\top, b_3^\top, u_1^\top, u_2^\top, u_3^\top]^\top \in \mathbb{R}^{3m+3n}$, $x = [x_1^\top, x_2^\top, x_3^\top, y_1^\top, y_2^\top, w_1^\top, w_2^\top, w_3^\top]^\top \in \mathbb{R}_+^{3n+2m+3n}$ e

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_1 & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_2 & \mathbf{0} & \mathbf{0} & -\mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{A}_3 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix} \in \mathbb{M}_{(3m+3n) \times (3n+2m+3n)}(\mathbb{R}).$$

Por fim, podemos encontrar problemas onde o vetor de variáveis de decisão em (1.4) é livre (pode assumir valores positivos, negativos ou mistos). Neste caso, uma

alternativa é reescrever o vetor x como $x = u - v$, onde $u, v \in \mathbb{R}_+^n$. O problema pode, então, ser reescrito como

$$\begin{aligned} \min \quad & c^\top u - c^\top v \\ \text{s. a} \quad & \mathbf{A}u - \mathbf{A}v = b \\ & u, v \geq 0. \end{aligned}$$

Exemplo 1.4. Considere o problema de programação linear dado abaixo, onde $\mathbf{A}_1, \mathbf{A}_2, \mathbf{A}_3 \in \mathbb{M}_{m \times n}(\mathbb{R})$, $x_1, x_2, x_3 \in \mathbb{R}^n$ são livres e $b_1, b_2, b_3 \in \mathbb{R}^m$:

$$\begin{aligned} \min \quad & c_1^\top x_1 + c_2^\top x_2 + c_3^\top x_3 \\ \text{s. a} \quad & \mathbf{A}_1 x_1 \leq b_1 \\ & \mathbf{A}_2 x_2 \geq b_2 \\ & \mathbf{A}_3 x_3 = b_3. \end{aligned}$$

Vamos transformá-lo em sua forma padrão. Fazendo $x_i = u_i - v_i$, com $u_i, v_i \geq 0$, $i = 1, 2, 3$, obtemos

$$\begin{aligned} \min \quad & c_1^\top u_1 - c_1^\top v_1 + c_2^\top u_2 - c_2^\top v_2 + c_3^\top u_3 - c_3^\top v_3 + 0^\top y_1 + 0^\top y_2 \\ \text{s. a} \quad & \mathbf{A}_1 u_1 - \mathbf{A}_1 v_1 + \mathbf{I} y_1 = b_1 \\ & \mathbf{A}_2 u_2 - \mathbf{A}_2 v_2 - \mathbf{I} y_2 = b_2 \\ & \mathbf{A}_3 u_3 - \mathbf{A}_3 v_3 = b_3 \\ & u_1, u_2, u_3, v_1, v_2, v_3 \geq 0 \\ & y_1, y_2 \geq 0, \end{aligned}$$

que é equivalente a (1.4) tomando $c = [c_1^\top, -c_1^\top, c_2^\top, -c_2^\top, c_3^\top, -c_3^\top, 0^\top, 0^\top]^\top \in \mathbb{R}^{6n+2m}$,

$b = [b_1^\top, b_2^\top, b_3^\top]^\top \in \mathbb{R}^{3m}$, $x = [u_1^\top, v_1^\top, u_2^\top, v_2^\top, u_3^\top, v_3^\top, y_1^\top, y_2^\top]^\top \in \mathbb{R}_+^{6n+2m}$ e

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_1 & -\mathbf{A}_1 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{A}_2 & -\mathbf{A}_2 & \mathbf{0} & \mathbf{0} & \mathbf{0} & -\mathbf{I} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{A}_3 & -\mathbf{A}_3 & \mathbf{0} & \mathbf{0} \end{bmatrix} \in \mathbb{M}_{(3m) \times (6n+2m)}(\mathbb{R}).$$

Em [Luenberger e Ye \(2008, p. 12-14\)](#) podemos encontrar mais exemplos de como transformar um problema de programação linear para a sua forma padrão, bem como mais algumas questões envolvendo variáveis livres.

Como $\text{posto}(\mathbf{A}) = m$, podemos considerar a seguinte partição $\mathbf{A} = [\mathbf{B} \ \mathbf{N}]$ onde $\mathbf{B} \in \mathbb{M}_m(\mathbb{R})$ é inversível. Agora, suponha que $\bar{x} \in \mathbb{R}^n$ seja uma solução para o sistema linear $\mathbf{A}x = b$. Particionando o vetor \bar{x} de forma equivalente, temos $\bar{x} = [x_B^\top, x_N^\top]^\top$ e podemos reescrever o sistema linear $\mathbf{A}\bar{x} = b$ como $\mathbf{B}x_B + \mathbf{N}x_N = b$. Se tomarmos $x_N = 0$, segue que $\mathbf{B}x_B = b$. Dizemos que $\bar{x} = [x_B^\top, 0^\top]^\top$ é uma *solução básica* e as componentes de x_B são denominadas *variáveis básicas*. Se alguma componente de x_B for nula, então \bar{x} é dita *solução básica degenerada* para (1.4).

Supondo que o vetor $\tilde{x} \in \mathbb{R}^n$ é uma solução para o sistema linear $\mathbf{A}x = b$ e que $\tilde{x} \geq 0$, então dizemos que \tilde{x} é uma *solução factível* ou *viável*. Se $\tilde{x} = [x_B^\top, 0^\top]^\top$, então dizemos que é uma *solução básica factível* para (1.4) (pode ser degenerada ou não).

Agora, se $x^* \in \mathbb{R}^n$ é solução de $\mathbf{A}x = b$, $x \geq 0$ e tem, ainda, a propriedade de fornecer o menor valor para $c^\top x$, então x^* é uma *solução ótima factível* para (1.4). Se $x^* = [x_B^\top, 0^\top]^\top$, então dizemos que é uma *solução básica ótima factível* para (1.4).

1.2 Convexidade

Dizemos que um conjunto $\mathcal{X} \subseteq \mathbb{R}^n$ é *convexo* se, para quaisquer pontos $x, y \in \mathcal{X}$, o segmento xy está inteiramente contido no conjunto \mathcal{X} , isto é, para quaisquer pontos $x, y \in \mathcal{X}$ e todo $t \in [0, 1]$, temos $tx + (1 - t)y \in \mathcal{X}$. Os conjuntos convexos que comumente encontramos na literatura de programação linear são de um tipo especial: *poliedros convexos* (MATOUSEK; GÄRTNER, 2007).

Um *hiperplano* em \mathbb{R}^n é um subespaço de dimensão $n - 1$. Um hiperplano divide o \mathbb{R}^n em dois *semi-espacos* e constitui sua fronteira comum. Um *semi-espaco* é o conjunto de todas as soluções de uma única desigualdade linear (com pelo menos um coeficiente não nulo). Para o hiperplano com equação $a_1x_1 + a_2x_2 + \dots + a_nx_n = b$, por exemplo, os dois semi-espacos são $\{x \in \mathbb{R}^n; a_1x_1 + a_2x_2 + \dots + a_nx_n \geq b\}$ e $\{x \in \mathbb{R}^n; a_1x_1 + a_2x_2 + \dots + a_nx_n \leq b\}$ (MATOUSEK; GÄRTNER, 2007).

Um *poliedro convexo* é uma interseção de finitos semi-espacos fechados em \mathbb{R}^n . Um semi-espaco é convexo, bem como a interseção de semi-espacos. O conjunto de todas as soluções de um sistema que contém um número finito de desigualdades lineares, também conhecido como o conjunto de todas as soluções viáveis de um programa linear, é, geometricamente, a interseção de um número finito de semi-espacos, ou seja, um poliedro convexo. Notemos que um poliedro convexo pode ser ilimitado. Um poliedro convexo limitado recebe o nome de *politopo convexo* (MATOUSEK; GÄRTNER, 2007).

Um ponto x em um conjunto convexo \mathcal{X} é dito *ponto extremo* de \mathcal{X} se não existem dois pontos distintos $x_1, x_2 \in \mathcal{X}$ tais que $x = tx_1 + (1 - t)x_2$, para algum $t \in (0, 1)$. Um *vértice* é definido como um ponto onde alguma função linear atinge um máximo único. Assim, um ponto v é chamado de vértice de um poliedro convexo $\mathcal{X} \subseteq \mathbb{R}^n$ se $v \in \mathcal{X}$ e existe um vetor $c \in \mathbb{R}^n$, não nulo, onde $c^\top v > c^\top y$, qualquer que seja $y \in \mathcal{X} \setminus \{v\}$. Geometricamente, isso significa que o hiperplano $\{x \in \mathbb{R}^n; c^\top x = c^\top v\}$ toca o poliedro \mathcal{X} exatamente em v . Para poliedros convexos, pontos extremos são exatamente os vértices (MATOUSEK; GÄRTNER, 2007).

Um subconjunto $\mathcal{F} \subseteq \mathcal{X}$ é uma *face k-dimensional* de um poliedro convexo \mathcal{X} se \mathcal{F} tem dimensão k e existem um vetor $c \in \mathbb{R}^n$, não nulo, e um escalar $z \in \mathbb{R}$ tais que $c^\top x = z$, qualquer que seja $x \in \mathcal{F}$, e $c^\top x < z$, qualquer que seja $x \in \mathcal{X} \setminus \mathcal{F}$. Em outras palavras, existe um hiperplano que toca \mathcal{X} exatamente em \mathcal{F} . Como \mathcal{F} é a interseção de um hiperplano com um poliedro convexo, \mathcal{F} é um poliedro convexo e sua dimensão está bem definida (MATOUSEK; GÄRTNER, 2007).

Teorema 1.1. *Seja $\mathbf{A} \in \mathbb{M}_{m \times n}(\mathbb{R})$ com $\text{posto}(\mathbf{A}) = m$ e um vetor $b \in \mathbb{R}^m$. Considere \mathcal{X} um politopo consistindo de todos os vetores $x \in \mathbb{R}^n$ satisfazendo*

$$\begin{aligned} \mathbf{A}x &= b \\ x &\geq 0. \end{aligned} \tag{1.5}$$

Um vetor x é um ponto extremo de \mathcal{X} se, e somente se, x é uma solução básica factível para (1.5).

Demonstração. Vide [Luemberger e Ye \(2008, p. 23\)](#). □

Corolário 1.1. *Se existe uma solução ótima finita para o problema de programação linear (1.4), então existe uma solução ótima finita que é ponto extremo do conjunto de restrições.*

Demonstração. Vide [Luemberger e Ye \(2008, p. 24\)](#). □

Teorema 1.2. *Considere o seguinte problema de programação linear*

$$\begin{aligned} \min \quad & c^\top x \\ \text{s. a} \quad & \mathbf{A}x = b \\ & x \geq 0. \end{aligned}$$

(i) *Se houver pelo menos uma solução viável e a função objetivo é limitada superiormente no conjunto de todas as soluções viáveis, então existe uma solução ótima;*

(ii) *Se existe uma solução ótima, então existe uma solução básica factível que é ótima.*

Demonstração. Vide [Matousek e Gärtner \(2007, p. 54\)](#). □

Para qualquer conjunto $\mathcal{X} \subseteq \mathbb{R}^n$, o menor conjunto contendo \mathcal{X} , denominado *casca convexa* de \mathcal{X} , pode ser construído como a interseção de todos os conjuntos convexos contendo \mathcal{X} :

$$\text{conv}(\mathcal{X}) := \bigcap \{ \mathcal{S} \subseteq \mathbb{R}^n; \mathcal{X} \subseteq \mathcal{S}, \mathcal{S} \text{ convexo} \}.$$

Um politopo convexo \mathcal{X} , isto é, um poliedro convexo, sempre possui vértices; \mathcal{X} é igual à casca convexa do conjunto de seus vértices ([ZIEGLER, 2006](#)).

1.3 Dualidade e condições de otimalidade

Consideremos um problema de programação linear na sua forma canônica dada por (1.2). Associado a este problema está o problema *dual* de programação linear dado como

$$\begin{aligned} \max \quad & y^\top b \\ \text{s. a} \quad & y \in \mathcal{Y}, \end{aligned} \tag{1.6}$$

onde $\mathcal{Y} := \{y \in \mathbb{R}^m; y^\top \mathbf{A} \leq c^\top, y \geq 0\}$. Ressaltamos que esta é a forma simétrica de Von Neumann. A motivação para a construção do problema dual é encontrar limitantes superiores para o valor ótimo do problema *primal* (1.2), e detalhes sobre o desenvolvimento da motivação podem ser obtidos em Chvátal (1983, p. 54). Dado um problema primal de programação linear, sempre podemos encontrar o seu problema dual.

Exemplo 1.5. Considere o problema de programação linear na sua forma padrão

$$\begin{aligned} \min \quad & c^\top x \\ \text{s. a} \quad & \mathbf{A}x = b \\ & x \geq 0. \end{aligned}$$

Vamos encontrar o problema dual. A igualdade $\mathbf{A}x = b$ pode ser escrita como $\mathbf{A}x \geq b$ e $-\mathbf{A}x \geq -b$. Logo, o problema acima pode ser escrito na forma canônica e, então, podemos definir seu dual como sendo

$$\begin{aligned} \max \quad & y_1^\top b - y_2^\top b \\ \text{s. a} \quad & y_1^\top \mathbf{A} - y_2^\top \mathbf{A} \leq c^\top \\ & y_1, y_2 \geq 0. \end{aligned}$$

Se definirmos $y = y_1 - y_2$, percebe-se que y se torna uma variável livre. Assim, reescrevemos o problema como

$$\begin{aligned} \max \quad & y^\top b \\ \text{s. a} \quad & y^\top \mathbf{A} \leq c^\top. \end{aligned}$$

Exemplo 1.6. Considere o problema de programação linear dado abaixo e sua forma matricial.

$$\begin{aligned} \min \quad & c_1^\top x_1 + c_2^\top x_2 + c_3^\top x_3 \\ \text{s. a} \quad & \mathbf{A}_1 x_1 \geq b_1 \\ & \mathbf{A}_2 x_2 \leq b_2 \\ & \mathbf{A}_3 x_3 = b_3 \\ & x_1, x_2, x_3 \geq 0 \end{aligned} \quad \Rightarrow \quad \begin{aligned} \min \quad & [c_1^\top, c_2^\top, c_3^\top][x_1^\top, x_2^\top, x_3^\top]^\top \\ \text{s. a} \quad & \begin{bmatrix} \mathbf{A}_1 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -\mathbf{A}_2 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{A}_3 \\ \mathbf{0} & \mathbf{0} & -\mathbf{A}_3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \geq \begin{bmatrix} b_1 \\ -b_2 \\ b_3 \\ -b_3 \end{bmatrix} \\ & x_1, x_2, x_3 \geq 0. \end{aligned}$$

Seu dual é dado como segue, tomando $y = y_3 - y_4$ livre

$$\begin{aligned} \max \quad & [y_1^\top, y_2^\top, y_3^\top, y_4^\top][b_1^\top, -b_2^\top, b_3^\top, -b_3^\top]^\top \\ \text{s. a} \quad & \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix}^\top \begin{bmatrix} \mathbf{A}_1 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -\mathbf{A}_2 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{A}_3 \\ \mathbf{0} & \mathbf{0} & -\mathbf{A}_3 \end{bmatrix} \leq \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}^\top \\ & y_1, y_2, y_3, y_4 \geq 0 \end{aligned} \quad \Rightarrow \quad \begin{aligned} \max \quad & y_1^\top b_1 - y_2^\top b_2 + y_3^\top b_3 \\ \text{s. a} \quad & y_1^\top \mathbf{A}_1 \leq c_1^\top \\ & y_2^\top \mathbf{A}_2 \geq -c_2^\top \\ & y_3^\top \mathbf{A}_3 \leq c_3^\top \\ & y_1, y_2 \geq 0 \\ & y \text{ livre.} \end{aligned}$$

Assim como visto nos exemplos anteriores, podemos encontrar problemas em que o vetor de variáveis de decisão x está limitado superior e inferiormente, bem como ser livre.

Exemplo 1.7. Considere o problema de programação linear dado abaixo e sua forma matricial.

$$\begin{array}{ll}
 \min & c_1^\top x_1 + c_2^\top x_2 + c_3^\top x_3 \\
 \text{s. a} & \mathbf{A}_1 x_1 \geq b_1 \\
 & \mathbf{A}_2 x_2 \leq b_2 \\
 & \mathbf{A}_3 x_3 = b_3 \\
 & 0 \leq x_1 \leq u_1 \\
 & 0 \leq x_2 \leq u_2 \\
 & 0 \leq x_3 \leq u_3
 \end{array}
 \quad \Longrightarrow \quad
 \begin{array}{ll}
 \min & [c_1^\top, c_2^\top, c_3^\top][x_1^\top, x_2^\top, x_3^\top]^\top \\
 \text{s. a} & \begin{bmatrix} \mathbf{A}_1 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -\mathbf{A}_2 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{A}_3 \\ \mathbf{0} & \mathbf{0} & -\mathbf{A}_3 \\ -\mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -\mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & -\mathbf{I} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \geq \begin{bmatrix} b_1 \\ -b_2 \\ b_3 \\ -b_3 \\ -u_1 \\ -u_2 \\ -u_3 \end{bmatrix} \\
 & x_1, x_2, x_3 \geq 0.
 \end{array}$$

Seu dual é dado como segue

$$\begin{array}{ll}
 \max & [y_1^\top, y_2^\top, y_3^\top, y_4^\top, y_5^\top, y_6^\top, y_7^\top][b_1^\top, -b_2^\top, b_3^\top, -b_3^\top, -u_1^\top, -u_2^\top, -u_3^\top]^\top \\
 \text{s. a} & \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \end{bmatrix}^\top \begin{bmatrix} \mathbf{A}_1 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -\mathbf{A}_2 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{A}_3 \\ \mathbf{0} & \mathbf{0} & -\mathbf{A}_3 \\ -\mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -\mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & -\mathbf{I} \end{bmatrix} \leq \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}^\top \\
 & y_i \geq 0 \quad i = 1, 2, \dots, 7.
 \end{array}$$

Agora, tomando $y = y_3 - y_4$ livre, temos

$$\begin{array}{ll}
 \max & y_1^\top b_1 - y_2^\top b_2 + y^\top b_3 - y_5^\top u_1 - y_6^\top u_2 - y_7^\top u_3 \\
 \text{s. a} & y_1^\top \mathbf{A}_1 - y_5^\top \mathbf{I} \leq c_1^\top \\
 & y_2^\top \mathbf{A}_2 + y_6^\top \mathbf{I} \geq -c_2^\top \\
 & y^\top \mathbf{A}_3 - y_7^\top \mathbf{I} \leq c_3^\top \\
 & y_1, y_2, y_5, y_6, y_7 \geq 0 \\
 & y \text{ livre.}
 \end{array}$$

Exemplo 1.8. Considere o problema de programação linear dado abaixo, onde $x_1, x_2, x_3 \in \mathbb{R}^n$ são livres

$$\begin{array}{ll}
 \min & c_1^\top x_1 + c_2^\top x_2 + c_3^\top x_3 \\
 \text{s. a} & \mathbf{A}_1 x_1 \leq b_1 \\
 & \mathbf{A}_2 x_2 \geq b_2 \\
 & \mathbf{A}_3 x_3 = b_3.
 \end{array}$$

Sua forma matricial canônica, já com as mudanças de variáveis dadas por $x_i = u_i - v_i$, $u_i, v_i \geq 0$, $i = 1, 2, 3$, pode ser dada como

$$\begin{aligned} \min \quad & [c_1^\top, -c_1^\top, c_2^\top, -c_2^\top, c_3^\top, -c_3^\top] [u_1^\top, v_1^\top, u_2^\top, v_2^\top, u_3^\top, v_3^\top]^\top \\ \text{s. a} \quad & \begin{bmatrix} \mathbf{A}_1 & -\mathbf{A}_1 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & -\mathbf{A}_2 & \mathbf{A}_2 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{A}_3 & -\mathbf{A}_3 \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & -\mathbf{A}_3 & \mathbf{A}_3 \end{bmatrix} \begin{bmatrix} u_1 \\ v_1 \\ u_2 \\ v_2 \\ u_3 \\ v_3 \end{bmatrix} \geq \begin{bmatrix} b_1 \\ -b_2 \\ b_3 \\ -b_3 \end{bmatrix} \\ & u_i, v_i \geq 0 \quad i = 1, 2, 3. \end{aligned}$$

Seu dual pode ser dado como

$$\begin{aligned} \max \quad & [y_1^\top, y_2^\top, y_3^\top, y_4^\top] [b_1^\top, -b_2^\top, b_3^\top, -b_3^\top]^\top \\ \text{s. a} \quad & \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix}^\top \begin{bmatrix} \mathbf{A}_1 & -\mathbf{A}_1 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & -\mathbf{A}_2 & \mathbf{A}_2 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{A}_3 & -\mathbf{A}_3 \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & -\mathbf{A}_3 & \mathbf{A}_3 \end{bmatrix} \leq \begin{bmatrix} c_1 \\ -c_1 \\ c_2 \\ -c_2 \\ c_3 \\ -c_3 \end{bmatrix} \\ & y_1, y_2, y_3, y_4 \geq 0. \end{aligned}$$

Em forma matricial, já considerando $y = y_3 - y_4$ livre, pode ser dada por

$$\begin{aligned} \max \quad & y_1^\top b_1 - y_2^\top b_2 + y^\top b_3 \\ \text{s. a} \quad & y_1^\top \mathbf{A}_1 = c_1^\top \\ & y_2^\top \mathbf{A}_2 = -c_2^\top \\ & y^\top \mathbf{A}_3 = c_3^\top \\ & y_1, y_2 \geq 0 \\ & y \text{ livre.} \end{aligned}$$

O problema dual em (1.6) pode ser reescrito como

$$\begin{aligned} \max \quad & b^\top y \\ \text{s. a} \quad & \mathbf{A}^\top y \leq c \\ & y \geq 0. \end{aligned} \tag{1.7}$$

Considere o problema primal (forma canônica) e seu respectivo dual

$$\begin{aligned} (P_c) \quad & \min_x \{c^\top x; \mathbf{A}x \geq b, x \geq 0\} \\ (D_c) \quad & \max_y \{b^\top y; \mathbf{A}^\top y \leq c, y \geq 0\}. \end{aligned}$$

Lema 1.1 (Fraco de Dualidade). *Seja x factível para (P_c) e y factível para (D_c) . Então*

$$b^\top y \leq c^\top x.$$

Demonstração. Se x é factível para (P_c) e y é factível para (D_c) , então $x \geq 0$, $y \geq 0$, $\mathbf{A}x \geq b$ e $\mathbf{A}^\top y \leq c$. Daí, segue que $b^\top y \leq (\mathbf{A}x)^\top y = x^\top \mathbf{A}^\top y \leq x^\top c = c^\top x$. Logo, $b^\top y \leq c^\top x$. \square

Observe que uma solução factível para (D_c) fornece um limitante inferior para (P_c) e uma solução factível para (P_c) fornece um limitante superior para (D_c) .

Corolário 1.2. *Se x é factível para (P_c) , y é factível para (D_c) e $c^\top x = b^\top y$, então x é uma solução ótima para (P_c) e y é uma solução ótima para (D_c) .*

Demonstração. Suponha, por absurdo, que x e y não sejam soluções ótimas, ou seja, existem vetores \bar{x} e \bar{y} factíveis para (P_c) e (D_c) , respectivamente, tais que $c^\top \bar{x} < c^\top x$ e $b^\top \bar{y} > b^\top y$. Como são factíveis, $\mathbf{A}\bar{x} \geq b$ e $\mathbf{A}^\top \bar{y} \leq c$. Mas como $c^\top x = b^\top y$, obtemos $c^\top \bar{x} < c^\top x = b^\top y < b^\top \bar{y}$. Ou seja, concluímos que $b^\top \bar{y} > c^\top \bar{x}$, o que é um absurdo pelo lema anterior. \square

Teorema 1.3 (Forte de Dualidade). *Se (P_c) e (D_c) possuem soluções factíveis, então existem x^* e y^* soluções ótimas factíveis para (P_c) e (D_c) , respectivamente, tais que $b^\top y^* = c^\top x^*$.*

A demonstração para este teorema pode ser encontrada em [Dantzig e Thapa \(2003, p. 45\)](#) e em [Matousek e Gärtner \(2007, p. 88-89, 94\)](#). Neste último, duas versões da demonstração são apresentadas: a primeira usando o Método Simplex para programação linear e a segunda usando o Lema de Farkas.

A diferença $c^\top x - b^\top y \geq 0$ é chamada de *gap* de dualidade para o par (x, y) . Se para algum par (x^*, y^*) o gap se anula, então tal par é constituído por soluções ótimas para (P_c) e (D_c) , respectivamente ([ROOS; TERLAKY; VIAL, 2005](#), p. 19).

Podemos escrever os problemas (P_c) e (D_c) na forma padrão, acrescentando às suas formulações as variáveis de folga e excesso não negativas x_s e y_s , obtendo, respectivamente

$$\begin{aligned} (P_p) \quad & \min \quad z = c^\top x \\ & \text{s. a} \quad \mathbf{A}x - \mathbf{I}x_s = b \\ & \quad x \geq 0 \\ & \quad x_s = (x_{n+1}, x_{n+2}, \dots, x_{n+m})^\top \in \mathbb{R}_+^m \end{aligned}$$

e

$$\begin{aligned} (D_p) \quad & \max \quad v = b^\top y \\ & \text{s. a} \quad \mathbf{A}^\top y + \mathbf{I}y_s = c \\ & \quad y \geq 0 \\ & \quad y_s = (y_{m+1}, y_{m+2}, \dots, y_{m+n})^\top \in \mathbb{R}_+^n. \end{aligned}$$

Seja (x, x_s) solução factível para (P_p) e (y, y_s) solução factível para (D_p) . De $\mathbf{A}x - \mathbf{I}x_s = b$ segue que $c^\top x - y^\top \mathbf{A}x + y^\top x_s = z - y^\top b$. Por outro lado, de $\mathbf{A}^\top y + \mathbf{I}y_s = c$ segue

que $x^\top \mathbf{A}^\top y + x^\top y_s - b^\top y = x^\top c - v$. Somando estas equações, temos $x^\top y_s + y^\top x_s = z - v \geq 0$. O Teorema 1.3 garante a existência de uma solução ótima factível (x^*, x_s^*) e (y^*, y_s^*) onde $z^* = c^\top x^* = b^\top y^* = v^*$. Assim, $(x^*)^\top y_s^* + (y^*)^\top x_s^* = z^* - v^* = 0$. Como x^* , y^* , x_s^* e y_s^* são não negativos, temos $(x^*)^\top y_s^* = 0$ e $(y^*)^\top x_s^* = 0$. Logo, $(x^*)_i (y_s^*)_i = 0$ para $i = 1, \dots, n$ e $(y^*)_j (x_s^*)_j = 0$ para $j = 1, \dots, m$. Isto prova o seguinte teorema:

Teorema 1.4 (Folga Complementar ou Complementaridade). *Para soluções ótimas factíveis de (P_p) e (D_p) , sempre que a k -ésima relação de qualquer sistema possui folga, a k -ésima variável de seu dual é nula. Se a k -ésima variável for positiva em qualquer sistema, a k -ésima relação não possui folga. Ou seja,*

$$x_k y_{m+k} = 0, \quad k = 1, \dots, n \quad e \quad y_k x_{n+k} = 0, \quad k = 1, \dots, m$$

De acordo com Jansen (1997, p 14), em métodos de pontos interiores, a complementaridade estrita é o tema central. Considere o problema primal (P) na forma padrão e seu respectivo dual (D), também na forma padrão, onde $s \in \mathbb{R}_+^n$ é uma variável de folga acrescentada à formulação dual

$$\begin{aligned} (P) \quad & \min_x \{c^\top x; \mathbf{A}x = b, x \geq 0\}, \\ (D) \quad & \max_{y,s} \{b^\top y; \mathbf{A}^\top y + s = c, s \geq 0\}. \end{aligned}$$

Sejam \mathcal{X} e \mathcal{D} os conjuntos de soluções factíveis para (P) e (D), respectivamente.

Teorema 1.5. *Se (P) e (D) são factíveis, então existe $x^* \in \mathcal{X}$ e $(y^*, s^*) \in \mathcal{D}$ tais que $(x^*)^\top s^* = 0$ e $x_i^* + s_i^* > 0$, $i = 1, \dots, n$.*

Demonstração. Vide Goldman e Tucker (1956 apud JANSEN, 1997, p. 15). □

De acordo com Wright (1997, p. 3), as condições de otimalidade são condições algébricas que devem ser satisfeitas pelas soluções dos problemas de programação matemática. Em programação linear, tais condições são oriundas da teoria de dualidade e também podem ser tratadas como casos especiais das condições de otimalidade para otimização restrita geral, conhecidas como condições de Karush-Kuhn-Tucker (KKT) (que podem ser encontradas em Wright (1997, p. 240)). Resumidamente, temos:

O vetor $x^* \in \mathbb{R}^n$ é uma solução ótima de (P) se, e somente se, existem vetores $y^* \in \mathbb{R}^m$ e $s^* \in \mathbb{R}^n$ para os quais as seguintes condições são válidas

$$\mathbf{A}x^* = b \quad (1.8a)$$

$$\mathbf{A}^\top y^* + s^* = c \quad (1.8b)$$

$$x_i^* s_i^* = 0, \quad i = 1, \dots, n \quad (1.8c)$$

$$(x^*, s^*) \geq 0. \quad (1.8d)$$

O vetor $(y^*, s^*) \in \mathbb{R}^m \times \mathbb{R}^n$ é uma solução de (D) se, e somente se, existe um vetor $x^* \in \mathbb{R}^n$ tal que as condições (1.8) são válidas.

Definição 1.1. *Sejam x factível para (P) e (y, s) factível para (D). Dizemos que (x, y, s) é uma solução primal-dual factível para (P) e (D).*

Definição 1.2. *Sejam x factível para (P) e (y, s) factível para (D), tais que $x > 0$ e $s > 0$. Então (x, s) é um par primal-dual positivo e (x, y, s) é uma solução primal-dual estritamente factível.*

Reescrevendo as condições de otimalidade acima, segue que uma solução primal-dual dada (x^*, y^*, s^*) é ótima para (P) e (D) se, e somente se, é satisfeito que

$$\mathbf{A}x^* - b = 0, \quad x^* \geq 0, \quad (1.9a)$$

$$\mathbf{A}^\top y^* + s^* - c = 0, \quad s^* \geq 0, \quad (1.9b)$$

$$\mathbf{X}^* \mathbf{S}^* e = 0, \quad (1.9c)$$

onde $\mathbf{X}^* = \text{diag}(x^*)$, $\mathbf{S}^* = \text{diag}(s^*)$ e $e = (1, \dots, 1)^\top$.

Conforme explica Wright (1997, p. 5) todos os métodos primais-duais geram soluções iteradas $(x^{(k)}, y^{(k)}, s^{(k)})$ que satisfazem a restrição $(x^{(k)}, s^{(k)}) > 0$, propriedade que deu origem ao termo *ponto interior*. Em métodos tipo primal-dual que iniciam com um ponto inicial factível, cada iteração satisfaz o sistema

$$\mathbf{A}x = b, \quad x > 0, \quad (1.10a)$$

$$\mathbf{A}^\top y + s = c, \quad s > 0, \quad (1.10b)$$

$$x_i s_i = \omega_i, \quad i = 1, \dots, n, \quad (1.10c)$$

onde $\omega = (\omega_1, \omega_2, \dots, \omega_n) \in \mathbb{R}_{++}^n$. Observe que para (x, s) satisfazendo o sistema (1.10), o resíduo na complementaridade é dado por $x^\top s = e^\top \omega$. Fazendo $\omega_i = \tau > 0$, para todo i , em (1.10), as soluções variando o escalar τ caracterizam o *caminho central* do problema. Decrescendo o valor de τ o caminho central leva a uma solução estritamente complementar do problema, podendo ser usado como guia para otimalidade de métodos seguidores de caminho.

Teorema 1.6. *Suponha que exista ao menos um par prima-dual positivo para (P) e (D). Então, para cada $\omega > 0$, existe um único par primal-dual (x, s) tal que $x_i s_i = \omega_i$, $i = 1, \dots, n$.*

Demonstração. Vide Kojima et al. (1991 apud JANSEN, 1997, p. 16). \square

1.4 Método de pontos interiores primais-duais

Nesta seção, consideramos o par de problemas (P) e (D) definidos anteriormente, bem como as condições de otimalidade (1.8) desenvolvidas com base neles. Conforme explica Wright (1997, p. 4), os métodos de pontos interiores primal-dual encontram soluções primais-duais (x^*, y^*, s^*) aplicando variantes do Método de Newton¹ em (1.9) e modificando as direções de busca e comprimentos de passo de modo a obter pares primais-duais positivos a cada iteração.

Podemos reescrever as equações (1.8a) a (1.8c) fazendo $\mathbf{F}: \mathbb{R}^{2n+m} \rightarrow \mathbb{R}^{2n+m}$ por

$$\mathbf{F}(x, y, s) = \begin{bmatrix} \mathbf{A}x - b \\ \mathbf{A}^\top y + s - c \\ \mathbf{X}\mathbf{S}e \end{bmatrix} = 0, \quad (1.11)$$

onde $\mathbf{X} = \text{diag}(x_1, x_2, \dots, x_n)$, $\mathbf{S} = \text{diag}(s_1, s_2, \dots, s_n)$ e $e = (1, 1, \dots, 1)^\top \in \mathbb{R}^n$. \mathbf{F} é linear nos dois primeiros termos e não linear no termo remanescente. Todos os métodos primais-duais garantem soluções iteradas $(x^{(k)}, y^{(k)}, s^{(k)})$ que fornecem pares primais-duais positivos.

Em programação matemática, a maioria dos métodos iterativos possuem dois procedimentos que se destacam, a saber um procedimento que determina o passo (direções de busca) e um procedimento que determina a conveniência de cada ponto no espaço de busca. Wright (1997, p. 5) esclarece que os métodos de pontos interiores primais-duais têm como base para o primeiro procedimento o método de Newton aplicado às equações não lineares (1.11). Ele forma um modelo linear em torno do ponto atual e obtém uma direção de busca $(\Delta x, \Delta y, \Delta s)$ resolvendo o seguinte sistema de equações lineares

$$\mathbf{J}(x, y, s) \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta s \end{bmatrix} = -\mathbf{F}(x, y, s), \quad (1.12)$$

onde \mathbf{J} é o Jacobiano de \mathbf{F} . Considere o conjunto \mathcal{F} de soluções primais-duais factíveis e o conjunto \mathcal{F}° de soluções primais-duais estritamente factíveis, ou seja, conforme Definições 1.1 e 1.2

$$\mathcal{F} := \{(x, y, s); \mathbf{A}x = b, \mathbf{A}^\top y + s = c, (x, s) \geq 0\},$$

¹ O Método de Newton está descrito no Apêndice A.

$$\mathcal{F}^\circ := \{(x, y, s); \mathbf{A}x = b, \mathbf{A}^\top y + s = c, (x, s) > 0\}.$$

Se a solução atual é estritamente factível as equações de Newton se tornam

$$\begin{bmatrix} \mathbf{A} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{A}^\top & \mathbf{I} \\ \mathbf{S} & \mathbf{0} & \mathbf{X} \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta s \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -\mathbf{XSe} \end{bmatrix}. \quad (1.13)$$

Se admitirmos um passo completo ao longo desta direção, poderemos violar a condição $(x, s) \geq 0$. Em vez disso, realizamos uma busca ao longo da linha de direção de Newton, tal que a nova iterada seja $(x, y, s) + \alpha(\Delta x, \Delta y, \Delta s)$ para algum parâmetro $\alpha \in (0, 1]$. Como, na maioria das vezes, podemos somente tomar um pequeno passo ao longo da direção ($\alpha \ll 1$) antes de violar a condição $(x, s) > 0$, a direção de Newton pura não nos permite fazer grandes progressos em direção a uma solução. Para evitar esta dificuldade, os métodos primais-duais modificam o procedimento de Newton de forma que as direções de busca estejam polarizadas dentro do octante não negativo $(x, s) \geq 0$ (desta forma se pode mover ainda mais ao longo da direção sem violar a positividade). Outra modificação no procedimento de Newton, também observado nos métodos primais-duais, é manter as componentes de (x, s) movendo-se próximas da fronteira do octante não negativo (pequenos progressos podem ser feitos) (WRIGHT, 1997).

1.4.1 Métodos seguidores de caminho: o método Preditor-Corretor

O *caminho central* é um arco \mathcal{C} de pontos primais-duais estritamente factíveis. Parametrizando por um escalar $\tau > 0$, cada ponto $(x_\tau, y_\tau, s_\tau) \in \mathcal{C}$ satisfaz o seguinte sistema, para $(x, y, s) = (x_\tau, y_\tau, s_\tau)$

$$\mathbf{A}x = b, \quad x > 0, \quad (1.14a)$$

$$\mathbf{A}^\top y + s = c, \quad s > 0, \quad (1.14b)$$

$$x_i s_i = \tau, \quad i = 1, \dots, n, \quad (1.14c)$$

ou, equivalentemente

$$\mathbf{F}(x_\tau, y_\tau, s_\tau) = \begin{bmatrix} 0 \\ 0 \\ \tau e \end{bmatrix}, \quad (x_\tau, s_\tau) > 0. \quad (1.15)$$

É possível mostrar que (x_τ, y_τ, s_τ) é definido unicamente para cada $\tau > 0$ se, e somente se, $\mathcal{F}^\circ \neq \emptyset$ (WRIGHT, 1997).

Podemos observar que, à medida que τ tende a zero, as equações (1.14) se aproximam das equações (1.8). Assim, se \mathcal{C} é convergente quando $\tau \rightarrow 0$ então deve convergir para uma solução primal-dual do problema de programação linear. Logo, podemos

modificar o procedimento de Newton para que as direções de busca forneçam passos em direção a pontos de \mathcal{C} para os quais $\tau > 0$ (desta forma podemos dar passos mais longos no interior do octante $(x, s) \geq 0$).

Considere o *parâmetro de centragem* $\sigma \in [0, 1]$ e a *medida de dualidade*

$$\mu := \frac{1}{n} \sum_{i=1}^n x_i s_i,$$

que fornece o valor médio dos pares de produtos $x_i s_i$. As equações de passo podem então ser definidas como

$$\begin{bmatrix} \mathbf{A} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{A}^\top & \mathbf{I} \\ \mathbf{S} & \mathbf{0} & \mathbf{X} \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta s \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -\mathbf{X}\mathbf{S}e + \sigma\mu e \end{bmatrix}. \quad (1.16)$$

Nas equações acima, o passo de Newton $(\Delta x, \Delta y, \Delta s)$ é em direção a um ponto $(x_{\sigma\mu}, y_{\sigma\mu}, s_{\sigma\mu}) \in \mathcal{C}$. Por um lado, se $\sigma = 1$ dizemos que o sistema (1.16) define uma *direção de centragem*, que é tendenciosa em direção ao octante não negativo e faz pouco progresso na redução de μ . Contudo, direções de centragem fornecem condições favoráveis para progressos substanciais na iteração subsequente, devido à proximidade apresentada pela solução com o arco \mathcal{C} , permitindo, assim, passos maiores na próxima iteração. Por outro lado, $\sigma = 0$ fornece um passo padrão de Newton, como em (1.11), também chamado de direção *afim-escala*. Em linhas gerais, o [Algoritmo 1](#) apresenta um resumo dos procedimentos encontrados na maioria dos métodos de pontos interiores primais-duais ([WRIGHT, 1997](#)).

Algoritmo 1: Método de pontos interiores primal-dual

Entrada: $(x^{(0)}, y^{(0)}, s^{(0)}) \in \mathcal{F}^\circ$

Saída: (x^*, y^*, s^*)

início

para $k = 1, 2, \dots$ **faça**

 Faça $(x, y, s) \leftarrow (x^{(k)}, y^{(k)}, s^{(k)})$, $\sigma \leftarrow \sigma_{(k)}$, $\mu \leftarrow \mu_{(k)}$ e resolva (1.16) onde

$\sigma_{(k)} \in [0, 1]$ e $\mu_{(k)} = \frac{1}{n} (x^{(k)})^\top s^{(k)}$;

 Faça $(x^{(k+1)}, y^{(k+1)}, s^{(k+1)}) \leftarrow (x^{(k)}, y^{(k)}, s^{(k)}) + \alpha_{(k)} (\Delta x^{(k)}, \Delta y^{(k)}, \Delta s^{(k)})$,

 escolhendo $\alpha_{(k)}$ tal que $(x^{(k+1)}, s^{(k+1)}) > 0$;

fim

fim

As soluções iteradas resultantes das direções de centragem possuem boas propriedades pela proximidade com \mathcal{C} . Para aproveitar tais propriedades e possibilitar maiores progressos na redução de μ , podemos definir uma vizinhança do caminho central \mathcal{C} e restringir as iteradas a esta vizinhança, seguindo \mathcal{C} até obtermos a solução ótima do

problema de programação linear. Algoritmos que realizam estes passos são chamados de algoritmos *seguidores de caminho*.

Nestes algoritmos, a medida de dualidade μ é o ingrediente chave no procedimento que determina a conveniência de cada ponto no espaço de busca: ela é forçada a zero quando $k \rightarrow \infty$, fazendo com que as iteradas $(x^{(k)}, y^{(k)}, s^{(k)})$ se tornem mais próximas de satisfazer as condições KKT (1.8). Cada direção de busca é um passo de Newton em direção a um ponto de \mathcal{C} que fornece uma medida de dualidade μ menor que ou igual à medida de dualidade atual (o valor $\tau = \sigma\mu$ é usado). Comumente, os algoritmos desviam do caminho central uma vez que os pares de produtos $x_i s_i$ geralmente não são idênticos, fazendo com que a condição (1.14c) não seja satisfeita exatamente. Tal desvio pode ser medido realizando uma comparação entre os pares de produtos e seu valor médio μ usando, por exemplo, uma norma escalada definida por

$$\frac{1}{\mu} \|\mathbf{XSe} - \mu e\| = \left\| \begin{bmatrix} x_1 s_1 \\ \vdots \\ x_n s_n \end{bmatrix} - \left(\frac{x^\top s}{n} \right) e \right\|. \quad (1.17)$$

Tanto para a norma-2 quanto para a norma- ∞ é possível garantir que x e s sejam estritamente positivos requerendo que a expressão em (1.17) seja estritamente menor que 1 (se a componente i de x ou de s for zero, teremos $\|\mathbf{XSe} - \mu e\| \geq |x_i s_i - \mu| = \mu$). Considerando a norma-2 em (1.17) e restringindo o desvio para ser menor que uma constante $\theta \in [0, 1)$, temos a vizinhança $\mathcal{V}_2(\theta) := \{(x, y, s) \in \mathcal{F}^\circ; \|\mathbf{XSe} - \mu e\|_2 \leq \theta\mu\}$ (tipicamente $\theta = 0, 5$). Podemos considerar a norma- ∞ em (1.17) com uma leve modificação, obtendo a vizinhança norma- ∞ $\mathcal{V}_{-\infty}(\theta) := \{(x, y, s) \in \mathcal{F}^\circ; x_i s_i \geq \gamma\mu\}$, $\gamma \in (0, 1)$ (tipicamente $\gamma = 10^{-3}$) (WRIGHT, 1997).

Podemos observar que quando γ tende a zero a vizinhança $\mathcal{V}_{-\infty}$ se amplia bastante, abrangendo a maior parte da região viável \mathcal{F} . Já a vizinhança \mathcal{V}_2 é mais restritiva pois existem pontos de \mathcal{F}° que não pertencem a \mathcal{V}_2 não importa o quão próximo θ é escolhido de seus limites. Para os métodos primais-duais, as vizinhanças são cônicas, tendendo a serem amplas e abertas à medida que aumentamos a medida de dualidade e diminuindo à medida que $\mu \rightarrow 0$ (WRIGHT, 1997).

Segundo Wright (1997, p. 91), escolhas de $\sigma \in [0, 1]$ atingem o duplo objetivo de reduzir o valor de μ e promover a centralidade (proximidade com \mathcal{C}). O método *Predictor-Corretor* realiza este objetivo em passos duplos: um para reduzir μ ($\sigma = 0$) e outro para promover centralidade ($\sigma = 1$). Para tanto, define-se um par de vizinhanças \mathcal{V}_2 aninhadas uma dentro da outra, de tal modo que as soluções iteradas com k par estejam confinadas no interior comum das vizinhanças, e iteradas com k ímpar estejam na vizinhança exterior. De acordo com Wright (1997, p. 84) este método é, às vezes, dito o método Predictor-Corretor de Mizuno-Todd-Ye, que primeiro declararam e analisaram o método.

O algoritmo inicia com um ponto $(x^{(0)}, y^{(0)}, s^{(0)})$, pertencente à vizinhança mais interna (portanto mais próximo de \mathcal{C}), calcula um passo predictor ($\sigma = 0$) e um comprimento de passo α , tal que a iterada seguinte $(x^{(1)}, y^{(1)}, s^{(1)})$ atinja a fronteira da vizinhança exterior. Posteriormente, o algoritmo calcula um passo corretor ($\sigma = 1$), usando $\alpha = 1$, para obter $(x^{(2)}, y^{(2)}, s^{(2)})$ de volta à vizinhança interior. Observa-se que, nos passos corretores, o valor μ permanece inalterado.

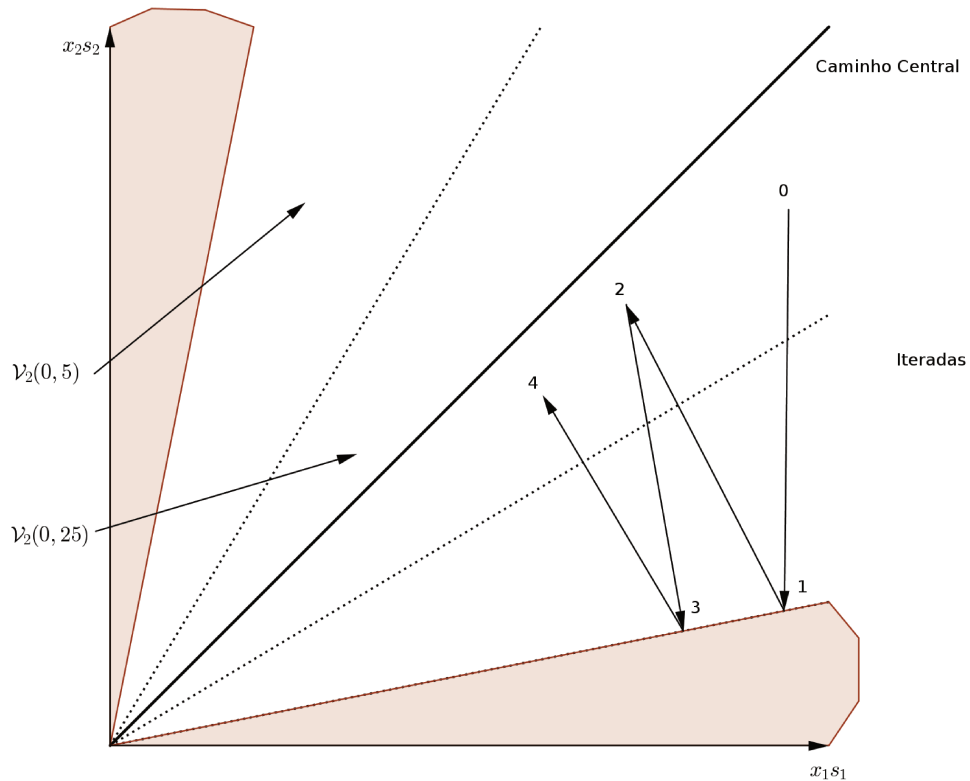
Com intuito ilustrativo, considere as vizinhanças interna $\mathcal{V}_2(0, 25)$ e externa $\mathcal{V}_2(0, 5)$. A [Figura 1](#) ilustra o procedimento descrito acima, que pode ser resumido no [Algoritmo 2](#), onde definimos

$$(x(\alpha), y(\alpha), s(\alpha)) = (x, y, s) + \alpha(\Delta x, \Delta y, \Delta s)$$

e

$$\mu(\alpha) = \frac{1}{n} x(\alpha)^\top s(\alpha).$$

Figura 1 – Soluções iteradas do Algoritmo Predictor-Corretor no espaço (x, s)



Fonte: Produzido pelo autor via software GeoGebra, adaptado de [Wright \(1997, p. 93\)](#).

Algoritmo 2: Método Preditor-Corretor de Mizuno-Todd-Ye**Entrada:** $(x^{(0)}, y^{(0)}, s^{(0)}) \in \mathcal{V}_2(0, 25)$ **Saída:** (x^*, y^*, s^*) **início** **para** $k = 1, 2, \dots$ **faça** **se** k *é par* **então**

/* passo preditor

*/

 Faça $(x, y, s) \leftarrow (x^{(k)}, y^{(k)}, s^{(k)})$, $\sigma \leftarrow \sigma_{(k)}$, $\mu \leftarrow \mu_{(k)}$ e resolva (1.16) com $\sigma_{(k)} = 0$ para obter $(\Delta x^{(k)}, \Delta y^{(k)}, \Delta s^{(k)})$; Escolha $\alpha_{(k)}$ como o maior valor de $\alpha \in [0, 1]$ tal que $(x^{(k)}(\alpha), y^{(k)}(\alpha), s^{(k)}(\alpha)) \in \mathcal{V}_2(0, 5)$; Faça $(x^{(k+1)}, y^{(k+1)}, s^{(k+1)}) \leftarrow (x^{(k)}(\alpha_{(k)}), y^{(k)}(\alpha_{(k)}), s^{(k)}(\alpha_{(k)}))$; **senão**

/* passo corretor

*/

 Faça $(x, y, s) \leftarrow (x^{(k)}, y^{(k)}, s^{(k)})$, $\sigma \leftarrow \sigma_{(k)}$, $\mu \leftarrow \mu_{(k)}$ e resolva (1.16) com $\sigma_{(k)} = 1$ para obter $(\Delta x^{(k)}, \Delta y^{(k)}, \Delta s^{(k)})$; Faça $(x^{(k+1)}, y^{(k+1)}, s^{(k+1)}) \leftarrow (x^{(k)}, y^{(k)}, s^{(k)}) + (\Delta x^{(k)}, \Delta y^{(k)}, \Delta s^{(k)})$; **fim** **fim****fim**

1.4.2 Ponto inicial infactível: o método Preditor-Corretor de Mehrotra

Em muitos casos, obter uma solução primal-dual factível para iniciar os métodos iterativos pode ser uma tarefa bastante difícil. Mesmo reformulando o problema, distorções podem surgir e tornar o problema reformulado mais difícil de ser resolvido. Uma alternativa para estes inconvenientes é iniciar os métodos com pontos iniciais infactíveis, ou seja, exigindo apenas a positividade do par primal-dual. Neste caso, devemos modificar as direções de busca para que se possa prover factibilidade tão bem quanto centralidade. Para tanto, definamos os resíduos das equações lineares em (1.16) pondo $r_p = b - \mathbf{A}x$, $r_d = c - \mathbf{A}^\top y - s$ e $r_c = \sigma \mu e - \mathbf{X}S e$, para obtermos

$$\begin{bmatrix} \mathbf{A} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{A}^\top & \mathbf{I} \\ \mathbf{S} & \mathbf{0} & \mathbf{X} \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta s \end{bmatrix} = \begin{bmatrix} r_p \\ r_d \\ r_c \end{bmatrix}. \quad (1.18)$$

Neste caso, a direção de busca continua sendo um passo de Newton na direção do ponto $(x_{\sigma\mu}, y_{\sigma\mu}, s_{\sigma\mu}) \in \mathcal{C}$. De acordo com Wright (1997, p. 12), ele tenta remover a infactibilidade nas restrições de igualdade em um único passo. Se um passo completo é sempre tomado ($\alpha = 1$) os resíduos r_p e r_d se anulam e as iteradas subsequentes permanecem estritamente factíveis.

No contexto de métodos infactíveis, destaca-se o método Preditor-Corretor desenvolvido por Mehrotra (1992). O algoritmo de Mehrotra gera uma sequência de iteradas infactíveis (x^k, y^k, s^k) que fornecem pares primais-duais positivos. A direção da busca em cada iteração consiste em três componentes:

1. uma direção “preditora” afim-escala – direção de Newton pura definida por (1.18);
2. um termo de centragem cujo tamanho é regido pela escolha adaptativa do parâmetro de centragem σ ;
3. uma direção “corretora”, que tenta compensar a não linearidade na direção afim-escala.

A escolha adaptativa do parâmetro σ é dependente da direção afim-escala “preditora”: o parâmetro recebe um valor mais próximo de zero se a direção afim-escala faz um bom progresso na redução da medida dualidade μ , permanecendo dentro do octante positivo definido por $(x, s) > 0$, e recebe um valor mais próximo de um se podemos mover apenas a uma curta distância, ao longo da direção de afim-escala, antes de violar as restrições $(x, s) > 0$. Esta configuração de cálculo separado é desvantajosa pela necessidade de resolver dois sistemas lineares a cada iteração. Contudo, como a matriz de coeficientes é a mesma, necessitamos realizar apenas uma decomposição, o que reduz bastante o esforço computacional (WRIGHT, 1997).

O cálculo separado da direção afim-escala permite a estimação do erro na linearização, que por sua vez permite calcular uma componente corretora, melhorando o modelo linear de primeira ordem e obtendo um modelo de segunda ordem (quadrático). Não há necessidade de calcular separadamente as componentes de centragem e correção, uma vez que são independentes uma da outra e são obtidas pela resolução de sistemas lineares que compartilham a mesma matriz. Podemos fundi-las em uma única direção, adicionando seus correspondentes lados direitos, e calcular a direção combinada com uma única retro-substituição. Para o custo de uma única retro-substituição extra, o algoritmo de Mehrotra oferece uma escolha adaptativa de σ e uma melhoria de ordem superior do passo puro de Newton (passo afim-escala). Em geral, o custo extra por iteração é facilmente justificado por uma redução significativa no número de iterações (WRIGHT, 1997).

Mehrotra (1992) usa diferentes comprimentos de passo nas variáveis primal e dual, ideia que tem sido usada no desenvolvimento teórico e prático dos métodos primais-duais desde os primeiros dias. Computacionalmente, este acessório economiza cerca de 10%~20% em um problema típico (WRIGHT, 1997).

1.4.2.1 Uma variante do algoritmo de Mehrotra

Partindo de um ponto inicial (x, y, s) com $(x, s) > 0$, obtemos a solução $(\Delta x^{\text{aff}}, \Delta y^{\text{aff}}, \Delta s^{\text{aff}})$ resolvendo o sistema linear (1.18) com $\sigma = 0$. Os comprimentos de passo ao longo desta direção para as variáveis primal e dual são calculados separadamente pondo

$$\alpha_{\text{aff}}^{\text{pri}} = \max\{\alpha \in [0, 1]; x + \alpha \Delta x^{\text{aff}} \geq 0\}, \quad (1.19a)$$

$$\alpha_{\text{aff}}^{\text{dual}} = \max\{\alpha \in [0, 1]; s + \alpha \Delta s^{\text{aff}} \geq 0\}. \quad (1.19b)$$

A medida da eficiência da direção afim-escala é feita usando a medida

$$\mu_{\text{aff}} = (1/n)(x + \alpha_{\text{aff}}^{\text{pri}} \Delta x^{\text{aff}})^\top (s + \alpha_{\text{aff}}^{\text{dual}} \Delta s^{\text{aff}}). \quad (1.20)$$

O parâmetro de centragem σ é escolhido próximo de zero se $\mu_{\text{aff}} \ll \mu$, e é escolhido próximo de um caso contrário. Desta forma, o método se aproxima do caminho central \mathcal{C} de modo que esteja em boas condições para atingir uma substancial redução da medida de dualidade μ na iteração subsequente. Mehrotra (1992) sugere a heurística seguinte, que provou ser eficaz em exaustivos testes computacionais

$$\sigma = \left(\frac{\mu_{\text{aff}}}{\mu} \right)^3. \quad (1.21)$$

O passo combinado centragem-corretor $(\Delta x^{\text{cc}}, \Delta y^{\text{cc}}, \Delta s^{\text{cc}})$ é obtido resolvendo

$$\begin{bmatrix} \mathbf{A} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{A}^\top & \mathbf{I} \\ \mathbf{S} & \mathbf{0} & \mathbf{X} \end{bmatrix} \begin{bmatrix} \Delta x^{\text{cc}} \\ \Delta y^{\text{cc}} \\ \Delta s^{\text{cc}} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \sigma \mu e - \Delta \mathbf{X}^{\text{aff}} \Delta \mathbf{S}^{\text{aff}} e \end{bmatrix}. \quad (1.22)$$

Os fatores da matriz de coeficientes já estão disponíveis a partir da solução de (1.18), de modo que (1.22) pode ser resolvido com o custo de executar uma única retro-substituição. O Algoritmo 3 resume o procedimento descrito acima.

1.4.3 Calculando as direções de busca

Em termos computacionais, o maior esforço está em resolver o sistema (1.16) ou mesmo (1.18). Podemos reduzir as dimensões do problema e reescrevê-lo de duas outras formas alternativas mais compactas.

Tomando como ponto de partida o sistema (1.18) podemos eliminar a variável Δs escrevendo

$$\begin{bmatrix} -\mathbf{D} & \mathbf{A}^\top \\ \mathbf{A} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} r_d - \mathbf{X}^{-1} r_c \\ r_p \end{bmatrix}, \quad (1.23)$$

onde $\mathbf{D} = \mathbf{X}^{-1} \mathbf{S}$ e Δs pode ser recuperado pondo $\Delta s = \mathbf{X}^{-1}(r_c - \mathbf{S} \Delta x)$. O sistema (1.23) recebe o nome de *sistema aumentado*.

Algoritmo 3: Método Preditor-Corretor de Mehrotra**Entrada:** $(x^{(0)}, y^{(0)}, s^{(0)})$, com $(x^{(0)}, s^{(0)}) > 0$ **Saída:** (x^*, y^*, s^*) **início****para** $k = 0, 1, 2, \dots$ **faça**Faça $(x, y, s) \leftarrow (x^{(k)}, y^{(k)}, s^{(k)})$ e resolva (1.18) obtendo $(\Delta x^{\text{aff}}, \Delta y^{\text{aff}}, \Delta s^{\text{aff}})$;Calcule $\alpha_{\text{aff}}^{\text{pri}}$, $\alpha_{\text{aff}}^{\text{dual}}$ e μ_{aff} de (1.19a), (1.19b) e (1.20), respectivamente;Calcule σ de (1.21);Resolva (1.22) obtendo $(\Delta x^{\text{cc}}, \Delta y^{\text{cc}}, \Delta s^{\text{cc}})$;Faça $(\Delta x^{(k)}, \Delta y^{(k)}, \Delta s^{(k)}) \leftarrow (\Delta x^{\text{aff}}, \Delta y^{\text{aff}}, \Delta s^{\text{aff}}) + (\Delta x^{\text{cc}}, \Delta y^{\text{cc}}, \Delta s^{\text{cc}})$;Faça $\alpha_{\text{max}}^{\text{pri}} \leftarrow \max\{\alpha \geq 0; x^{(k)} + \alpha \Delta x^{(k)} \geq 0\}$;Faça $\alpha_{\text{max}}^{\text{dual}} \leftarrow \max\{\alpha \geq 0; s^{(k)} + \alpha \Delta s^{(k)} \geq 0\}$;Calcule $\alpha_{(k)}^{\text{pri}} \leftarrow \min\{0, 99995\alpha_{\text{max}}^{\text{pri}}; 1\}$ e $\alpha_{(k)}^{\text{dual}} \leftarrow \min\{0, 99995\alpha_{\text{max}}^{\text{dual}}; 1\}$;Faça $x^{(k+1)} \leftarrow x^{(k)} + \alpha_{(k)}^{\text{pri}} \Delta x^{(k)}$;Faça $(y^{(k+1)}, s^{(k+1)}) \leftarrow (y^{(k)}, s^{(k)}) + \alpha_{(k)}^{\text{dual}} (\Delta y^{(k)}, \Delta s^{(k)})$;**fim****fim**

Por sua vez, eliminando Δx do sistema (1.23) acima, temos

$$\mathbf{Z}\Delta y = \mathbf{A}\mathbf{X}e - \sigma\mu\mathbf{A}e - \mathbf{A}\mathbf{D}^{-1}r_c - r_b, \quad (1.24)$$

onde $\mathbf{Z} = \mathbf{A}\mathbf{D}^{-1}\mathbf{A}^\top$ é chamado de *Complemento de Schur* e Δx pode ser recuperado pondo $\Delta x = \mathbf{D}^{-1}\mathbf{A}^\top\Delta y + \mathbf{D}^{-1}r_c - \mathbf{S}^{-1}r_p$.

Observe que \mathbf{D} é uma matriz diagonal, cujas entradas da diagonal principal são positivas, e que, nas matrizes dos sistemas (1.23) e (1.24), somente ela altera os valores de suas entradas a cada iteração. Conforme observa Oliveira (1997, p. 10), algumas entradas convergem para zero quando o método se aproxima de uma solução, enquanto outras tendem para o infinito.

Um fato importante é que muito trabalho pode ser evitado ao realizar uma única fatoração matricial para a variante preditor-corretora, uma vez que em ambos os passos (preditor e corretor) a matriz de coeficientes é a mesma (seja ela reduzida ao sistema aumentado ou ao complemento de Schur) (OLIVEIRA, 1997).

2 Sistema Aumentado

No capítulo anterior, vimos que métodos de pontos interiores podem ser usados para resolver problemas de programação linear, e que os métodos baseiam-se na aplicação do método de Newton nas condições de otimalidade do par de problemas primal-dual, referente ao problema em questão. O sistema linear obtido foi reduzido ao sistema aumentado e posteriormente ao complemento de Schur. Contudo, tais sistemas detêm propriedades distintas para solução. Além disso, neste trabalho, consideramos o uso de métodos iterativos em lugar de métodos diretos para resolver os sistemas lineares. Com isso, emerge a necessidade do uso de técnicas de pré-condicionamento, para acelerar a convergência.

Neste capítulo, abordaremos algumas propriedades inerentes ao sistema aumentado, bem como questões relacionadas a pré-condicionamento de sistemas lineares no contexto do sistema aumentado e do complemento de Schur, e descreveremos a classe de pré-condicionadores apresentada por [Oliveira e Sorensen \(2005, p. 6\)](#). Contudo, alguns resultados sobre inversa de matrizes, bem como alguns conceitos usados no capítulo, necessitam ser apresentados primeiramente.

2.1 Preliminares

O *número de condição* κ_p de uma matriz dada \mathbf{A} , em relação à norma-p, é dado por $\kappa_p(\mathbf{A}) := \|\mathbf{A}\|_p \|\mathbf{A}^{-1}\|_p$. É possível mostrar que $\kappa(\mathbf{A}) \geq 1$. Matrizes com $\kappa(\mathbf{A})$ muito grande são ditas *mal condicionadas*, enquanto que matrizes com $\kappa(\mathbf{A}) \approx 1$ são ditas *bem condicionadas*. Os *autovalores* de uma matriz $\mathbf{A} \in \mathbb{M}_n(\mathbb{C})$ são as raízes do seu *polinômio característico* $p(z) = \det(z\mathbf{I} - \mathbf{A})$. O conjunto dessas raízes é chamado de *espectro* de \mathbf{A} e é denotado por $\lambda(\mathbf{A}) := \{z; \det(z\mathbf{I} - \mathbf{A}) = 0\}$ ([GOLUB; VAN LOAN, 2013](#)).

Seja $\mathbf{A} \in \mathbb{M}_{m \times n}(\mathbb{R})$. O *espaço coluna* de \mathbf{A} é o subconjunto de \mathbb{R}^m definido por $\mathcal{R}(\mathbf{A}) := \{y \in \mathbb{R}^m; y = \mathbf{A}x, \text{ para algum } x \in \mathbb{R}^n\}$. O *espaço nulo* de \mathbf{A} é o subconjunto de \mathbb{R}^n definido por $\mathcal{N}(\mathbf{A}) := \{x \in \mathbb{R}^n; \mathbf{A}x = 0\}$.

Seja $\mathbf{M} \in \mathbb{M}_{m+n}(\mathbb{C})$ definida como

$$\mathbf{M} = \begin{bmatrix} \mathbf{B} & \mathbf{C} \\ \mathbf{F} & \mathbf{G} \end{bmatrix},$$

onde $\mathbf{B} \in \mathbb{M}_n(\mathbb{C})$ e $\mathbf{G} \in \mathbb{M}_m(\mathbb{C})$ são não singulares e $\mathbf{F}, \mathbf{C}^\top \in \mathbb{M}_{m \times n}(\mathbb{C})$.

Lema 2.1. *Se \mathbf{M} e \mathbf{B} são não singulares, então $\mathbf{S} = \mathbf{G} - \mathbf{F}\mathbf{B}^{-1}\mathbf{C}$ (o Complemento de Schur de \mathbf{B} em \mathbf{M}) também o é.*

Demonstração. Usando a ideia da eliminação gaussiana em blocos¹, existe uma única matriz \mathbf{P} tal que

$$\begin{bmatrix} \mathbf{I}_n & \mathbf{0} \\ -\mathbf{P} & \mathbf{I}_m \end{bmatrix} \begin{bmatrix} \mathbf{B} & \mathbf{C} \\ \mathbf{F} & \mathbf{G} \end{bmatrix} = \begin{bmatrix} \mathbf{B} & \mathbf{C} \\ \mathbf{0} & \mathbf{S} \end{bmatrix}. \quad (2.1)$$

Como \mathbf{B} é não singular, sua inversa é unicamente dada por \mathbf{B}^{-1} . Logo, tomando $\mathbf{P} = \mathbf{FB}^{-1}$ a equação (2.1) é válida com $\mathbf{S} = \mathbf{G} - \mathbf{FB}^{-1}\mathbf{C}$. Daí, como \mathbf{M} é não singular, denotando

$$\mathbf{N} = \begin{bmatrix} \mathbf{I}_n & \mathbf{0} \\ -\mathbf{P} & \mathbf{I}_m \end{bmatrix} \quad \text{e} \quad \mathbf{H} = \begin{bmatrix} \mathbf{B} & \mathbf{C} \\ \mathbf{0} & \mathbf{S} \end{bmatrix},$$

segue que $\det(\mathbf{H}) = \det(\mathbf{N})\det(\mathbf{M})$, ou seja, $\det(\mathbf{H}) = \det(\mathbf{M})$, o que implica que \mathbf{H} é não singular. Como $\det(\mathbf{H}) = \det(\mathbf{B})\det(\mathbf{S})$, concluímos que \mathbf{S} é não singular. \square

Proposição 2.1. *Se \mathbf{M} é não singular, sua inversa é dada por*

$$\mathbf{M}^{-1} = \begin{bmatrix} \mathbf{B}^{-1} + \mathbf{B}^{-1}\mathbf{CS}^{-1}\mathbf{FB}^{-1} & -\mathbf{B}^{-1}\mathbf{CS}^{-1} \\ -\mathbf{S}^{-1}\mathbf{FB}^{-1} & \mathbf{S}^{-1} \end{bmatrix},$$

onde $\mathbf{S} = \mathbf{G} - \mathbf{FB}^{-1}\mathbf{C}$ é o Complemento de Schur de \mathbf{B} em \mathbf{M} .

Demonstração. Pelo Lema 2.1, \mathbf{S} é não singular. Reescrevendo

$$\mathbf{M} = \begin{bmatrix} \mathbf{I}_n & \mathbf{0} \\ \mathbf{FB}^{-1} & \mathbf{I}_m \end{bmatrix} \begin{bmatrix} \mathbf{B} & \mathbf{0} \\ \mathbf{0} & \mathbf{S} \end{bmatrix} \begin{bmatrix} \mathbf{I}_n & \mathbf{B}^{-1}\mathbf{C} \\ \mathbf{0} & \mathbf{I}_m \end{bmatrix}$$

segue que

$$\begin{aligned} \mathbf{M}^{-1} &= \begin{bmatrix} \mathbf{I}_n & -\mathbf{B}^{-1}\mathbf{C} \\ \mathbf{0} & \mathbf{I}_m \end{bmatrix} \begin{bmatrix} \mathbf{B}^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{S}^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{I}_n & \mathbf{0} \\ -\mathbf{FB}^{-1} & \mathbf{I}_m \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{B}^{-1} + \mathbf{B}^{-1}\mathbf{CS}^{-1}\mathbf{FB}^{-1} & -\mathbf{B}^{-1}\mathbf{CS}^{-1} \\ -\mathbf{S}^{-1}\mathbf{FB}^{-1} & \mathbf{S}^{-1} \end{bmatrix}. \end{aligned}$$

\square

2.2 Precondicionamento de sistemas lineares

O interesse geral neste trabalho é resolver o problema de programação linear usando um método de pontos interiores primal-dual, mais especificamente o método Preditor-Corretor. Vimos que tais métodos baseiam-se na aplicação do método de Newton no sistema não linear que define as condições de otimalidade para o par de problemas primal-dual referente ao problema em questão. Obtivemos, assim, um novo sistema linear que, por sua vez, foi reduzido ao sistema aumentado e, por fim, no complemento de

¹ A eliminação gaussiana em blocos pode ser encontrada em [Watkins \(2010, p. 93\)](#).

Schur. Finalmente, resta-nos resolver um sistema linear, seja ele com a matriz do sistema aumentado ou do complemento de Schur. A matriz do sistema aumentado tem uma estrutura de grande porte (maior que a do complemento de Schur) além de ser mais esparsa. Uma matriz é dita esparsa quando possui grande quantidade de elementos nulos.

A teoria de Álgebra Linear Computacional mostra que resolver um sistema linear com fatorações diretas, tais como fatorações ortogonais, decomposição de Cholesky e fatoração LU, são demasiadamente caras (do ponto de vista do esforço computacional necessário para obter tais fatores) para muitos problemas de grande porte. Além disso, a estrutura de esparsidade (importante propriedade que torna os métodos de solução mais rápidos do ponto de vista computacional) geralmente é afetada quando usamos métodos diretos.

Uma abordagem alternativa é utilizar métodos iterativos para resolver sistemas lineares, como o método dos Gradientes Conjugados e suas variações, métodos de mínimo resíduo e suas variações, dentre outros. Métodos iterativos geram uma sequência de soluções aproximadas que convergem para a solução do sistema linear e, tipicamente, a matriz de coeficientes é envolvida somente no contexto de multiplicações matriz-vetor, operação mais barata que uma decomposição. Contudo, a razão de convergência dos métodos iterativos é dependente das propriedades espectrais da matriz em questão. Além disso, matrizes mal condicionadas tornam os métodos de solução mais lentos e, eventualmente, podem não convergir (BARRETT et al., 1994; OLIVEIRA, 1997).

Métodos de *splitting*, ou seja, que realizam uma decomposição de uma dada matriz \mathbf{A} fazendo $\mathbf{A} = \mathbf{C} - \mathbf{R}$ (onde \mathbf{C} é não singular) podem promover uma melhor distribuição dos autovalores da matriz de iteração $\mathbf{C}^{-1}\mathbf{A}$, de tal forma que os métodos iterativos convergem com maior velocidade do que sem *splitting*. Por esta razão, a matriz \mathbf{C} tem sido chamada de *matriz pré-condicionadora* (AXELSSON, 1994).

Dado um sistema linear $\mathbf{A}\mathbf{x} = \mathbf{b}$, resolvemos um sistema linear equivalente $\mathbf{M}^{-1}\mathbf{A}\mathbf{N}^{-1}\tilde{\mathbf{x}} = \tilde{\mathbf{b}}$, onde $\tilde{\mathbf{x}} = \mathbf{N}\mathbf{x}$ e $\tilde{\mathbf{b}} = \mathbf{M}^{-1}\mathbf{b}$. Dizemos que o sistema está pré-condicionado e que $\mathbf{M}^{-1}\mathbf{A}\mathbf{N}^{-1}$ é a matriz pré-condicionada. Um pré-condicionador é simétrico quando $\mathbf{N}^T = \mathbf{M}$, pois se a matriz \mathbf{A} é simétrica então a matriz pré-condicionada $\mathbf{M}^{-1}\mathbf{A}\mathbf{N}^{-T}$ também deve ser simétrica. Neste caso, devido à Lei da Inércia de Sylvester², a matriz pré-condicionada possui o mesmo número de autovalores positivos (ou negativos) que a matriz original. Desta forma, pré-condicionadores simétricos não podem ser usados para transformar um sistema linear indefinido em outro definido positivo (OLIVEIRA, 1997).

A tarefa de encontrar um pré-condicionador efetivo pode ser bastante complicada, devido às suas características desejáveis. Como queremos transformar o sistema linear originalmente mal condicionado em outro bem condicionado, idealmente a matriz

² A Lei da Inércia de Sylvester pode ser encontrada em Golub e Van Loan (2013, p. 448).

pré-condicionada $\mathbf{M}^{-1}\mathbf{A}\mathbf{N}^{-1}$ deve ser mais próxima da matriz identidade que a matriz \mathbf{A} , no sentido que seu número de condição deve ser mais próximo de 1 do que a matriz \mathbf{A} . Podemos, ainda, considerar o objetivo de tornar os autovalores da matriz mais próximos de ± 1 , o que melhora o espectro da matriz e acelera a convergência dos métodos iterativos. Além disso, os pré-condicionadores \mathbf{M} e \mathbf{N} devem ser tais que os sistemas lineares que os envolvem sejam mais fáceis de resolver. Contudo, a economia no número de iterações é superada pela despesa na resolução de sistemas lineares. Por conta disso, os pré-condicionadores, geralmente, são matrizes diagonais em blocos ou triangulares em blocos. Por todas estas questões levantadas aqui, pré-condicionadores são altamente dependentes do problema a ser resolvido e, desta forma, não existe uma “receita” para construir um bom pré-condicionador (OLIVEIRA, 1997).

Diferentes classes de métodos de pré-condicionamento existem. Uma das formas mais naturais de definir um pré-condicionador é realizar uma fatoração incompleta da matriz \mathbf{A} . A ideia das fatorações incompletas é rejeitar o preenchimento de certas entradas (das matrizes que compõem os fatores obtidos da decomposição) que ocorrem em posições fora de um padrão de esparsidade escolhido, ou que são pequenas em relação a algumas entradas da diagonal. Quando a matriz \mathbf{A} é simétrica e definida positiva, \mathbf{C} pode ser obtida de modo a ser também simétrica e definida positiva e, assim, o espectro de $\mathbf{C}^{-1}\mathbf{A}$ permanece positivo. A eficácia de um pré-condicionador vai depender de quão bem \mathbf{C}^{-1} se aproxima de \mathbf{A}^{-1} . Contudo, fatorações incompletas podem nos levar a uma aproximação grosseira de \mathbf{A} e fazer o método requerer mais iterações para convergir para uma solução. Fatorações mais precisas requerem menos iterações para convergir, mas o custo para obter tais fatores aumenta bastante. Se vários sistemas com a mesma matriz devem ser resolvidos, o custo pode ser amortizado (AXELSSON, 1994; BARRETT et al., 1994; SAAD, 2003).

2.3 Propriedades do sistema aumentado

Para os resultados que serão discutidos nesta seção e nas seções seguintes deste capítulo, consideraremos as matrizes $\mathbf{A} \in \mathbb{M}_{m \times n}(\mathbb{R})$ com $n > m$, $\text{posto}(\mathbf{A}) = m$ e $\mathbf{D} \in \mathbb{M}_n(\mathbb{R})$ uma matriz diagonal com $d_{ii} > 0$, $i = 1, \dots, n$.

Lema 2.2. *A matriz $\mathbf{A}\mathbf{D}^{-1}\mathbf{A}^\top$ é definida positiva.*

Demonstração. Como $d_{ii} > 0$, temos que $d_{ii}^{-1} > 0$, o que implica que \mathbf{D} e \mathbf{D}^{-1} são definidas positivas. Seja $u \in \mathbb{R}^m$ um vetor não nulo. Como $\text{posto}(\mathbf{A}) = m$, segue que $\mathbf{A}^\top u \neq 0$. Logo, temos que $u^\top \mathbf{A}\mathbf{D}^{-1}\mathbf{A}^\top u = (\mathbf{A}^\top u)^\top \mathbf{D}^{-1}(\mathbf{A}^\top u) > 0$. Assim, $\mathbf{A}\mathbf{D}^{-1}\mathbf{A}^\top$ é definida positiva. \square

Lema 2.3. *A matriz $\mathbf{A}\mathbf{D}^{-1}\mathbf{A}^\top + \mathbf{E}$ é definida positiva, onde $\mathbf{E} \in \mathbb{M}_m(\mathbb{R})$ é uma matriz diagonal com $e_{ii} \geq 0$, $i = 1, \dots, m$.*

Demonstração. Seja $u \in \mathbb{R}^m$ um vetor não nulo. Usando o Lema 2.2 e a definição da matriz \mathbf{E} , segue que $u^\top (\mathbf{A}\mathbf{D}^{-1}\mathbf{A}^\top + \mathbf{E})u = u^\top \mathbf{A}\mathbf{D}^{-1}\mathbf{A}^\top u + u^\top \mathbf{E}u > 0$. Assim, $\mathbf{A}\mathbf{D}^{-1}\mathbf{A}^\top + \mathbf{E}$ é definida positiva. \square

Proposição 2.2. *A matriz*

$$\begin{bmatrix} -\mathbf{D} & \mathbf{A}^\top \\ \mathbf{A} & \mathbf{0} \end{bmatrix} \quad (2.2)$$

é não singular e indefinida.

Demonstração. Considere $x \in \mathbb{R}^n$, $y \in \mathbb{R}^m$ e o sistema homogêneo abaixo

$$\begin{bmatrix} -\mathbf{D} & \mathbf{A}^\top \\ \mathbf{A} & \mathbf{0} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \Rightarrow \begin{cases} \mathbf{A}^\top y - \mathbf{D}x = 0 \\ \mathbf{A}x = 0 \end{cases}$$

Temos que $x = \mathbf{D}^{-1}\mathbf{A}^\top y$. Substituindo na segunda equação, obtemos $\mathbf{A}\mathbf{D}^{-1}\mathbf{A}^\top y = 0$, que, pelo Lema 2.2, tem solução única $y = \mathbf{0}$. Daí, tem-se que $x = \mathbf{0}$, ou seja, matriz (2.2) é não singular. Agora, usando a relação

$$\begin{bmatrix} \mathbf{I}_n & \mathbf{0} \\ -\mathbf{A}\mathbf{D}^{-1} & \mathbf{I}_m \end{bmatrix} \begin{bmatrix} -\mathbf{D} & \mathbf{0} \\ \mathbf{0} & \mathbf{A}\mathbf{D}^{-1}\mathbf{A}^\top \end{bmatrix} \begin{bmatrix} \mathbf{I}_n & -\mathbf{D}^{-1}\mathbf{A}^\top \\ \mathbf{0} & \mathbf{I}_m \end{bmatrix} = \begin{bmatrix} -\mathbf{D} & \mathbf{A}^\top \\ \mathbf{A} & \mathbf{0} \end{bmatrix}$$

e a Lei da Inércia de Sylvester, segue que as matrizes

$$\begin{bmatrix} -\mathbf{D} & \mathbf{0} \\ \mathbf{0} & \mathbf{A}\mathbf{D}^{-1}\mathbf{A}^\top \end{bmatrix} \quad \text{e} \quad \begin{bmatrix} -\mathbf{D} & \mathbf{A}^\top \\ \mathbf{A} & \mathbf{0} \end{bmatrix}$$

possuem a mesma inércia, isto é, possuem m autovalores positivos e n autovalores negativos. Portanto, a matriz (2.2) é indefinida. \square

Proposição 2.3. *A matriz*

$$\begin{bmatrix} -\mathbf{D} & \mathbf{A}^\top \\ \mathbf{A} & \mathbf{E} \end{bmatrix} \quad (2.3)$$

é não singular e indefinida, onde $\mathbf{E} \in \mathbb{M}_m(\mathbb{R})$ é uma matriz diagonal com $e_{ii} \geq 0$, $i = 1, \dots, m$.

Demonstração. Considere $x \in \mathbb{R}^n$, $y \in \mathbb{R}^m$ e o sistema homogêneo abaixo

$$\begin{bmatrix} -\mathbf{D} & \mathbf{A}^\top \\ \mathbf{A} & \mathbf{E} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \Rightarrow \begin{cases} \mathbf{A}^\top y - \mathbf{D}x = 0 \\ \mathbf{A}x + \mathbf{E}y = 0 \end{cases}$$

Temos que $x = \mathbf{D}^{-1}\mathbf{A}^\top y$. Substituindo na segunda equação, obtemos $\mathbf{A}\mathbf{D}^{-1}\mathbf{A}^\top y + \mathbf{E}y = 0$, ou seja, $(\mathbf{A}\mathbf{D}^{-1}\mathbf{A}^\top + \mathbf{E})y = 0$, que, pelo Lema 2.3, tem solução única $y = 0$. Daí, tem-se que $x = 0$, ou seja, matriz (2.3) é não singular. Agora, usando a relação

$$\begin{bmatrix} \mathbf{I}_n & \mathbf{0} \\ -\mathbf{A}\mathbf{D}^{-1} & \mathbf{I}_m \end{bmatrix} \begin{bmatrix} -\mathbf{D} & \mathbf{0} \\ \mathbf{0} & \mathbf{A}\mathbf{D}^{-1}\mathbf{A}^\top + \mathbf{E} \end{bmatrix} \begin{bmatrix} \mathbf{I}_n & -\mathbf{D}^{-1}\mathbf{A}^\top \\ \mathbf{0} & \mathbf{I}_m \end{bmatrix} = \begin{bmatrix} -\mathbf{D} & \mathbf{A}^\top \\ \mathbf{A} & \mathbf{E} \end{bmatrix}$$

e a Lei da Inércia de Sylvester, segue que as matrizes

$$\begin{bmatrix} -\mathbf{D} & \mathbf{0} \\ \mathbf{0} & \mathbf{A}\mathbf{D}^{-1}\mathbf{A}^\top + \mathbf{E} \end{bmatrix} \quad \text{e} \quad \begin{bmatrix} -\mathbf{D} & \mathbf{A}^\top \\ \mathbf{A} & \mathbf{0} \end{bmatrix}$$

possuem a mesma inércia, isto é, possuem m autovalores positivos e n autovalores negativos. Portanto, a matriz (2.3) é indefinida. \square

Proposição 2.4. *A inversa da matriz (2.2) é dada por*

$$\begin{bmatrix} -\mathbf{D} & \mathbf{A}^\top \\ \mathbf{A} & \mathbf{0} \end{bmatrix}^{-1} = \begin{bmatrix} -\mathbf{D}^{-1} + \mathbf{D}^{-1}\mathbf{A}^\top\mathbf{Z}^{-1}\mathbf{A}\mathbf{D}^{-1} & \mathbf{D}^{-1}\mathbf{A}^\top\mathbf{Z}^{-1} \\ \mathbf{Z}^{-1}\mathbf{A}\mathbf{D}^{-1} & \mathbf{Z}^{-1} \end{bmatrix}, \quad (2.4)$$

onde $\mathbf{Z} = \mathbf{A}\mathbf{D}^{-1}\mathbf{A}^\top$ é o Complemento de Schur de \mathbf{D} em (2.2).

Demonstração. Da Proposição 2.2, a matriz (2.2) é não singular. Logo, basta aplicar a Proposição 2.1. \square

Proposição 2.5. *A inversa da matriz (2.3) é dada por*

$$\begin{bmatrix} -\mathbf{D} & \mathbf{A}^\top \\ \mathbf{A} & \mathbf{E} \end{bmatrix}^{-1} = \begin{bmatrix} -\mathbf{D}^{-1} + \mathbf{D}^{-1}\mathbf{A}^\top\mathbf{Z}^{-1}\mathbf{A}\mathbf{D}^{-1} & \mathbf{D}^{-1}\mathbf{A}^\top\mathbf{Z}^{-1} \\ \mathbf{Z}^{-1}\mathbf{A}\mathbf{D}^{-1} & \mathbf{Z}^{-1} \end{bmatrix}, \quad (2.5)$$

onde $\mathbf{Z} = \mathbf{A}\mathbf{D}^{-1}\mathbf{A}^\top + \mathbf{E}$ é o Complemento de Schur de \mathbf{D} em (2.3).

Demonstração. Da Proposição 2.3, a matriz (2.3) é não singular. Logo, basta aplicar a Proposição 2.1. \square

Os resultados acima mencionados mostram que a matriz do sistema aumentado é indefinida, não singular e possui inversa dada por (2.4) ou (2.5), a depender da complexidade da formulação do problema de programação linear. Além disso, mostram que a matriz do complemento de Schur é definida positiva. Na seção 2.4, exibiremos um *pré-condicionador* baseado na inversa do sistema aumentado.

2.4 Pré-condicionadores no contexto do sistema aumentado e do complemento de Schur

Uma desvantagem em usar o complemento de Schur para resolver o problema de programação linear é que a estrutura de esparsidade inerente à matriz do sistema aumentado é perdida, o que pode ser desvantajoso e complicado para alguns problemas. Além disso, informações consideravelmente importantes podem ser perdidas ao realizar a redução. Em várias implementações conhecidas de métodos iterativos, o cálculo do complemento de Schur é necessário para construir pré-condicionadores. Contudo, nenhuma informação é perdida ao considerarmos o sistema aumentado, que contém o complemento de Schur em algum sentido (OLIVEIRA, 1997).

Lema 2.4. *Oliveira e Sorensen (2005, p. 5). Seja*

$$\begin{bmatrix} -\mathbf{D} & \mathbf{A}^\top \\ \mathbf{A} & \mathbf{E} \end{bmatrix}$$

não singular e \mathbf{D} simétrica definida positiva. Então, existe um par de pré-condicionadores \mathbf{M} e \mathbf{N} tais que esta matriz pode ser reduzida como

$$\mathbf{M}^{-1} \begin{bmatrix} -\mathbf{D} & \mathbf{A}^\top \\ \mathbf{A} & \mathbf{E} \end{bmatrix} \mathbf{N}^{-1} = \begin{bmatrix} \mathbf{I}_n & \mathbf{0} \\ \mathbf{0} & \mathbf{JZH}^\top \end{bmatrix},$$

onde \mathbf{Z} é o complemento de Schur e \mathbf{J} e \mathbf{H} são matrizes não singulares escolhidas arbitrariamente (de dimensões apropriadas).

Demonstração. Como \mathbf{D} é simétrica definida positiva, consideremos sua fatoração de Cholesky $\mathbf{D} = \mathbf{LL}^\top$. Considere, ainda

$$\mathbf{M}^{-1} = \begin{bmatrix} -\mathbf{L}^{-1} & \mathbf{0} \\ \mathbf{JAD}^{-1} & \mathbf{J} \end{bmatrix} \quad \text{e} \quad \mathbf{N}^{-1} = \begin{bmatrix} \mathbf{L}^{-\top} & \mathbf{D}^{-1}\mathbf{A}^\top\mathbf{H}^\top \\ \mathbf{0} & \mathbf{H}^\top \end{bmatrix}.$$

Então

$$\mathbf{M}^{-1} \begin{bmatrix} -\mathbf{D} & \mathbf{A}^\top \\ \mathbf{A} & \mathbf{E} \end{bmatrix} \mathbf{N}^{-1} = \begin{bmatrix} \mathbf{I}_n & \mathbf{0} \\ \mathbf{0} & \mathbf{JZH}^\top \end{bmatrix},$$

onde $\mathbf{Z} = \mathbf{AD}^{-1}\mathbf{A}^\top + \mathbf{E}$. □

Lema 2.5. *Oliveira e Sorensen (2005, p. 5). Considere o sistema linear dado por*

$$\begin{bmatrix} -\mathbf{D} & \mathbf{A}^\top \\ \mathbf{A} & \mathbf{E} \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} r_1 \\ r_2 \end{bmatrix},$$

onde $\mathbf{E} \in \mathbb{M}_m(\mathbb{R})$ é uma matriz diagonal com $e_{ii} \geq 0$, $i = 1, \dots, m$, e seu complemento de Schur $\mathbf{Z} = \mathbf{AD}^{-1}\mathbf{A}^\top + \mathbf{E}$, onde \mathbf{D} é não singular e $\text{posto}(\mathbf{A}) = m$. Então, qualquer pré-condicionador triangular em blocos

$$\begin{bmatrix} \mathbf{H} & \mathbf{0} \\ \mathbf{F} & \mathbf{J} \end{bmatrix},$$

com $\mathbf{H} \in \mathbb{M}_m(\mathbb{R})$ de posto completo, $\mathbf{J} \in \mathbb{M}_n(\mathbb{R})$ de posto completo e $\mathbf{F} \in \mathbb{M}_{m \times n}(\mathbb{R})$ com $\text{posto}(\mathbf{F}) = m$, fornece um sistema pré-condicionado para \mathbf{JZJ}^\top , independente de \mathbf{H} e \mathbf{F} .

Demonstração. Considere a classe de pré-condicionadores simétricos

$$\begin{bmatrix} \mathbf{H} & \mathbf{0} \\ \mathbf{F} & \mathbf{J} \end{bmatrix} \begin{bmatrix} -\mathbf{D} & \mathbf{A}^\top \\ \mathbf{A} & \mathbf{E} \end{bmatrix} \begin{bmatrix} \mathbf{H}^\top & \mathbf{F}^\top \\ \mathbf{0} & \mathbf{J}^\top \end{bmatrix} = \begin{bmatrix} -\mathbf{HDH}^\top & \mathbf{B}^\top \\ \mathbf{B} & \mathbf{C} \end{bmatrix},$$

onde $\mathbf{B} = -\mathbf{F}\mathbf{D}\mathbf{H}^\top + \mathbf{J}\mathbf{A}\mathbf{H}^\top$ e $\mathbf{C} = -\mathbf{F}\mathbf{D}\mathbf{F}^\top + \mathbf{F}\mathbf{A}^\top\mathbf{J}^\top + \mathbf{J}\mathbf{A}\mathbf{F}^\top + \mathbf{J}\mathbf{E}\mathbf{J}^\top$. Então, o sistema pré-condicionado será como segue

$$\begin{bmatrix} -\mathbf{H}\mathbf{D}\mathbf{H}^\top & \mathbf{B}^\top \\ \mathbf{B} & \mathbf{C} \end{bmatrix} \begin{bmatrix} \Delta\tilde{x} \\ \Delta\tilde{y} \end{bmatrix} = \begin{bmatrix} \mathbf{H}r_1 \\ \mathbf{J}r_2 + \mathbf{F}r_1 \end{bmatrix},$$

onde $\Delta\tilde{x} = \mathbf{H}^{-\top}(\Delta x - \mathbf{F}^\top\mathbf{J}^{-\top}\Delta y)$ e $\Delta\tilde{y} = \mathbf{J}^{-\top}\Delta y$. Eliminando $\Delta\tilde{x}$ obtemos

$$\mathbf{J}\mathbf{Z}\mathbf{J}^\top\Delta\tilde{y} = \mathbf{J}r_2 + \mathbf{A}\mathbf{D}^{-1}r_1 \quad (2.6)$$

que, por sua vez, não possui dependências sobre \mathbf{F} ou \mathbf{H} . Portanto qualquer escolha para estas matrizes (que preserve a não singularidade) são pré-condicionadores válidos que nos levam ao sistema (2.6). \square

De acordo com Oliveira e Sorensen (2005, p. 4), os resultados acima justificam o fato de considerarmos o uso do sistema aumentado em lugar do complemento de Schur, quando estamos interessados em desenvolver pré-condicionadores no contexto da solução iterativa dos sistemas lineares oriundos do método de pontos interiores. Estes resultados mostram que todo pré-condicionador para o complemento de Schur pode ser replicado para o sistema aumentado, mas a recíproca não é verdadeira. Portanto, o sistema do complemento de Schur pode ser visto como caso particular contido em uma classe mais geral de pré-condicionadores para o sistema aumentado.

2.5 Uma classe de pré-condicionadores para o método de pontos interiores

No trabalho desenvolvido por Oliveira e Sorensen (2005), uma nova classe de pré-condicionadores foi desenvolvida e seu bom desempenho foi exibido. A seguir, as ideias da construção de um pré-condicionador serão abordadas, bem como as propriedades teóricas da classe desenvolvida.

2.5.1 Construção de um pré-condicionador

Iniciando com um pré-condicionador genérico, particionado em blocos, temos

$$\mathbf{M}^{-1} = \mathbf{N}^{-\top} = \begin{bmatrix} \mathbf{F} & \mathbf{J} \\ \mathbf{H} & \mathbf{K} \end{bmatrix},$$

onde $\mathbf{F} \in \mathbb{M}_n(\mathbb{R})$, $\mathbf{K} \in \mathbb{M}_m(\mathbb{R})$ e $\mathbf{J}^\top, \mathbf{H} \in \mathbb{M}_{m \times n}(\mathbb{R})$, com $\text{posto}(\mathbf{H}) = m = \text{posto}(\mathbf{J}^\top)$, de modo a ser um pré-condicionador compatível com a matriz do sistema aumentado definida em (1.23). O sistema aumentado pré-condicionado fica, então

$$\mathbf{M}^{-1} \begin{bmatrix} -\mathbf{D} & \mathbf{A}^\top \\ \mathbf{A} & \mathbf{0} \end{bmatrix} \mathbf{N}^{-1} = \begin{bmatrix} \mathbf{F} & \mathbf{J} \\ \mathbf{H} & \mathbf{K} \end{bmatrix} \begin{bmatrix} -\mathbf{D} & \mathbf{A}^\top \\ \mathbf{A} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{F}^\top & \mathbf{H}^\top \\ \mathbf{J}^\top & \mathbf{K}^\top \end{bmatrix},$$

ou seja,

$$\begin{bmatrix} -\mathbf{FDF}^\top + \mathbf{JAF}^\top + \mathbf{FA}^\top \mathbf{J}^\top & -\mathbf{FDH}^\top + \mathbf{JAH}^\top + \mathbf{FA}^\top \mathbf{K}^\top \\ -\mathbf{HDF}^\top + \mathbf{KAF}^\top + \mathbf{HA}^\top \mathbf{J}^\top & -\mathbf{HDH}^\top + \mathbf{KAH}^\top + \mathbf{HA}^\top \mathbf{K}^\top \end{bmatrix}. \quad (2.7)$$

Observe que a matriz pré-condicionada é simétrica.

A partir deste ponto, iniciou-se um processo de decisões para construção do pré-condicionador, de forma a evitar a formação do complemento de Schur. O bloco inferior direito foi o guia inicial, uma vez que, para muitas escolhas de \mathbf{K} e \mathbf{H} , o complemento de Schur aparece. Desta forma, escolheu-se $\mathbf{K} = \mathbf{0}$, ficando a matriz pré-condicionada do sistema aumentado da seguinte forma

$$\begin{bmatrix} -\mathbf{FDF}^\top + \mathbf{JAF}^\top + \mathbf{FA}^\top \mathbf{J}^\top & -\mathbf{FDH}^\top + \mathbf{JAH}^\top \\ -\mathbf{HDF}^\top + \mathbf{HA}^\top \mathbf{J}^\top & -\mathbf{HDH}^\top \end{bmatrix}.$$

Contudo, com a matriz pré-condicionadora da forma

$$\begin{bmatrix} \mathbf{F} & \mathbf{J} \\ \mathbf{H} & \mathbf{0} \end{bmatrix}$$

torna-se necessário garantir que seja não singular. Em poucos passos podemos observar que se a interseção dos espaços nulos das matrizes \mathbf{F} e \mathbf{J} contém um vetor não nulo, então a matriz pré-condicionadora acima é singular. Como a matriz \mathbf{J}^\top é retangular, $n \times m$, $\text{posto}(\mathbf{J}) = m$ e $n > m$, então \mathbf{J}^\top tem espaço nulo de dimensão até $n - m$ (Teorema do Núcleo e Imagem). Além disso, a escolha $\mathbf{F}^\top = \mathbf{D}^{-1} \mathbf{A}^\top \mathbf{J}^\top$ não é razoável, pois $\mathcal{N}(\mathbf{F}^\top) \subset \mathcal{N}(\mathbf{J}^\top)$, apesar de proporcionar uma matriz pré-condicionada diagonal em blocos. Uma alternativa de proporcionar isto sem, contudo, construir um pré-condicionador singular é escolher $\mathbf{J}^\top = (\mathbf{HA}^\top)^{-1} \mathbf{HDF}^\top$, que fornece uma matriz pré-condicionada como

$$\begin{bmatrix} -\mathbf{FDF}^\top + \mathbf{JAF}^\top + \mathbf{FA}^\top \mathbf{J}^\top & \mathbf{0} \\ \mathbf{0} & -\mathbf{HDH}^\top \end{bmatrix}.$$

A escolha da matriz \mathbf{H} trouxe de volta a preocupação com a formação de uma matriz como a do complemento de Schur, o que fez com que se descartassem possibilidades como \mathbf{A} , \mathbf{AD}^{-1} e variações. A escolha do autor caiu sobre $\mathbf{H} = [\mathbf{I}_m \ \mathbf{0}] \mathbf{P}$ onde $\mathbf{P} \in \mathbf{M}_n(\mathbb{R})$ é uma matriz de permutação tal que \mathbf{HA}^\top é não singular. Desta forma, o bloco inferior direito da matriz pré-condicionada fica

$$-\mathbf{HDH}^\top = -[\mathbf{I}_m \ \mathbf{0}] \mathbf{PDP}^\top \begin{bmatrix} \mathbf{I}_m \\ \mathbf{0} \end{bmatrix} = -[\mathbf{I}_m \ \mathbf{0}] \begin{bmatrix} \mathbf{D}_B & \mathbf{0} \\ \mathbf{0} & \mathbf{D}_N \end{bmatrix} \begin{bmatrix} \mathbf{I}_m \\ \mathbf{0} \end{bmatrix} = -\mathbf{D}_B,$$

onde usamos que

$$\mathbf{PDP}^\top = \begin{bmatrix} \mathbf{D}_B & \mathbf{0} \\ \mathbf{0} & \mathbf{D}_N \end{bmatrix},$$

com $\mathbf{D}_B \in \mathbb{M}_m(\mathbb{R})$ e $\mathbf{D}_N \in \mathbb{M}_{n-m}(\mathbb{R})$ são matrizes diagonais.

Sendo assim, já obtemos as escolhas para \mathbf{K} , \mathbf{J} e \mathbf{H} para a matriz pré-condicionadora. Resta decidir sobre \mathbf{F} . O autor optou por fazer $\mathbf{F} = \mathbf{D}^{-\frac{1}{2}}$. Obteve-se, assim, a matriz pré-condicionada

$$\begin{bmatrix} -\mathbf{I}_n + \mathbf{D}^{-\frac{1}{2}} \mathbf{A}^\top \mathbf{J}^\top + \mathbf{J} \mathbf{A} \mathbf{D}^{-\frac{1}{2}} & \mathbf{0} \\ \mathbf{0} & -\mathbf{D}_B \end{bmatrix} \quad (2.8)$$

e uma matriz pré-condicionadora dada por

$$\mathbf{M}^{-1} = \mathbf{N}^{-\top} = \begin{bmatrix} \mathbf{D}^{-\frac{1}{2}} & \mathbf{J} \\ \mathbf{H} & \mathbf{0} \end{bmatrix},$$

onde $\mathbf{J} = \mathbf{H}^\top \mathbf{D}_B^{\frac{1}{2}} \mathbf{B}^{-1}$, $\mathbf{H} \mathbf{P}^\top = [\mathbf{I}_m \mathbf{0}]$ e $\mathbf{A} \mathbf{P}^\top = [\mathbf{B} \mathbf{N}]$. O autor ainda alerta que a notação $\mathbf{A} \mathbf{P}^\top = [\mathbf{B} \mathbf{N}]$ indica que o produto $\mathbf{A} \mathbf{P}^\top$ forma uma partição onde a matriz \mathbf{B} é uma matriz não singular.

Podemos observar que o objetivo de construir um pré-condicionador que evita a formação do complemento de Schur foi atingido ao preço de resolver sistemas lineares com a matriz \mathbf{B} , que deve ser obtida de \mathbf{A} .

Conforme observa Oliveira (1997, p. 60), uma fatoraçoão do tipo $\mathbf{QB} = \mathbf{LU}$ é tipicamente mais fácil de obter que a fatoraçoão de Cholesky, uma vez que uma seleção das colunas de \mathbf{B} não causa mudança na estrutura do problema. Além disso, a fatoraçoão de \mathbf{B} não depende de \mathbf{D} , que é a única alteração na matriz do sistema aumentado de uma iteraçoão para outra. Mantendo a matriz \mathbf{P} inalterada, o próximo pré-condicionador irá requerer o cálculo de $\mathbf{D}^{-\frac{1}{2}}$ a partir de \mathbf{D}^{-1} , ou seja, calcular n raízes quadradas.

A Proposição 2.2 mostra que a matriz do sistema aumentado possui m autovalores positivos e n autovalores negativos. Esta inércia deve ser preservada na matriz pré-condicionada, conforme observamos anteriormente.

Lema 2.6. *Seja $\mathbf{A} = [\mathbf{B} \mathbf{N}]$ com \mathbf{B} não singular e $\mathbf{J} = \mathbf{H}^\top \mathbf{D}_B^{\frac{1}{2}} \mathbf{B}^{-1}$ onde $\mathbf{H} = [\mathbf{I}_m \mathbf{0}]$. Então $\mathbf{J}^\top \mathbf{D}^{-\frac{1}{2}} \mathbf{A}^\top = \mathbf{A} \mathbf{D}^{-\frac{1}{2}} \mathbf{J} = \mathbf{I}_m$.*

Demonstração. Vemos que

$$\mathbf{J}^\top \mathbf{D}^{-\frac{1}{2}} \mathbf{A}^\top = \mathbf{B}^{-\top} \mathbf{D}_B^{\frac{1}{2}} \mathbf{H} \mathbf{D}^{-\frac{1}{2}} \mathbf{A}^\top = \mathbf{A} \mathbf{D}^{-\frac{1}{2}} \mathbf{H}^\top \mathbf{D}_B^{\frac{1}{2}} \mathbf{B}^{-1} = \mathbf{A} \mathbf{D}^{-\frac{1}{2}} \mathbf{J}.$$

Agora, vamos mostrar que $\mathbf{J}^\top \mathbf{D}^{-\frac{1}{2}} \mathbf{A}^\top = \mathbf{I}_m$.

$$\begin{aligned} \mathbf{B}^{-\top} \mathbf{D}_B^{\frac{1}{2}} [\mathbf{I} \mathbf{0}] \mathbf{D}^{-\frac{1}{2}} [\mathbf{B} \mathbf{N}]^\top &= \\ \mathbf{B}^{-\top} \mathbf{D}_B^{\frac{1}{2}} [\mathbf{D}_B^{-\frac{1}{2}} \mathbf{0}] [\mathbf{B} \mathbf{N}]^\top &= \\ \mathbf{B}^{-\top} [\mathbf{I} \mathbf{0}] [\mathbf{B} \mathbf{N}]^\top &= \mathbf{I}_m \end{aligned}$$

□

Este resultado também pode ser estendido à matriz $\mathbf{A} = [\mathbf{B} \mathbf{N}] \mathbf{P}$.

2.5.2 Propriedades do sistema pré-condicionado

O sistema aumentado pré-condicionado da forma como foi abordada anteriormente pertence a uma classe de matrizes mais genérica, que pode ser representada como

$$\mathbf{L} = -\mathbf{I}_n + \mathbf{U}^\top \mathbf{V}^\top + \mathbf{V}\mathbf{U}, \quad (2.9)$$

onde $\mathbf{U}\mathbf{V} = \mathbf{V}^\top \mathbf{U}^\top = \mathbf{I}_m$, com $\mathbf{U}, \mathbf{V}^\top \in \mathbb{M}_{m \times n}(\mathbb{R})$.

Teorema 2.1. *Seja λ um autovalor de \mathbf{L} dada por (2.9) onde $\mathbf{U}, \mathbf{V}^\top \in \mathbb{M}_{m \times n}(\mathbb{R})$. Então $|\lambda| \geq 1$.*

Demonstração. Seja v um autovetor normalizado associado ao autovalor λ . Então

$$\begin{aligned} \mathbf{L}v &= \lambda v \\ \mathbf{L}^2v &= \lambda^2 v \\ (\mathbf{I}_n - \mathbf{U}^\top \mathbf{V}^\top - \mathbf{V}\mathbf{U} + \mathbf{U}^\top \mathbf{V}^\top \mathbf{V}\mathbf{U} + \mathbf{V}\mathbf{U}\mathbf{U}^\top \mathbf{V}^\top)v &= \lambda^2 v \\ v + (\mathbf{U}^\top \mathbf{V}^\top - \mathbf{V}\mathbf{U})(\mathbf{V}\mathbf{U} - \mathbf{U}^\top \mathbf{V}^\top)v &= \lambda^2 v. \end{aligned}$$

Multiplicando à direita por v^\top obtemos

$$\begin{aligned} 1 + v^\top (\mathbf{U}^\top \mathbf{V}^\top - \mathbf{V}\mathbf{U})(\mathbf{V}\mathbf{U} - \mathbf{U}^\top \mathbf{V}^\top) &= \lambda^2 \\ 1 + w^\top w &= \lambda^2, \end{aligned}$$

onde $w = (\mathbf{V}\mathbf{U} - \mathbf{U}^\top \mathbf{V}^\top)v$. Desta forma, obtivemos $\lambda^2 \geq 1$. \square

Pelo teorema acima, podemos ver que a matriz \mathbf{L} é não singular. Além disso, ela não possui autovalores na vizinhança de zero, propriedade desejável para que muitos métodos iterativos apresentem bom desempenho.

Outro fato importante é que, como $\mathcal{R}(\mathbf{V}\mathbf{U}) \subseteq \mathcal{R}(\mathbf{V})$, $\mathbf{U}\mathbf{V} = \mathbf{I}_m$ e $(\mathbf{V}\mathbf{U})^2 = \mathbf{V}\mathbf{U}$, segue que $\mathbf{V}\mathbf{U}$ é uma projeção sobre $\mathcal{R}(\mathbf{V})$.

Teorema 2.2. *A matriz \mathbf{L} em (2.9) onde $\mathbf{U}, \mathbf{V}^\top \in \mathbb{M}_{m \times n}(\mathbb{R})$ possui ao menos um autovalor λ tal que $|\lambda| = 1$.*

Demonstração. Consideremos 3 casos, a saber:

1. $n > 2m$. Como $\mathbf{L} + \mathbf{I}_n = \mathbf{U}^\top \mathbf{V}^\top + \mathbf{V}\mathbf{U}$, segue que $\text{posto}(\mathbf{L} + \mathbf{I}_n) = \text{posto}(\mathbf{U}^\top \mathbf{V}^\top + \mathbf{V}\mathbf{U})$. Além disso, $\text{posto}(\mathbf{U}^\top \mathbf{V}^\top + \mathbf{V}\mathbf{U}) \leq \text{posto}(\mathbf{U}^\top \mathbf{V}^\top) + \text{posto}(\mathbf{V}\mathbf{U})$ e temos de $\mathbf{U}\mathbf{V} = \mathbf{I}_m$ que $\text{posto}(\mathbf{U}) = m = \text{posto}(\mathbf{V}^\top)$. Assim, $\text{posto}(\mathbf{U}^\top \mathbf{V}^\top) = m = \text{posto}(\mathbf{V}\mathbf{U})$. Logo, obtemos $\text{posto}(\mathbf{U}^\top \mathbf{V}^\top + \mathbf{V}\mathbf{U}) \leq 2m < n$. Ou seja, $\text{posto}(\mathbf{L} + \mathbf{I}_n) < n$ e concluímos que $\mathbf{L} + \mathbf{I}_n$ é singular. Sendo $\mathbf{L} + \mathbf{I}_n = \mathbf{U}^\top \mathbf{V}^\top + \mathbf{V}\mathbf{U}$ singular, então $(\mathbf{L} + \mathbf{I}_n)$ tem ao

menos um autovalor nulo. Tendo $\mathbf{L} + \mathbf{I}_n$ um autovalor nulo, segue que \mathbf{L} tem um autovalor $\lambda = -1$.

2. $n < 2m$. Como $\mathbf{U}, \mathbf{V}^\top \in \mathbb{M}_{m \times n}(\mathbb{R})$ segue que \mathbf{U}^\top e \mathbf{V} são transformações lineares $\mathbf{U}^\top, \mathbf{V}: \mathbb{R}^m \rightarrow \mathbb{R}^n$. Daí, temos que $\mathcal{R}(\mathbf{U}^\top) \subseteq \mathbb{R}^n$ e $\mathcal{R}(\mathbf{V}) \subseteq \mathbb{R}^n$, ou seja, temos $\mathcal{R}(\mathbf{U}^\top) \cup \mathcal{R}(\mathbf{V}) \subseteq \mathbb{R}^n$ e, assim, $\dim(\mathcal{R}(\mathbf{U}^\top) + \mathcal{R}(\mathbf{V})) \leq n$. Adicionalmente, temos $\text{posto}(\mathbf{U}^\top) = \text{posto}(\mathbf{V}) = m$, daí

$$\begin{aligned} \dim(\mathcal{R}(\mathbf{U}^\top) \cap \mathcal{R}(\mathbf{V})) &= \dim(\mathcal{R}(\mathbf{U}^\top)) + \dim(\mathcal{R}(\mathbf{V})) - \dim(\mathcal{R}(\mathbf{U}^\top) + \mathcal{R}(\mathbf{V})) \\ &= m + m - \dim(\mathcal{R}(\mathbf{U}^\top) + \mathcal{R}(\mathbf{V})) \\ &\geq 2m - n > 0, \end{aligned}$$

uma vez que $n < 2m$. Então, existe um vetor $v \neq 0 \in (\mathcal{R}(\mathbf{U}^\top) \cap \mathcal{R}(\mathbf{V}))$. Como $v \in \mathcal{R}(\mathbf{U}^\top)$, segue que $\mathbf{U}^\top \mathbf{V}^\top v = v$, já que $\mathbf{U}^\top \mathbf{V}^\top$ é uma projeção. Analogamente, $\mathbf{V} \mathbf{U} v = v$. Portanto, $\mathbf{L} v = (-\mathbf{I}_n + \mathbf{U}^\top \mathbf{V}^\top + \mathbf{V} \mathbf{U}) v = -v + v + v = v$, ou seja, v é um autovetor de \mathbf{L} associado ao autovalor $\lambda = 1$.

3. $n = 2m$. Do caso anterior temos que, se $\mathcal{R}(\mathbf{U}^\top) \cap \mathcal{R}(\mathbf{V}) \neq 0$, então existe um autovalor $\lambda = 1$. Assumindo que $\mathcal{R}(\mathbf{U}^\top) \cap \mathcal{R}(\mathbf{V}) = 0$, segue que existe um autovetor v tal que $v_{\mathcal{R}(\mathbf{U}^\top)} = 0$ ou $v_{\mathcal{R}(\mathbf{V})} = 0$. Sem perda de generalidade, $v_{\mathcal{R}(\mathbf{U}^\top)} = 0$. Então, $v \in \mathcal{N}(\mathbf{U}^\top)$ e existe um autopar $(\theta = \lambda + 1, v)$ onde $\mathbf{V} \mathbf{U} v = \theta v$. Daí, $\lambda = 0$ ou $\lambda = 1$, uma vez que $\mathbf{V} \mathbf{U}$ é um projetor. Contudo, do Teorema 2.1, concluímos que $\lambda = 1$.

Desta forma fica demonstrado o teorema. \square

Segue do teorema acima que o número de condição $\kappa_2(\mathbf{L})$, com \mathbf{L} definida em (2.9), é dado por $\lambda_{\max}(\mathbf{L}) := \max\{|\lambda_i|; \lambda_i \text{ autovalor de } \mathbf{L}\}$.

2.5.3 Redução a sistemas definidos positivos

Consideremos a matriz pré-condicionada do sistema aumentado, dada em (2.8), e seja $\mathbf{W} = -\mathbf{I}_n + \mathbf{D}^{-\frac{1}{2}} \mathbf{A}^\top \mathbf{J}^\top + \mathbf{J} \mathbf{A} \mathbf{D}^{-\frac{1}{2}}$ o bloco superior esquerdo, onde $\mathbf{J} = \mathbf{H}^\top \mathbf{D}_B^{\frac{1}{2}} \mathbf{B}^{-1}$, $\mathbf{H} = [\mathbf{I}_m \ 0] \mathbf{P}$ e $\mathbf{A} = [\mathbf{B} \ \mathbf{N}] \mathbf{P}$. Oliveira e Sorensen (2005, p. 9) afirmam que é possível explorar a estrutura do problema realizando uma redução para um sistema definido positivo. O argumento se baseia na expansão da equação $\mathbf{W} = -\mathbf{I}_n + \mathbf{D}^{-\frac{1}{2}} \mathbf{A}^\top \mathbf{J}^\top + \mathbf{J} \mathbf{A} \mathbf{D}^{-\frac{1}{2}}$.

Realizando as devidas substituições, temos

$$\begin{aligned}
\mathbf{W} &= -\mathbf{I}_n + \mathbf{D}^{-\frac{1}{2}} \mathbf{A}^\top \mathbf{J}^\top + \mathbf{J} \mathbf{A} \mathbf{D}^{-\frac{1}{2}} = \\
&= -\mathbf{I}_n + \mathbf{D}^{-\frac{1}{2}} \mathbf{A}^\top \mathbf{B}^{-\top} \mathbf{D}_B^{\frac{1}{2}} [\mathbf{I}_m \ 0] \mathbf{P} + \mathbf{P}^\top \begin{bmatrix} \mathbf{I}_m \\ 0 \end{bmatrix} \mathbf{D}_B^{\frac{1}{2}} \mathbf{B}^{-1} \mathbf{A} \mathbf{D}^{-\frac{1}{2}} = \\
&= -\mathbf{I}_n + \mathbf{D}^{-\frac{1}{2}} \mathbf{P}^\top \begin{bmatrix} \mathbf{B}^\top \\ \mathbf{N}^\top \end{bmatrix} \mathbf{B}^{-\top} \mathbf{D}_B^{\frac{1}{2}} [\mathbf{I}_m \ 0] \mathbf{P} + \mathbf{P}^\top \begin{bmatrix} \mathbf{I}_m \\ 0 \end{bmatrix} \mathbf{D}_B^{\frac{1}{2}} \mathbf{B}^{-1} [\mathbf{B} \ \mathbf{N}] \mathbf{P} \mathbf{D}^{-\frac{1}{2}} = \\
&= -\mathbf{I}_n + \mathbf{D}^{-\frac{1}{2}} \mathbf{P}^\top \begin{bmatrix} \mathbf{I}_m \\ \mathbf{N}^\top \mathbf{B}^{-\top} \end{bmatrix} [\mathbf{D}_B^{\frac{1}{2}} \ 0] \mathbf{P} + \mathbf{P}^\top \begin{bmatrix} \mathbf{D}_B^{\frac{1}{2}} \\ 0 \end{bmatrix} [\mathbf{I}_m \ \mathbf{B}^{-1} \mathbf{N}] \mathbf{P} \mathbf{D}^{-\frac{1}{2}} = \\
&= -\mathbf{I}_n + \mathbf{D}^{-\frac{1}{2}} \mathbf{P}^\top \begin{bmatrix} \mathbf{D}_B^{\frac{1}{2}} & 0 \\ \mathbf{N}^\top \mathbf{B}^{-\top} \mathbf{D}_B^{\frac{1}{2}} & 0 \end{bmatrix} \mathbf{P} + \mathbf{P}^\top \begin{bmatrix} \mathbf{D}_B^{\frac{1}{2}} & \mathbf{D}_B^{\frac{1}{2}} \mathbf{B}^{-1} \mathbf{N} \\ 0 & 0 \end{bmatrix} \mathbf{P} \mathbf{D}^{-\frac{1}{2}}.
\end{aligned}$$

A última expressão acima ainda pode ser reescrita como

$$-\mathbf{P}^\top \mathbf{P} + \mathbf{P}^\top \begin{bmatrix} \mathbf{D}_B^{-\frac{1}{2}} & 0 \\ 0 & \mathbf{D}_N^{-\frac{1}{2}} \end{bmatrix} \begin{bmatrix} \mathbf{D}_B^{\frac{1}{2}} & 0 \\ \mathbf{N}^\top \mathbf{B}^{-\top} \mathbf{D}_B^{\frac{1}{2}} & 0 \end{bmatrix} \mathbf{P} + \mathbf{P}^\top \begin{bmatrix} \mathbf{D}_B^{\frac{1}{2}} & \mathbf{D}_B^{\frac{1}{2}} \mathbf{B}^{-1} \mathbf{N} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{D}_B^{-\frac{1}{2}} & 0 \\ 0 & \mathbf{D}_N^{-\frac{1}{2}} \end{bmatrix} \mathbf{P},$$

onde foi usado que \mathbf{P} é uma matriz de permutação ($\mathbf{P}^\top \mathbf{P} = \mathbf{I}_n$). Continuando, podemos fazer

$$\mathbf{P}^\top \left(-\mathbf{I}_n + \begin{bmatrix} \mathbf{D}_B^{-\frac{1}{2}} & 0 \\ 0 & \mathbf{D}_N^{-\frac{1}{2}} \end{bmatrix} \begin{bmatrix} \mathbf{D}_B^{\frac{1}{2}} & 0 \\ \mathbf{N}^\top \mathbf{B}^{-\top} \mathbf{D}_B^{\frac{1}{2}} & 0 \end{bmatrix} + \begin{bmatrix} \mathbf{D}_B^{\frac{1}{2}} & \mathbf{D}_B^{\frac{1}{2}} \mathbf{B}^{-1} \mathbf{N} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{D}_B^{-\frac{1}{2}} & 0 \\ 0 & \mathbf{D}_N^{-\frac{1}{2}} \end{bmatrix} \right) \mathbf{P}$$

obtendo, desta última

$$\mathbf{P}^\top \left(-\mathbf{I}_n + \begin{bmatrix} \mathbf{I}_m & 0 \\ \mathbf{D}_N^{-\frac{1}{2}} \mathbf{N}^\top \mathbf{B}^{-\top} \mathbf{D}_B^{\frac{1}{2}} & 0 \end{bmatrix} + \begin{bmatrix} \mathbf{I}_m & \mathbf{D}_B^{\frac{1}{2}} \mathbf{B}^{-1} \mathbf{N} \mathbf{D}_N^{-\frac{1}{2}} \\ 0 & 0 \end{bmatrix} \right) \mathbf{P}$$

chegando, finalmente, no sistema

$$\mathbf{P}^\top \begin{bmatrix} \mathbf{I}_m & \mathbf{D}_B^{\frac{1}{2}} \mathbf{B}^{-1} \mathbf{N} \mathbf{D}_N^{-\frac{1}{2}} \\ \mathbf{D}_N^{-\frac{1}{2}} \mathbf{N}^\top \mathbf{B}^{-\top} \mathbf{D}_B^{\frac{1}{2}} & -\mathbf{I}_{n-m} \end{bmatrix} \mathbf{P} = \mathbf{W}. \quad (2.10)$$

Desta forma, o problema pode ser reduzido a resolver um sistema linear definido positivo envolvendo uma das seguintes matrizes:

$$\mathbf{I}_m + \mathbf{D}_B^{\frac{1}{2}} \mathbf{B}^{-1} \mathbf{N} \mathbf{D}_N^{-1} \mathbf{N}^\top \mathbf{B}^{-\top} \mathbf{D}_B^{\frac{1}{2}} \quad (2.11a)$$

$$\mathbf{I}_{n-m} + \mathbf{D}_N^{-\frac{1}{2}} \mathbf{N}^\top \mathbf{B}^{-\top} \mathbf{D}_B \mathbf{B}^{-1} \mathbf{N} \mathbf{D}_N^{-\frac{1}{2}}. \quad (2.11b)$$

Se denotarmos $\mathbf{F} = \mathbf{D}_N^{-\frac{1}{2}} \mathbf{N}^\top \mathbf{B}^{-\top} \mathbf{D}_B^{\frac{1}{2}}$, então podemos reescrever a matriz (2.11a) como $\mathbf{I}_m + \mathbf{F}^\top \mathbf{F}$, a matriz (2.11b) como $\mathbf{I}_{n-m} + \mathbf{F} \mathbf{F}^\top$, fazendo com que a matriz (2.10) se torne

$$\begin{bmatrix} \mathbf{I}_m & \mathbf{F}^\top \\ \mathbf{F} & -\mathbf{I}_{n-m} \end{bmatrix}.$$

Teorema 2.3. *As matrizes (2.11a) e (2.11b) são definidas positivas e seus autovalores são maiores ou iguais a 1.*

Demonstração. Seja v um autovetor normalizado associado ao autovalor λ , (λ, v) autopar de $\mathbf{I}_{n-m} + \mathbf{F}\mathbf{F}^\top$. Então

$$\begin{aligned} (\mathbf{I}_{n-m} + \mathbf{F}\mathbf{F}^\top)v &= \lambda v \\ v^\top(v + \mathbf{F}\mathbf{F}^\top v) &= \lambda v^\top v \\ 1 + u^\top u &= \lambda, \end{aligned}$$

onde $u = \mathbf{F}^\top v$. Então $\lambda \geq 1$. Analogamente, seja w um autovetor normalizado associado ao autovalor θ , (θ, w) autopar de $\mathbf{I}_m + \mathbf{F}^\top \mathbf{F}$. Então

$$\begin{aligned} (\mathbf{I}_m + \mathbf{F}^\top \mathbf{F})w &= \theta w \\ w^\top(w + \mathbf{F}^\top \mathbf{F}w) &= \theta w^\top w \\ 1 + s^\top s &= \theta, \end{aligned}$$

onde $s = \mathbf{F}w$. então $\theta \geq 1$. □

Observa-se que as matrizes (2.11a) e (2.11b) têm o mesmo conjunto de autovalores, com exceção dos autovalores extras iguais a um da matriz de maior dimensão. Os resultados acima, juntamente com o Corolário 11.3.2 de Golub e Van Loan (2013, p. 630), mostram que o método dos gradientes conjugados converge em até $\min\{m, n - m\} + 1$ iterações. Apesar da pouca diferença de desempenho entre ambas as matrizes, neste trabalho estamos interessados em abordar a matriz (2.11b).

Teorema 2.4. *Considere o seguinte problema de autovalor*

$$\begin{bmatrix} \mathbf{I}_m & \mathbf{F}^\top \\ \mathbf{F} & -\mathbf{I}_{n-m} \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \lambda \begin{bmatrix} u \\ v \end{bmatrix}.$$

Então, (θ, u) é um autopar de $\mathbf{I}_m + \mathbf{F}^\top \mathbf{F}$ e (θ, v) é um autopar de $\mathbf{I}_{n-m} + \mathbf{F}\mathbf{F}^\top$, onde $\theta = \lambda^2$.

Demonstração. Notemos que

$$\begin{bmatrix} \mathbf{I}_m & \mathbf{F}^\top \\ \mathbf{F} & -\mathbf{I}_{n-m} \end{bmatrix} \begin{bmatrix} \mathbf{I}_m & \mathbf{F}^\top \\ \mathbf{F} & -\mathbf{I}_{n-m} \end{bmatrix} = \begin{bmatrix} \mathbf{I}_m + \mathbf{F}^\top \mathbf{F} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{n-m} + \mathbf{F}\mathbf{F}^\top \end{bmatrix}.$$

Então

$$\begin{bmatrix} \mathbf{I}_m + \mathbf{F}^\top \mathbf{F} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{n-m} + \mathbf{F}\mathbf{F}^\top \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \lambda^2 \begin{bmatrix} u \\ v \end{bmatrix}.$$

□

De (2.10) obtemos

$$(\mathbf{PWP}^\top)^{-1} = \begin{bmatrix} \mathbf{I}_m - \mathbf{F}^\top \mathbf{T} \mathbf{F} & \mathbf{F}^\top \mathbf{T} \\ \mathbf{T} \mathbf{F} & -\mathbf{T} \end{bmatrix},$$

onde $\mathbf{T} = (\mathbf{I}_{n-m} + \mathbf{F} \mathbf{F}^\top)^{-1}$. Segue, então, que o menor autovalor de \mathbf{W}^{-1} é

$$|\lambda|^2 = \left\| \begin{bmatrix} u - \mathbf{F}^\top \mathbf{T} \mathbf{F} u + \mathbf{F}^\top \mathbf{T} v \\ \mathbf{T} \mathbf{F} u - \mathbf{F}^\top \mathbf{T} v \end{bmatrix} \right\|,$$

onde $\|u^\top v^\top\| = 1$. Perto de uma solução, as últimas $n - m$ entradas de \mathbf{D} são grandes. Então, para uma escolha adequada das colunas de \mathbf{B} obtemos entradas da diagonal de \mathbf{D}_B e de \mathbf{D}_N^{-1} pequenas (quando próximo de uma solução). Assim, $\mathbf{F} = \mathbf{D}_N^{-\frac{1}{2}} \tilde{\mathbf{N}}^\top \mathbf{D}_B^{\frac{1}{2}}$ (com $\tilde{\mathbf{N}} = \mathbf{B}^{-1} \mathbf{N}$) se aproxima da matriz nula, \mathbf{T} se aproxima da matriz identidade e tanto $\kappa_2(\mathbf{W})$ quanto o maior autovalor de \mathbf{W} se aproximam de 1 (OLIVEIRA; SORENSEN, 2005).

2.5.4 Escolhendo o conjunto de colunas

Do Teorema 1.4 e do Teorema 1.5, podemos concluir que, em uma solução ótima, somente um de x_i^* e s_i^* é zero, para $i = 1, \dots, n$. O conjunto de índices $\{1, 2, \dots, n\}$ pode, então, ser particionado em dois conjuntos disjuntos, a saber

$$\mathcal{B} := \{i \in \{1, 2, \dots, n\}; x_i^* > 0\} \quad \mathcal{N} := \{i \in \{1, 2, \dots, n\}; s_i^* > 0\}.$$

Desta forma e considerando não degenerescência, se $i \in \mathcal{B}$ em uma solução ótima, então $x_i \rightarrow x^* > 0$ e $s_i \rightarrow 0$, fazendo com que a entrada d_i correspondente seja tal que $d_i \rightarrow 0$. Analogamente, se $i \in \mathcal{N}$ então $x_i \rightarrow 0$ e $s_i \rightarrow s^* > 0$, fazendo com que a entrada d_i correspondente seja tal que $d_i \rightarrow \infty$. Resumindo, temos

$$d_i \rightarrow \begin{cases} 0, & \text{se } i \in \mathcal{B} \\ \infty, & \text{se } i \in \mathcal{N} \end{cases} \quad \text{e} \quad d_i^{-1} \rightarrow \begin{cases} \infty, & \text{se } i \in \mathcal{B} \\ 0, & \text{se } i \in \mathcal{N} \end{cases}.$$

Particionando

$$\mathbf{D} = \begin{bmatrix} \mathbf{D}_B & \mathbf{0} \\ \mathbf{0} & \mathbf{D}_N \end{bmatrix},$$

podemos concluir que $\mathbf{D}_B \approx \mathbf{0}$ e $\mathbf{D}_N \approx \infty$, tal como $\mathbf{D}_B^{-1} \approx \infty$ e $\mathbf{D}_N^{-1} \approx \mathbf{0}$. Estas propriedades são responsáveis pelo mal condicionamento dos sistemas (1.23) e (1.24). Uma vez escolhido o conjunto de colunas de \mathbf{A} que formam \mathbf{B} , considera-se a partição $\mathbf{A} = [\mathbf{B} \ \mathbf{N}]$ e usamos a notação \mathbf{D}_B e \mathbf{D}_N para representar \mathbf{D}_B e \mathbf{D}_N , respectivamente. (AL-JEIROUDI; GONDZIO; HALL, 2008).

Castro e Oliveira (2015, p. 2931) mostra que podemos obter um limitante superior para os autovalores de $\mathbf{I}_m + \mathbf{F}^\top \mathbf{F}$ e de $\mathbf{I}_{n-m} + \mathbf{F} \mathbf{F}^\top$. Do Teorema 2.3 temos

que $1 + s^\top s = \theta$, onde $s = \mathbf{F}w$ e (θ, w) é um autopar de $\mathbf{I}_m + \mathbf{F}^\top \mathbf{F}$. Então, dado que $s^\top s = \|\mathbf{F}w\|_2 \leq \|\mathbf{F}\|_2 \|w\|_2$ e que $\|w\|_2 = 1$, obtemos

$$\theta = 1 + s^\top s = 1 + \|\mathbf{F}w\|_2^2 \leq 1 + \|\mathbf{F}\|_2^2.$$

Ainda pelo Teorema 2.3, temos que $1 + u^\top u = \lambda$, onde $u = \mathbf{F}^\top v$ e (λ, v) é um autopar de $\mathbf{I}_{n-m} + \mathbf{F}\mathbf{F}^\top$. Então, dado que $u^\top u = \|\mathbf{F}^\top v\|_2 \leq \|\mathbf{F}^\top\|_2 \|v\|_2$ e que $\|v\|_2 = 1$, obtemos

$$\lambda = 1 + u^\top u = 1 + \|\mathbf{F}^\top v\|_2^2 \leq 1 + \|\mathbf{F}^\top\|_2^2.$$

Conforme mencionado por Oliveira e Sorensen (2005, p. 11), uma escolha adequada do conjunto de colunas de \mathbf{A} que formam \mathbf{B} faz com que o pré-condicionador apresentado tenha melhor desempenho nas iterações finais do método de pontos interiores. Pela abordagem anterior, tal escolha pode ser feita minimizando $\|\mathbf{F}\|_2$ ou $\|\mathbf{F}^\top\|_2$, onde $\mathbf{F} = \mathbf{D}_N^{-\frac{1}{2}} \mathbf{N}^\top \mathbf{B}^{-\top} \mathbf{D}_B^{\frac{1}{2}}$.

Por outro lado, como $\mathbf{F}\mathbf{F}^\top$ é simétrica, pelo Teorema Espectral para Matrizes Reais Simétricas³ existem uma matriz ortogonal \mathbf{U} , formada pelos autovetores de $\mathbf{F}\mathbf{F}^\top$, e uma matriz diagonal \mathbf{E} , formada pelos autovalores de $\mathbf{F}\mathbf{F}^\top$, tais que $\mathbf{F}\mathbf{F}^\top = \mathbf{U}\mathbf{E}\mathbf{U}^\top$. Assim, vemos que

$$\|\mathbf{F}^\top\|_2^2 = \gamma_m \leq \text{tr}(\mathbf{E}) = \text{tr}(\mathbf{U}\mathbf{E}\mathbf{U}^\top) = \text{tr}(\mathbf{F}\mathbf{F}^\top) = \|\mathbf{F}^\top\|_F^2,$$

onde γ_m é o maior autovalor de $\mathbf{F}\mathbf{F}^\top$. Ou seja, temos que $\|\mathbf{F}^\top\|_2 \leq \|\mathbf{F}^\top\|_F$. A norma de Frobenius de uma matriz $\mathbf{X} \in \mathbb{M}_n(\mathbb{R})$ pode ser escrita como $\|\mathbf{X}\|_F^2 = \sum_{j=1}^n \|\mathbf{X}^j\|_2^2$, onde \mathbf{X}^j é a coluna j de \mathbf{X} .

Proposição 2.6. *Castro e Oliveira (2015, p. 2932). Considere $\mathbf{F} = \mathbf{D}_N^{-\frac{1}{2}} \mathbf{N}^\top \mathbf{B}^{-\top} \mathbf{D}_B^{\frac{1}{2}}$. Se a matriz \mathbf{B} for composta pelas primeiras m colunas \mathbf{A}^j de \mathbf{A} linearmente independentes, tais que j corresponde aos maiores valores de $\|\mathbf{A}^j d_j^{-\frac{1}{2}}\|_2$, então $\|\mathbf{F}^\top\|_F$ é minimizada.*

Demonstração. Como $\|\mathbf{F}^\top\|_F \leq \|\mathbf{D}_B^{\frac{1}{2}} \mathbf{B}^{-1}\|_F \|\mathbf{N} \mathbf{D}_N^{-\frac{1}{2}}\|_F$, segue que precisamos minimizar $\|\mathbf{D}_B^{\frac{1}{2}} \mathbf{B}^{-1}\|_F$ e $\|\mathbf{N} \mathbf{D}_N^{-\frac{1}{2}}\|_F$. Para o problema de minimizar $\|\mathbf{D}_B^{\frac{1}{2}} \mathbf{B}^{-1}\|_F$, dado que $\|\mathbf{B} \mathbf{D}_B^{-\frac{1}{2}}\|_F \geq \frac{1}{\|\mathbf{D}_B^{\frac{1}{2}} \mathbf{B}^{-1}\|_F}$ podemos concluir que maximizando $\|\mathbf{B} \mathbf{D}_B^{-\frac{1}{2}}\|_F$ estaremos a minimizar $\|\mathbf{D}_B^{\frac{1}{2}} \mathbf{B}^{-1}\|_F$. Agora, como $\mathbf{A} \mathbf{D}^{-\frac{1}{2}} = \begin{bmatrix} \mathbf{B} \mathbf{D}_B^{-\frac{1}{2}} & \mathbf{N} \mathbf{D}_N^{-\frac{1}{2}} \end{bmatrix}$ e pela definição da norma de Frobenius dada acima, segue que maximizar $\|\mathbf{B} \mathbf{D}_B^{-\frac{1}{2}}\|_F$ é o mesmo que escolher as colunas de $\mathbf{A} \mathbf{D}^{-\frac{1}{2}}$ de maior norma-2 e que minimizar $\|\mathbf{N} \mathbf{D}_N^{-\frac{1}{2}}\|_F$ é o mesmo que escolher as colunas de $\mathbf{A} \mathbf{D}^{-\frac{1}{2}}$ de menor norma-2. Assim, \mathbf{B} deve ser composta pelas m colunas linearmente independentes de $\mathbf{A} \mathbf{D}^{-\frac{1}{2}}$ de maior norma-2. O fato de $\mathbf{A} \mathbf{D}^{-\frac{1}{2}} = \begin{bmatrix} \mathbf{B} \mathbf{D}_B^{-\frac{1}{2}} & \mathbf{N} \mathbf{D}_N^{-\frac{1}{2}} \end{bmatrix}$ diz que maximizar $\|\mathbf{B} \mathbf{D}_B^{-\frac{1}{2}}\|_F$ e minimizar $\|\mathbf{N} \mathbf{D}_N^{-\frac{1}{2}}\|_F$ são condições equivalentes. \square

³ O Teorema Espectral para Matrizes Reais Simétricas pode ser encontrado em Watkins (2010, p. 341).

A heurística descrita pela Proposição 2.6 foi implementada no trabalho de Velazco, Oliveira e Campos (2010). Apesar de ser uma heurística relativamente barata de calcular, seus resultados não foram satisfatórios. No mesmo trabalho, a heurística de escolher sobre as colunas de \mathbf{AD}^{-1} se mostrou mais promissora nos resultados computacionais dos problemas comparados e, atualmente, esta é a heurística que está implementada. Tal escolha tende a produzir matrizes com melhor número de condição quando o método de pontos interiores se aproxima de uma solução.

3 Aspectos Práticos

Para realizar os testes computacionais, optou-se por utilizar uma versão modificada do *solver* PCx, proposta por Velazco, Oliveira e Campos (2010).

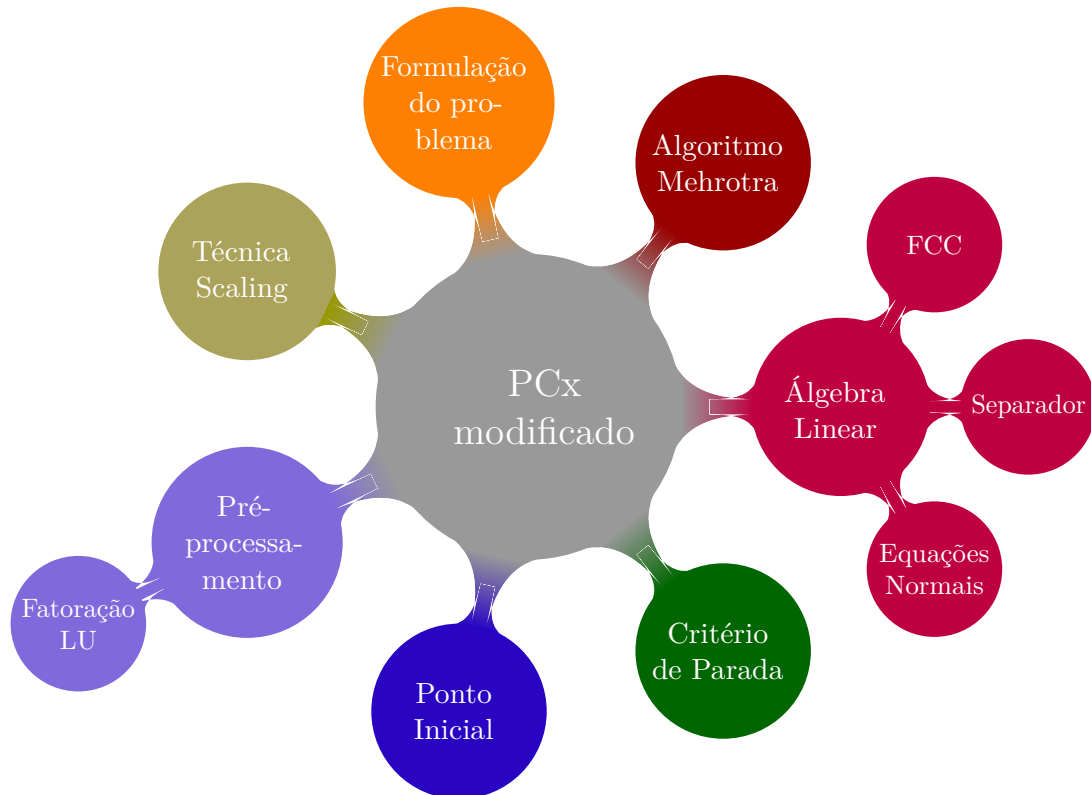
Figura 2 – Técnicas adotadas pela versão original do *solver* PCx



Produzida pelo autor.

A versão original do PCx contém uma implementação do método Preditor-Corretor de Mehrotra, descrito no Algoritmo 3. Conforme explicam Czyzyk et al. (1997, p. 8), o PCx implementa um algoritmo para fatoração de Cholesky esparsa na decomposição da matriz de coeficientes do sistema do complemento de Schur (1.24). Esta versão incorpora, ainda, uma estratégia de múltiplas correções de centralidade proposta por Gondzio (1996). Czyzyk et al. (1997, p. 6) explicam que, nesta estratégia, são configurados diferentes vetores para o lado direito do sistema oriundo da aplicação do método de Newton nas condições de otimalidade do problema para resolver diferentes sistemas lineares, obtendo direções de centragem/corretoras adicionais. Outras técnicas também são adotadas, como *scaling* e pré-processamento, além de critérios de parada e da estratégia desenvolvida por Mehrotra (1992, p. 589) para cálculo do ponto inicial. A Figura 2 ilustra as técnicas adotadas pela versão original, que serão abordadas brevemente neste capítulo.

Na versão modificada do PCx, está implementada uma versão híbrida de

Figura 3 – Técnicas adotadas pela versão modificada do *solver* PCx, obtida pelo autor

Produzida pelo autor.

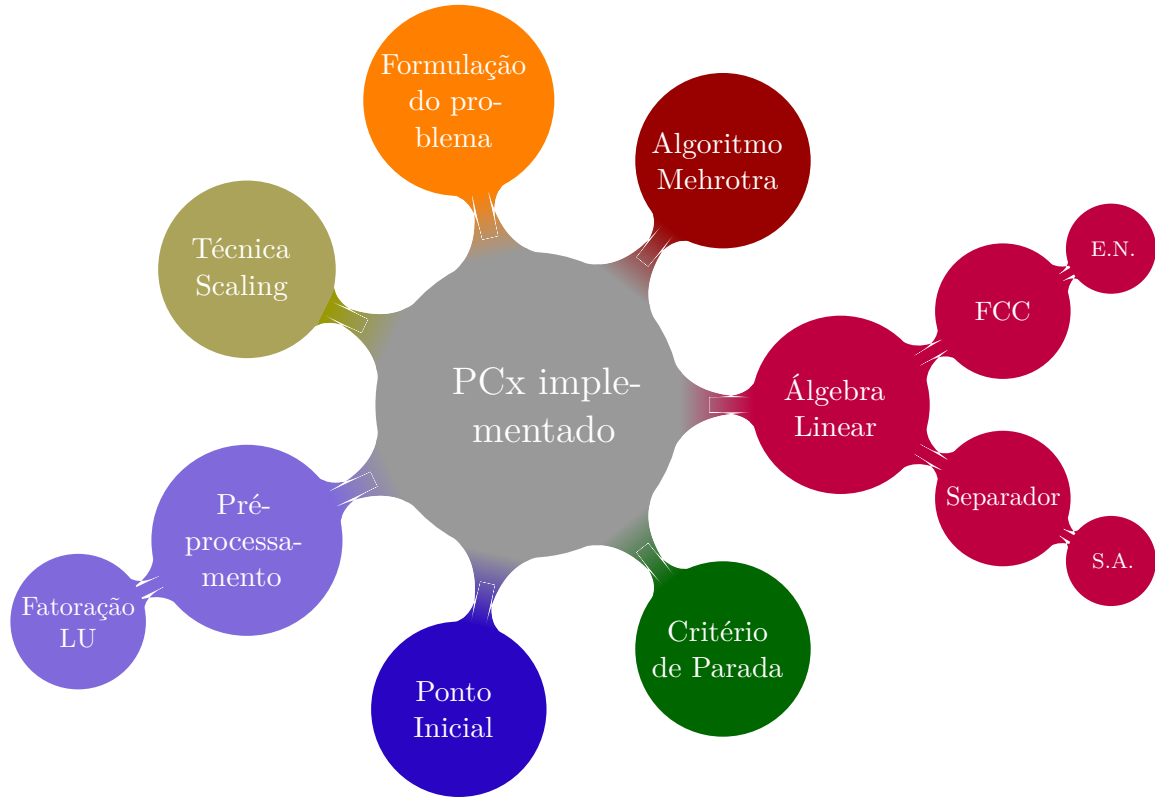
pré-condicionamento para a matriz (1.24) e um algoritmo de Gradientes Conjugados Pré-condicionado¹ para obter a solução dos sistemas lineares. Assumindo a existência de duas fases durante as iterações do método de pontos interiores, a versão modificada constrói, durante a primeira fase, o pré-condicionador da Fatoração Controlada de Cholesky (FCC) para a matriz de (1.24) (um pré-condicionador baseado em uma fatoração incompleta de Cholesky), enquanto que, na segunda fase, o pré-condicionador separador (descrito na seção 2.5) é utilizado. Nesta versão, as múltiplas correções de alta ordem estão desativadas e um procedimento que identifica linhas redundantes, baseado na fatoração LU, está incorporado à etapa de pré-processamento. A Figura 3 ilustra as técnicas adotadas por esta versão.

A versão adotada pelo autor para implementação das propostas desta dissertação difere da versão proposta por Velazco, Oliveira e Campos (2010) apenas na solução dos sistemas lineares. Na fase 1, os sistemas são resolvidos utilizando o pré-condicionador da Fatoração Controlada de Cholesky no sistema de equações normais, enquanto que a fase 2 resolve os sistemas lineares usando o pré-condicionador separador no sistema aumentado. A figura Figura 4 apresenta um resumo das técnicas adotadas pela versão implementada do PCx (abreviamos “equações normais” para E.N. e “sistema aumentado” para S.A.).

¹ O método de Gradientes Conjugados Pré-condicionado está descrito no Apêndice B.

Neste capítulo, serão abordadas algumas considerações sobre a estratégia híbrida de pré-condicionamento, bem como sobre aspectos práticos associados à implementação de códigos computacionais em programação linear.

Figura 4 – Técnicas adotadas pelo *solver* PCx na versão implementada pelo autor



Produzida pelo autor.

3.1 Fatoração Incompleta de Cholesky

Qualquer matriz $\mathbf{A} \in \mathbb{M}_n(\mathbb{R})$, simétrica definida positiva, pode ser fatorada como $\mathbf{A} = \mathbf{L}\mathbf{L}^\top$, onde \mathbf{L} é uma matriz triangular inferior com entradas positivas na diagonal principal. Esta é a fatoração de Cholesky de \mathbf{A} . Se \mathbf{A} é esparsa, \mathbf{L} pode ser muito menos esparsa que \mathbf{A} . Podemos obter uma aproximação para a fatoração de \mathbf{A} impondo que o novo fator $\tilde{\mathbf{L}}$ apresente algum padrão de esparsidade \mathcal{P} semelhante ao padrão de esparsidade de \mathbf{A} . Tal fatoração aproximada é chamada *fatoração incompleta de Cholesky*. A matriz $\mathbf{C} = \tilde{\mathbf{L}}\tilde{\mathbf{L}}^\top$ pode ser usada como pré-condicionador em algum método iterativo, como o dos Gradientes Conjugados descrito em [Golub e Van Loan \(2013, p. 651\)](#) ([GREENBAUM, 1997](#)).

Os resultados apresentados a seguir garantem a existência da fatoração incompleta de Cholesky, uma vez definido o padrão de esparsidade \mathcal{P} . Para tanto, definimos o

conjunto

$$\mathcal{P}_n(\mathbf{A}) := \{(i, j); i \neq j, 1 \leq i \leq n, 1 \leq j \leq n\},$$

onde n é a dimensão da matriz \mathbf{A} . Note que o conjunto $\mathcal{P}_n(\mathbf{A})$ contém todos os pares de índices de entradas que estão fora da diagonal principal de \mathbf{A} .

Definição 3.1. Uma matriz $\mathbf{A} \in \mathbb{M}_n(\mathbb{R})$ é dita ser uma *M-matriz* se satisfaz as seguintes propriedades:

1. $a_{i,i} > 0$ para $i = 1 \dots, n$.
2. $a_{i,j} \leq 0$ para $i \neq j, i, j = 1 \dots, n$.
3. \mathbf{A} é não singular.
4. $\mathbf{A}^{-1} \geq \mathbf{0}$.

A última propriedade da definição acima indica que $a_{i,j}^{-1} \geq 0$ para $i, j = 1, \dots, n$.

Definição 3.2. Para matrizes $\mathbf{A}, \mathbf{C}, \mathbf{R} \in \mathbb{M}_n(\mathbb{R})$, $\mathbf{A} = \mathbf{C} - \mathbf{R}$ é um *splitting regular* da matriz \mathbf{A} se \mathbf{C} é não singular, $\mathbf{C}^{-1} \geq \mathbf{0}$ e $\mathbf{R} \geq \mathbf{0}$.

Teorema 3.1. Se $\mathbf{A} \in \mathbb{M}_n(\mathbb{R})$ é uma *M-matriz*, então para todo $\mathcal{P} \subset \mathcal{P}_n(\mathbf{A})$ existem uma matriz \mathbf{L} triangular inferior com diagonal principal unitária, uma matriz \mathbf{U} triangular superior e uma matriz \mathbf{R} , onde

$$\begin{aligned} l_{i,j} &= 0 & \text{se } (i, j) \in \mathcal{P}, \\ u_{i,j} &= 0 & \text{se } (i, j) \in \mathcal{P}, \\ r_{i,j} &= 0 & \text{se } (i, j) \notin \mathcal{P}, \end{aligned}$$

tais que $\mathbf{A} = \mathbf{LU} - \mathbf{R}$ é um *splitting regular*. Os fatores \mathbf{L} e \mathbf{U} são únicos.

Demonstração. Vide [Meijerink e van der Vorst \(1977, p. 150\)](#). □

Como pode ser visto em [Golub e Van Loan \(2013, p. 114\)](#), satisfeitas as condições de existência, podemos fatorar a matriz \mathbf{A} da forma $\mathbf{A} = \mathbf{LU}$, onde \mathbf{L} é uma matriz triangular inferior com diagonal principal unitária e \mathbf{U} é uma matriz triangular superior. Os fatores \mathbf{L} e \mathbf{U} mencionados no teorema acima são aqueles oriundos da fatoração incompleta \mathbf{LU} de \mathbf{A} . Se a matriz \mathbf{A} possui a fatoração \mathbf{LU} , podemos escrever $\mathbf{A} = \mathbf{LDV}$ onde \mathbf{L} é uma matriz triangular inferior com diagonal unitária, $\mathbf{D} = \text{diag}(u_{11}, \dots, u_{nn})$ é uma matriz diagonal e $\mathbf{V} = \mathbf{D}^{-1}\mathbf{U}$ é uma matriz triangular superior com diagonal unitária. Se \mathbf{A} é simétrica, então $\mathbf{A} = \mathbf{LDL}^\top$ com $\mathbf{L}^\top = \mathbf{V}$. Se \mathbf{A} é simétrica e definida positiva, então as entradas da diagonal principal de \mathbf{D} são positivas, permitindo-nos escrever $\mathbf{D} = \mathbf{D}^{\frac{1}{2}}\mathbf{D}^{\frac{1}{2}}$. Assim, $\mathbf{A} = \mathbf{LDL}^\top = \mathbf{LD}^{\frac{1}{2}}\mathbf{D}^{\frac{1}{2}}\mathbf{L}^\top = \left(\mathbf{LD}^{\frac{1}{2}}\right)\left(\mathbf{LD}^{\frac{1}{2}}\right)^\top = \mathbf{GG}^\top$, onde $\mathbf{G} = \mathbf{LD}^{\frac{1}{2}}$ é o fator

de Cholesky de \mathbf{A} . Uma M-matriz simétrica é uma matriz positiva definida, e neste caso a chamamos de matriz de Stieltjes ([HACKBUSCH, 1994](#); [GREENBAUM, 1997](#); [WATKINS, 2010](#)).

Teorema 3.2. *Se $\mathbf{A} \in \mathbb{M}_n(\mathbb{R})$ é uma M-matriz simétrica, então para cada $\mathcal{P} \subset \mathcal{P}_n(\mathbf{A})$ com a propriedade que $(i, j) \in \mathcal{P}$ implica $(j, i) \in \mathcal{P}$ existem uma única matriz triangular inferior $\tilde{\mathbf{L}}$ e uma matriz não negativa \mathbf{R} com*

$$\begin{aligned} \tilde{l}_{i,j} &= 0 & \text{se } (i, j) \in \mathcal{P}, \\ r_{i,j} &= 0 & \text{se } (i, j) \notin \mathcal{P}, \end{aligned}$$

tais que $\mathbf{A} = \tilde{\mathbf{L}}\tilde{\mathbf{L}}^\top - \mathbf{R}$ é um splitting regular.

O [Teorema 3.2](#) mostra que condições devem ser satisfeitas para que a fatoração incompleta de Cholesky exista. Sua demonstração possui relação com a existência da fatoração incompleta LU de \mathbf{A} , conforme observa [Meijerink e van der Vorst \(1977, p. 151\)](#).

A maioria dos métodos iterativos é baseada na seguinte ideia: dado o sistema linear $\mathbf{A}x = b$, se \mathbf{C} é não singular, então $\mathbf{A} = \mathbf{C} - \mathbf{R}$ é um splitting de \mathbf{A} e associado a ele está o método iterativo

$$\mathbf{C}x^{(k+1)} = (\mathbf{C} - \mathbf{A})x^{(k)} + b = \mathbf{R}x^{(k)} + b$$

ou

$$x^{(k+1)} = x^{(k)} + \mathbf{C}^{-1}(b - \mathbf{A}x^{(k)}) = x^{(k)} + \Delta x^{(k)},$$

e resolvemos, a cada iteração,

$$\mathbf{C}\Delta x^{(k)} = x^{(k+1)} - x^{(k)} = b - \mathbf{A}x^{(k)}.$$

Teorema 3.3. *Se \mathbf{A} , $\tilde{\mathbf{L}}$ e \mathbf{R} são definidas como no [Teorema 3.2](#), então o método iterativo*

$$\tilde{\mathbf{L}}\tilde{\mathbf{L}}^\top x^{(k+1)} = \mathbf{R}x^{(k)} + b$$

irá convergir para a solução de $\mathbf{A}x = b$, para qualquer escolha de $x^{(0)}$.

Demonstração. Vide [Varga \(1962, p. 89\)](#). □

3.1.1 Fatoração Controlada de Cholesky

O pré-condicionador da Fatoração Controlada de Cholesky (FCC) foi desenvolvido para resolver sistemas lineares definidos positivos e usado com sucesso na solução de sistemas lineares oriundos de equações diferenciais parciais com dependência de tempo ([CAMPOS FILHO, 1995](#)).

Considere uma matriz simétrica e definida positiva, digamos $\mathbf{A} \in \mathbb{M}_n(\mathbb{R})$, e seja

$$\mathbf{A} = \mathbf{L}\mathbf{L}^\top = \tilde{\mathbf{L}}\tilde{\mathbf{L}}^\top + \mathbf{R},$$

sendo $\mathbf{L} \in \mathbb{M}_n(\mathbb{R})$ o fator completo de Cholesky de \mathbf{A} , $\tilde{\mathbf{L}} \in \mathbb{M}_n(\mathbb{R})$ o fator incompleto de Cholesky da mesma matriz \mathbf{A} e $\mathbf{R} \in \mathbb{M}_n(\mathbb{R})$ o resíduo da fatoração incompleta. Usando $\tilde{\mathbf{L}}$ como pré-condicionador para a matriz \mathbf{A} podemos escrever

$$\tilde{\mathbf{L}}^{-1}\mathbf{A}\tilde{\mathbf{L}}^{-\top} = \tilde{\mathbf{L}}^{-1}\mathbf{L}\mathbf{L}^\top\tilde{\mathbf{L}}^{-\top} = (\tilde{\mathbf{L}}^{-1}\mathbf{L})(\tilde{\mathbf{L}}^{-1}\mathbf{L})^\top. \quad (3.1)$$

Definindo $\mathbf{E} = \mathbf{L} - \tilde{\mathbf{L}}$, podemos reescrever a última expressão de (3.1) como

$$(\tilde{\mathbf{L}}^{-1}\mathbf{L})(\tilde{\mathbf{L}}^{-1}\mathbf{L})^\top = (\mathbf{I}_n + \tilde{\mathbf{L}}^{-1}\mathbf{E})(\mathbf{I}_n + \tilde{\mathbf{L}}^{-1}\mathbf{E})^\top.$$

Podemos observar que, à medida em que a matriz $\tilde{\mathbf{L}}$ se aproxima da matriz \mathbf{L} , então a matriz \mathbf{E} se aproxima da matriz $\mathbf{0}$ e a matriz $\tilde{\mathbf{L}}^{-1}\mathbf{A}\tilde{\mathbf{L}}^{-\top}$ se aproxima da matriz \mathbf{I}_n (CAMPOS FILHO, 1995).

Uma vez que o número de iterações do método de Gradientes Conjugados possui relação com a norma da matriz $\mathbf{R} = \mathbf{L}\mathbf{E}^\top + \mathbf{E}\tilde{\mathbf{L}}^\top = \mathbf{L}\mathbf{L}^\top - \tilde{\mathbf{L}}\tilde{\mathbf{L}}^\top$, a FCC tem como base a minimização da norma de Frobenius da matriz \mathbf{E} (se $\|\mathbf{E}\|_F \rightarrow \mathbf{0}$ então $\|\mathbf{R}\|_F \rightarrow \mathbf{0}$). Para tanto, considera-se o problema

$$\min \|\mathbf{E}\|_F^2 = \sum_{j=1}^n c_j \quad \text{onde} \quad c_j = \sum_{i=1}^n |l_{ij} - \tilde{l}_{ij}|^2, \quad (3.2)$$

com l_{ij} e \tilde{l}_{ij} representando o ij -ésimo elemento das matrizes \mathbf{L} e $\tilde{\mathbf{L}}$, respectivamente. Divide-se, ainda, c_j como

$$c_j = \sum_{k=1}^{m_j+\eta} |l_{ikj} - \tilde{l}_{ikj}|^2 + \sum_{k=m_j+\eta+1}^n |l_{ikj}|^2, \quad (3.3)$$

onde m_j é o número de elementos não nulos abaixo da diagonal na j -ésima coluna da matriz \mathbf{A} , η é o número extra de elementos não nulos permitido por coluna e n é a ordem da matriz. Aqui, os $m_j + \eta$ elementos não nulos da j -ésima coluna de $\tilde{\mathbf{L}}$ estão no primeiro somatório, enquanto que os elementos do fator completo \mathbf{L} que não estão em $\tilde{\mathbf{L}}$ aparecem no segundo somatório. Feitas estas considerações, a seguinte estratégia é utilizada para resolver o problema (3.2):

1. Aumentando o valor do parâmetro η , permitindo maior preenchimento, c_j decrescerá;
2. Escolhendo os maiores $m_j + \eta$ elementos de $\tilde{\mathbf{L}}$ (em valor absoluto) para um η fixo, agrupamos os maiores elementos no primeiro somatório e os menores no segundo, construindo um fator $\tilde{\mathbf{L}}$ ótimo para uma determinada quantidade de armazenamento.

Como a matriz $\tilde{\mathbf{L}}$ é construída por colunas, ela precisa apenas da j -ésima coluna de \mathbf{A} por vez, obtendo uma melhor aproximação para o fator completo com a quantidade de memória disponível. As principais características do pré-condicionador FCC podem ser encontradas no trabalho de [Campos e Birkett \(1998, p. 129\)](#) e estão resumidas a seguir:

1. Escolha das entradas por valor, o que nos permite trabalhar com as maiores, em valor absoluto;
2. Armazenamento previsível;
3. Incremento exponencial para evitar falhas na fatoração incompleta;
4. Versatilidade: o número de entradas não nulas por coluna pode variar de 1 a m .

3.2 Mudança de Fases

A versão modificada do PCx, que foi considerada neste trabalho, possui uma implementação de um pré-condicionador híbrido proposta por [Bocanegra, Campos e Oliveira \(2007, p. 153\)](#). Nesta versão, considera-se a existência de duas fases no decorrer das operações do método Preditor-Corretor de Mehrotra. Ambas as fases foram desenvolvidas orientadas para solução dos sistemas lineares do método de Mehrotra a partir do método de Gradientes Conjugados Pré-condicionado. Sendo assim, foi considerada a matriz $\tilde{\mathbf{L}}$, obtida do procedimento da FCC, como matriz pré-condicionadora para a fase 1, enquanto que o pré-condicionador separador, descrito na [seção 2.5](#), foi considerado para a fase 2.

[Velazco, Oliveira e Campos \(2010, p. 327\)](#) propuseram a seguinte heurística para a mudança de fases: atualiza-se o parâmetro η da FCC fazendo $\eta \leftarrow \eta + 10$ sempre que a quantidade de iterações necessárias para a convergência do método de Gradientes Conjugados Pré-condicionado ultrapassar o valor $m/6$, onde m é a dimensão da matriz de coeficientes do sistema do complemento de Schur (1.24). [Velazco, Oliveira e Campos \(2010, p. 327\)](#) mostraram resultados numéricos que comprovam a eficácia da heurística apresentada, comparada com a heurística apresentada inicialmente por [Bocanegra, Campos e Oliveira \(2007, p. 157\)](#).

3.3 Considerações sobre o algoritmo utilizado

Nesta seção, abordaremos as questões práticas de implementação de códigos computacionais que resolvem problemas de programação linear. Dada uma formulação padrão como entrada do algoritmo, realiza-se uma etapa de pré-processamento a fim de retirar informações “inúteis” contidas na formulação padrão. A escolha do ponto inicial

pode afetar na velocidade de convergência do algoritmo, que deve ter parâmetros pré-configurados como critério de parada para aceitar uma solução como suficientemente boa.

3.3.1 Formulação adotada

Conforme descrito por Czyzyk et al. (1997, p. 2), o PCx processa os dados recebidos da modelagem do problema de programação linear de forma a obter uma formulação mais simples. Inicialmente, o solver obtém a seguinte formulação

$$\begin{aligned} \min \quad & c^\top x \\ \text{s. a} \quad & \mathbf{A}x = b \\ & 0 \leq x_i \quad i \in \mathcal{I} \\ & 0 \leq x_i \leq u_i \quad i \in \mathcal{B} \\ & x_i \text{ livre} \quad i \in \mathcal{L}, \end{aligned} \tag{3.4}$$

onde $\mathcal{I} \cup \mathcal{B} \cup \mathcal{L}$ é uma partição do conjunto de índices $\{1, 2, \dots, n\}$ em variáveis “normais”, “limitadas superiormente” e “livres”, respectivamente.

Posteriormente, para evitar o uso de variáveis livres, o solver faz $x_i = x_i^+ - x_i^-$ para todo $i \in \mathcal{L}$, obtendo, finalmente, a formulação

$$\begin{aligned} \min \quad & c^\top x \\ \text{s. a} \quad & \mathbf{A}x = b \\ & 0 \leq x_i \quad i \in \bar{\mathcal{I}} \\ & 0 \leq x_i \leq u_i \quad i \in \mathcal{B}, \end{aligned} \tag{3.5}$$

onde $\bar{\mathcal{I}} = \{1, 2, \dots, n\} \setminus \mathcal{B}$. A formulação (3.5) pode ser reescrita como

$$\begin{aligned} \min \quad & c^\top x \\ \text{s. a} \quad & \mathbf{A}x = b \\ & \mathbf{E}x \leq u \\ & x \geq 0, \end{aligned} \tag{3.6}$$

onde $\mathbf{E} \in \mathbb{M}_{|\mathcal{B}| \times n}(\mathbb{R})$ é uma matriz retangular que contém linhas da matriz identidade de ordem n . Como feito no Capítulo 1, podemos escrever o dual de (3.6) como

$$\begin{aligned} \max \quad & b^\top y - u^\top t \\ \text{s. a} \quad & \mathbf{A}^\top y - \mathbf{E}^\top t + \mathbf{I}_n s = c \\ & (t, s) \geq 0. \end{aligned} \tag{3.7}$$

Adicionando uma variável de folga à formulação (pondo $x_i + v_i = u_i$, $i \in \mathcal{B}$), as condições de otimalidade passam a satisfazer factibilidade primal ($\mathbf{A}x = b$), factibilidade dual ($\mathbf{A}^\top y - t_i + s_i = c_i$, $i \in \mathcal{B}$ e $\mathbf{A}^\top y + s_i = c_i$, $i \in \bar{\mathcal{I}}$) e a relação de complementaridade

$(x_i s_i =, i \in \{1, 2, \dots, n\}$ e $v_i t_i = 0, i \in \bar{\mathcal{B}})$. Logo, o sistema não linear que representa estas condições fica

$$\begin{bmatrix} \mathbf{A}x - b \\ \mathbf{A}^\top y - \mathbf{E}^\top t + \mathbf{I}_n s - \mathbf{I}_n c \\ \mathbf{E}x + \mathbf{I}_{|\mathcal{B}|} v - \mathbf{I}_{|\mathcal{B}|} u \\ \mathbf{X}S e \\ \mathbf{V}T e \end{bmatrix} = 0. \quad (3.8)$$

As equações de passo para (3.8), que resultam da aplicação do método de Newton, ficam

$$\begin{bmatrix} \mathbf{A} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{A}^\top & \mathbf{I}_n & \mathbf{0} & -\mathbf{E}^\top \\ \mathbf{E} & \mathbf{0} & \mathbf{0} & \mathbf{I}_{|\mathcal{B}|} & \mathbf{0} \\ \mathbf{S} & \mathbf{0} & \mathbf{X} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{T} & \mathbf{V} \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta s \\ \Delta v \\ \Delta t \end{bmatrix} = \begin{bmatrix} r_p \\ r_d \\ r_u \\ r_{xs} \\ r_{vt} \end{bmatrix}, \quad (3.9)$$

onde os resíduos são definidos por

$$\begin{aligned} r_p &= b - \mathbf{A}x, \\ r_d &= c + \mathbf{E}^\top t - \mathbf{A}^\top y - s, \\ r_u &= u - \mathbf{E}x - v, \\ r_{xs} &= \mu e - \mathbf{X}S e, \\ r_{vt} &= \mu e - \mathbf{V}T e. \end{aligned}$$

O sistema aumentado para este caso fica sendo

$$\begin{bmatrix} -\mathbf{D} & \mathbf{A}^\top \\ \mathbf{A} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} r \\ r_p \end{bmatrix}, \quad (3.10)$$

com $\mathbf{D} = (\mathbf{X})^{-1}\mathbf{S} + \mathbf{E}^\top \mathbf{V}^{-1}\mathbf{T}\mathbf{E}$ e $r = r_d + \mathbf{E}^\top \mathbf{V}^{-1}r_{vt} - \mathbf{X}^{-1}r_{xs} - \mathbf{E}^\top \mathbf{V}^{-1}\mathbf{T}r_u$. Para o complemento de Schur, temos

$$\mathbf{A}\mathbf{D}^{-1}\mathbf{A}^\top \Delta y = \mathbf{A}\mathbf{D}^{-1}r + r_p. \quad (3.11)$$

A medida de dualidade para este modelo é dada por

$$\mu = \frac{x^\top s + v^\top t}{n + |\mathcal{B}|}. \quad (3.12)$$

Após uma solução ser encontrada, o PCx reverte as transformações realizadas (que originaram das formulações (3.4) e (3.5)) e exprime a solução nos termos da formulação original.

3.3.2 A técnica de pré-processamento

As técnicas de pré-processamento estão presentes em praticamente todas as implementações de programação linear e objetivam manipular as informações do problema a fim de reduzir seu tamanho original, evitar informações redundantes e estruturas que não necessitam de tratamentos e/ou algoritmos sofisticados. O PCx realiza as seguintes tarefas:

1. Infactibilidade – para as variáveis limitadas, verifica que cada limite superior é não negativo; verifica que linhas nulas de \mathbf{A} possuem zeros correspondentes no lado direito (vetor b);
2. Linhas Vazias – linhas nulas com correspondentes zeros no vetor b são removidas;
3. Linhas Duplicadas – linhas da matriz \mathbf{A} (juntamente com o correspondente valor no vetor b) múltiplas de outras, são removidas da formulação;
4. Colunas Duplicadas – quando colunas de \mathbf{A} são múltiplas (e as variáveis correspondentes não são limitadas nem livres), tais colunas são combinadas. Neste caso, as variáveis primais correspondentes às colunas resultantes da combinação podem ser “livres” ou “normais” (dependendo se as colunas são múltiplos positivos ou negativos de cada outra), mas não limitadas;
5. Colunas Vazias – a variável correspondente é fixada em um de seus limites (inferior ou superior), dependendo do sinal do correspondente coeficiente em c . Caso o limite não exista, é declarado que o problema primal é ilimitado;
6. Variáveis Fixas – variáveis cujos limites superior e inferior são iguais são fixadas e, então, retiradas da formulação;
7. Linhas Únicas – linhas que possuem uma única entrada não nula não necessitam de tratamento sofisticado de dados: se a linha i satisfaz esta condição, faz-se $x_i = b_i / \mathbf{A}_{ij}$ e eliminamos a linha i (e a respectiva variável dual) da formulação;
8. Colunas Únicas – quando \mathbf{A}_{ij} é a única entrada não nula da coluna j de \mathbf{A} e x_j é uma variável livre, expressamos x_j em termos de outras variáveis representadas na linha i de \mathbf{A} e a eliminamos do problema. Mesmo se não for livre, x_j pode ser eliminada se seus limites são mais fracos do que aqueles sugeridos pelas imagens de outros elementos representados na linha i ;
9. Linhas Forçadas – tratam-se de linhas que forçam outras variáveis a assumirem valores fixos;

Por padrão, o pré-processamento embutido no PCx executa múltiplos passos, tratando os dados iniciais do problema realizando as tarefas acima até que uma iteração do pré-processamento seja realizada sem que as rotinas alterem o problema. Adicionalmente, um procedimento que verifica por linhas redundantes foi considerado. Esta técnica pode ser muito cara, uma vez que realiza uma decomposição LU retangular. Contudo, é necessária para aplicação de algumas estratégias consideradas para solução de sistemas lineares. Tal procedimento só ocorre após finalizada a etapa de pré-processamento padrão do PCx (CZYZYK et al., 1997; OLIVEIRA, 1997).

A forma escolhida por Oliveira e Sorensen (2005, p. 13) é a fatoração LU com atualização atrasada, que trabalha com uma coluna de cada vez. Nesta versão da fatoração, colunas linearmente dependentes são descartadas até que se obtenha colunas linearmente independentes.

Se o procedimento obter menos do que m colunas independentes, a fatoração é aplicada ao lado direito do problema de programação linear. Caso este último seja linearmente independente do conjunto de colunas independentes já obtido, declaramos infactibilidade. Caso contrário, as demais linhas de \mathbf{A} são redundantes e podem ser descartadas do problema (OLIVEIRA, 1997).

Várias outras técnicas de pré-processamento podem ser empregadas. O montante de operações de pré-processamento, geralmente, não ultrapassa o de uma iteração do método de pontos-interiores e pode representar um procedimento bastante efetivo para reduzir o tempo e o esforço computacional na solução do problema de programação linear (OLIVEIRA, 1997).

3.3.3 Técnica de *scaling*

O *scaling* consiste em mudar o problema para reduzir o intervalo de domínio das entradas da matriz. A proposta da técnica é obter melhor comportamento numérico para o solver e pode melhorar o desempenho sobre alguns problemas, assim como piorar para outros. Existem várias técnicas propostas e a técnica utilizada é a padrão contida no PCx, que utiliza a função objetivo

$$\sum_{\mathbf{A}_{ij} \neq 0} \log^2 |\mathbf{A}_{ij}|$$

para encontrar os fatores de *scaling* η_i e ξ_j para linhas e colunas, onde $i = 1, 2, \dots, m$ e $j = 1, 2, \dots, n$, tais que a versão escalada de \mathbf{A} (cujos elementos são $\mathbf{A}_{ij}/\eta_i\xi_j$) minimizam a função objetivo acima. O método dos gradientes conjugados se mostra muito eficaz quando aplicado ao problema dos mínimos quadrados de encontrar os fatores i e j , e a convergência para uma solução aproximada de precisão adequada é, geralmente, obtida em poucas iterações (CZYZYK et al., 1997; OLIVEIRA, 1997).

3.3.4 Escolha do ponto inicial

Wright (1997, p. 224) argumenta que boas escolhas para pontos iniciais dos diversos métodos primais-duais devem satisfazer duas importantes condições. A primeira delas diz respeito à centralidade do ponto quanto ao caminho central \mathcal{C} : os pares de produtos $x_i^{(0)} s_i^{(0)}$ devem ser semelhantes. A segunda trata da conveniência da infactibilidade apresentada pelo ponto: a razão

$$\frac{\|(r_p^{(0)}, r_d^{(0)})\|}{\mu_{(0)}}$$

de infactibilidade para a medida de dualidade não deve ser muito grande.

O ponto inicial usado pelo algoritmo PCx utiliza uma heurística que faz uso dos pontos \tilde{x} , \tilde{s} , \tilde{v} e \tilde{t} de norma-2 mínima para os quais os resíduos r_p , r_d e r_u são nulos. Esta heurística foi proposta por Mehrotra (1992, p. 19) e é resumida como segue: primeiramente são calculados os pontos

$$\begin{aligned}\tilde{y} &= (\mathbf{A}\mathbf{A}^\top)^{-1}\mathbf{A}c, \\ \tilde{s} &= 0,5(c - \mathbf{A}^\top \tilde{y}), \\ \tilde{t} &= -\tilde{s}, \\ \tilde{x} &= \frac{u}{2} - \mathbf{A}^\top (\mathbf{A}\mathbf{A}^\top)^{-1}(\mathbf{A}\frac{u}{2} - b), \\ \tilde{v} &= u - \tilde{x}.\end{aligned}$$

Em seguida, calcula-se

$$\begin{aligned}\bar{\delta}_p &= \max \left\{ -1,5 \min_i \{\tilde{x}_i\}, -1,5 \min_i \{\tilde{v}_i\}, 0 \right\}, \\ \bar{\delta}_d &= \max \left\{ -1,5 \min_i \{\tilde{s}_i\}, -1,5 \min_i \{\tilde{t}_i\}, 0 \right\},\end{aligned}$$

para, então, obter

$$\begin{aligned}\delta_p &= \bar{\delta}_p + 0,5 \frac{(\tilde{x} + \bar{\delta}_p e)^\top (\tilde{s} + \bar{\delta}_d e) + (\tilde{v} + \bar{\delta}_p e)^\top (\tilde{t} + \bar{\delta}_d e)}{\sum_{i=1}^n (\tilde{s}_i + \bar{\delta}_d + \tilde{t}_i + \bar{\delta}_d)}, \\ \delta_d &= \bar{\delta}_d + 0,5 \frac{(\tilde{x} + \bar{\delta}_p e)^\top (\tilde{s} + \bar{\delta}_d e) + (\tilde{v} + \bar{\delta}_p e)^\top (\tilde{t} + \bar{\delta}_d e)}{\sum_{i=1}^n (\tilde{x}_i + \bar{\delta}_p + \tilde{v}_i + \bar{\delta}_p)}.\end{aligned}$$

Finalmente, o ponto inicial é obtido fazendo

$$\begin{aligned}y^{(0)} &= \tilde{y}, \\ x^{(0)} &= \tilde{x} + \delta_p e, \\ v^{(0)} &= \tilde{v} + \delta_p e, \\ s^{(0)} &= \tilde{s} + \delta_d e, \\ t^{(0)} &= \tilde{t} + \delta_d e.\end{aligned}$$

Mehrotra (1992, p. 19) mostra a validade de sua abordagem para geração do ponto inicial para o método, mostrando que $(x^{(0)}, s^{(0)}, v^{(0)}, t^{(0)}) > 0$.

De acordo com o próprio autor, o procedimento não é influenciado por *scaling* simples ou *shifts* nas variáveis. Ressalta-se que o *scaling* de colunas e a presença de restrições redundantes afetam o ponto inicial proposto, com atenção especial para o *scaling* de colunas, uma vez que testes computacionais exibiram um aumento do número de iterações em 40%, enquanto que alguns problemas não tiveram mudanças significativas no número de iterações.

O autor salienta, ainda, que a escolha das constantes 1,5 e 0,5 foi arbitrária e que qualquer constante maior que 1 pode substituir o valor 1,5, bem como qualquer constante maior que 0 pode substituir o valor 0,5.

3.3.5 Critério de parada

De acordo com Wright (1997, p. 225), os algoritmos dos métodos de pontos interiores não encontram uma solução ótima exata para um problema de programação linear, de modo que critérios de parada devem ser usados para avaliar a conveniência da aproximação da solução apresentada pela iteração atual com o conjunto solução. A maioria dos algoritmos utilizam informações dos resíduos e da medida de dualidade para comparar com valores relativamente pequenos configurados previamente. Um teste típico para aceitar o ponto (x, y, s) como uma solução aproximada consiste nas seguintes condições:

$$\frac{\|r_p\|}{1 + \|b\|} = \frac{\|b - Ax\|}{1 + \|b\|} \leq \epsilon, \quad (3.13a)$$

$$\frac{\|r_d\|}{1 + \|c\|} = \frac{\|c - A^\top y - s\|}{1 + \|c\|} \leq \epsilon, \quad (3.13b)$$

$$\frac{|c^\top x - b^\top y|}{1 + |c^\top x|} \leq \epsilon, \quad (3.13c)$$

$$(3.13d)$$

onde ϵ é um número pequeno, em geral 10^{-8} . Para o caso implementado neste trabalho, foi usado o critério de parada do PCx, que verifica

$$\frac{\|(r_p^\top, r_u^\top)\|}{1 + \|(b^\top, u^\top)\|} \leq \epsilon, \quad (3.14a)$$

$$\frac{\|r_d\|}{1 + \|c\|} \leq \epsilon, \quad (3.14b)$$

$$\frac{|c^\top x - (b^\top y - u^\top t)|}{1 + |c^\top x|} \leq \epsilon. \quad (3.14c)$$

$$(3.14d)$$

Czyzyk et al. (1997, p.7) afirmam que o PCx utiliza uma função auxiliar definida como

$$\phi(x, y, s, v, t) = \frac{\|(r_p^\top, r_u^\top)\|}{\max\{1, \|(b^\top, u^\top)\|\}} + \frac{\|r_d\|}{\max\{1, \|c\|\}} + \frac{|c^\top x - (b^\top y - \sum_{i \in \mathcal{B}_{u_i} t_i})|}{\max\{1, \|(b^\top, u^\top)\|, \|c\|\}} \quad (3.15)$$

para avaliar as condições para declaração de estado *infactível*, *desconhecido* e *subótimo* para o problema em questão.

Os pontos (x, y, s, v, t) que satisfazem $(x, s, v, t) \geq 0$ e $\phi = 0$ são soluções primais-duais de (3.6) e (3.7) (e vice-versa). Para problemas factíveis, ϕ gera uma sequência decrescente para zero (a menos das primeiras poucas iterações, cujos valores oscilam). Sendo assim, o PCx armazena os valores de ϕ em um vetor $\phi[k]$ de modo que suas entradas correspondam a uma sequência decrescente à medida que k aumenta, ou seja,

$$\phi_{\min}[k] = \min_{l=1,2,\dots,k} \phi(x_l, y_l, s_l, v_l, t_l).$$

Por sua vez, problemas infactíveis são facilmente detectados por um aumento acentuado nestes valores. Assim, o PCx declara um problema como infactível na iteração k se este aumento ocorre e se

$$\phi(x_k, y_k, s_k, v_k, t_k) \geq \max\{10^{-8}, 10^5 \phi_{\min}[k]\}.$$

Podem existir situações em que há convergência lenta ou que a melhora na medida de dualidade μ supera a melhoria na infactibilidade primal e dual (respectivamente $\|(r_p^\top, r_u^\top)\|$ e $\|r_d\|$), fazendo com que μ perca a verdadeira relação com o *gap* entre os valores objetivos primal e dual. Assim, o PCx encerra sua execução com o estado *desconhecido*. Para isso, usa o teste de convergência

$$\phi_{\min}[k - 30] \geq 1/2 (\phi_{\min}[k]) \quad k \geq 30.$$

É indicado um aumento na relação infactibilidade–dualidade se ocorrer

$$\frac{\|([r_p^{(k)}]^\top, [r_u^{(k)}]^\top)\|}{1 + \|(b^\top, u^\top)\|} > \epsilon, \quad \text{ou} \quad \frac{\|r_d^{(k)}\|}{1 + \|c\|} > \epsilon,$$

e adicionalmente

$$\frac{\max \left\{ \left\| ([r_p^{(k)}]^\top, [r_u^{(k)}]^\top) \right\|, \|r_d^{(k)}\| \right\} / \mu_{(k)}}{\max \left\{ \left\| ([r_p^{(0)}]^\top, [r_u^{(0)}]^\top) \right\|, \|r_d^{(k)}\| \right\} / \mu_{(0)}} \geq 10^6.$$

Finalmente, o estado *subótimo* é declarado se o algoritmo excede o número máximo de iterações sem satisfazer quaisquer das condições anteriores.

3.4 Considerações sobre métodos iterativos

Nesta seção, apresentamos questões que devem ser consideradas para trabalhar com a matriz $-\mathbf{I}_n + \mathbf{D}^{-\frac{1}{2}}\mathbf{A}^\top\mathbf{J}^\top + \mathbf{JAD}^{-\frac{1}{2}}$. Por simplicidade, consideremos \mathbf{L} , \mathbf{U} e \mathbf{V} tal como definidas para (2.9).

Uma vez que os métodos iterativos utilizam as matrizes apenas para realizar produtos matriz-vetor e devido à estrutura da matriz \mathbf{L} , podemos reformular o produto $w = \mathbf{L}x$ de modo a obter operações mais estáveis. Reescrevendo $x = x_{\mathcal{R}(\mathbf{V})} + x_{\mathcal{N}(\mathbf{V}^\top)}$, onde $x_{\mathcal{R}(\mathbf{V})} \in \mathcal{R}(\mathbf{V})$ e $x_{\mathcal{N}(\mathbf{V}^\top)} \in \mathcal{N}(\mathbf{V}^\top)$, obtemos

$$\begin{aligned}\mathbf{L}x &= -x + \mathbf{U}^\top\mathbf{V}^\top x_{\mathcal{R}(\mathbf{V})} + \mathbf{V}\mathbf{U}x \\ &= -x_{\mathcal{N}(\mathbf{V}^\top)} + \mathbf{U}^\top\mathbf{V}^\top x_{\mathcal{R}(\mathbf{V})} + \mathbf{V}\mathbf{U}x_{\mathcal{N}(\mathbf{V}^\top)}\end{aligned}$$

De fato, uma vez que $\mathbf{U}, \mathbf{V}^\top \in \mathbb{M}_{m \times n}(\mathbb{R})$ são tais que $\mathbf{UV} = \mathbf{I}_m$, então $\mathbf{V}\mathbf{U}$ é uma projeção sobre $\mathcal{R}(\mathbf{V})$. Assim, se $x \in \mathcal{R}(\mathbf{V})$, então $\mathbf{V}\mathbf{U}x = x$. Agora, como $\mathbf{V}^\top = [\mathbf{I}_m \ \mathbf{0}]\mathbf{P}$ seus espaços nulo e imagem podem facilmente ser representados em um código e todo o cálculo pode ser feito utilizando alguns índices. Com esta estratégia, n operações de ponto flutuante são economizadas (OLIVEIRA; SORENSEN, 2005).

3.4.1 Recuperação da solução: estratégia inicial

Neste trabalho, temos como objetivo resolver o sistema (3.10) utilizando o método de Gradientes Conjugados Pré-condicionado. Para isso, conforme exibido no capítulo anterior, foi construído o pré-condicionador

$$\mathbf{M}^{-1} = \begin{bmatrix} \mathbf{D}^{-\frac{1}{2}} & \mathbf{J} \\ \mathbf{H} & \mathbf{0} \end{bmatrix},$$

com $\mathbf{J} = \mathbf{H}^\top\mathbf{D}_B^{\frac{1}{2}}\mathbf{B}^{-1}$, $\mathbf{H}\mathbf{P}^\top = [\mathbf{I}_m \ \mathbf{0}]$ e $\mathbf{A}\mathbf{P}^\top = [\mathbf{B} \ \mathbf{N}]$, para que possamos resolver o seguinte sistema pré-condicionado

$$\mathbf{M}^{-1} \begin{bmatrix} -\mathbf{D} & \mathbf{A}^\top \\ \mathbf{A} & \mathbf{0} \end{bmatrix} \mathbf{M}^{-\top} \mathbf{M}^\top \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \mathbf{M}^{-1} \begin{bmatrix} r \\ r_p \end{bmatrix}. \quad (3.16)$$

Fazendo as alterações

$$\mathbf{M}^\top \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} dx \\ dy \end{bmatrix} \quad \text{e} \quad \mathbf{M}^{-1} \begin{bmatrix} r \\ r_p \end{bmatrix} = \begin{bmatrix} \mathbf{D}^{-\frac{1}{2}}r + \mathbf{P}^\top \begin{bmatrix} \mathbf{I}_m \\ \mathbf{0} \end{bmatrix} \mathbf{D}_B^{\frac{1}{2}}\mathbf{B}^{-1}r_p \\ \begin{bmatrix} \mathbf{I}_m & \mathbf{0} \end{bmatrix} \mathbf{P}r \end{bmatrix} = \begin{bmatrix} r_1 \\ r_2 \end{bmatrix}, \quad (3.17)$$

podemos perceber que o sistema (3.16) é equivalente ao sistema

$$\begin{bmatrix} -\mathbf{I}_n + \mathbf{D}^{-\frac{1}{2}}\mathbf{A}^\top\mathbf{J}^\top + \mathbf{JAD}^{-\frac{1}{2}} & \mathbf{0} \\ \mathbf{0} & -\mathbf{D}_B \end{bmatrix} \begin{bmatrix} dx \\ dy \end{bmatrix} = \begin{bmatrix} r_1 \\ r_2 \end{bmatrix}. \quad (3.18)$$

Aqui, é possível obter a solução dy do sistema acima fazendo

$$dy = -\mathbf{D}_B^{-1}r_2 = -\mathbf{D}_B^{-1} \begin{bmatrix} \mathbf{I}_m & \mathbf{0} \end{bmatrix} \mathbf{P}r. \quad (3.19)$$

Para obtermos a solução dx relembremos da redução a sistemas definidos positivos que a expressão $-\mathbf{I}_n + \mathbf{D}^{-\frac{1}{2}}\mathbf{A}^\top\mathbf{J}^\top + \mathbf{JAD}^{-\frac{1}{2}}$ pode ser reescrita como

$$\mathbf{P}^\top \begin{bmatrix} \mathbf{I}_m & \mathbf{D}_B^{\frac{1}{2}}\mathbf{B}^{-1}\mathbf{ND}_N^{-\frac{1}{2}} \\ \mathbf{D}_N^{-\frac{1}{2}}\mathbf{N}^\top\mathbf{B}^{-\top}\mathbf{D}_B^{\frac{1}{2}} & -\mathbf{I}_{n-m} \end{bmatrix} \mathbf{P},$$

para obtermos o seguinte sistema

$$\mathbf{P}^\top \begin{bmatrix} \mathbf{I}_m & \mathbf{D}_B^{\frac{1}{2}}\mathbf{B}^{-1}\mathbf{ND}_N^{-\frac{1}{2}} \\ \mathbf{D}_N^{-\frac{1}{2}}\mathbf{N}^\top\mathbf{B}^{-\top}\mathbf{D}_B^{\frac{1}{2}} & -\mathbf{I}_{n-m} \end{bmatrix} \mathbf{P}dx = r_1. \quad (3.20)$$

Neste caso, resolvemos um sistema equivalente, pondo $\mathbf{P}dx = d\hat{x}$ e $\mathbf{P}r_1 = \hat{r}$, obtendo

$$\begin{bmatrix} \mathbf{I}_m & \mathbf{D}_B^{\frac{1}{2}}\mathbf{B}^{-1}\mathbf{ND}_N^{-\frac{1}{2}} \\ \mathbf{D}_N^{-\frac{1}{2}}\mathbf{N}^\top\mathbf{B}^{-\top}\mathbf{D}_B^{\frac{1}{2}} & -\mathbf{I}_{n-m} \end{bmatrix} \begin{bmatrix} d\hat{x}_1 \\ d\hat{x}_2 \end{bmatrix} = \begin{bmatrix} \hat{r}_1 \\ \hat{r}_2 \end{bmatrix}, \quad (3.21)$$

onde os vetores $d\hat{x}$ e \hat{r} foram apropriadamente particionados. Observe que

$$\begin{aligned} \hat{r} = \mathbf{P}r_1 &= \mathbf{P} \left[\mathbf{D}^{-\frac{1}{2}}r + \mathbf{P}^\top \begin{bmatrix} \mathbf{I}_m \\ \mathbf{0} \end{bmatrix} \mathbf{D}_B^{\frac{1}{2}}\mathbf{B}^{-1}r_p \right] = \mathbf{PD}^{-\frac{1}{2}}r + \begin{bmatrix} \mathbf{I}_m \\ \mathbf{0} \end{bmatrix} \mathbf{D}_B^{\frac{1}{2}}\mathbf{B}^{-1}r_p = \\ &= \mathbf{PD}^{-\frac{1}{2}}r + \begin{bmatrix} \mathbf{D}_B^{\frac{1}{2}}\mathbf{B}^{-1}r_p \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} \hat{r}_1 \\ \hat{r}_2 \end{bmatrix}. \end{aligned}$$

O sistema (3.21) pode ser reescrito usando a notação $\mathbf{F} = \mathbf{D}_N^{-\frac{1}{2}}\mathbf{N}^\top\mathbf{B}^{-\top}\mathbf{D}_B^{\frac{1}{2}}$ como

$$\begin{bmatrix} \mathbf{I}_m & \mathbf{F}^\top \\ \mathbf{F} & -\mathbf{I}_{n-m} \end{bmatrix} \begin{bmatrix} d\hat{x}_1 \\ d\hat{x}_2 \end{bmatrix} = \begin{bmatrix} \hat{r}_1 \\ \hat{r}_2 \end{bmatrix}.$$

Desta forma, o sistema resultante tornou-se

$$\begin{cases} d\hat{x}_1 + \mathbf{F}^\top d\hat{x}_2 = \hat{r}_1 \\ \mathbf{F}d\hat{x}_1 - d\hat{x}_2 = \hat{r}_2 \end{cases},$$

cujas soluções podem ser obtidas fazendo

$$d\hat{x}_1 = \hat{r}_1 - \mathbf{F}^\top d\hat{x}_2, \quad (3.22)$$

que é usado na segunda igualdade do sistema

$$\begin{aligned} \mathbf{F}(\hat{r}_1 - \mathbf{F}^\top d\hat{x}_2) - d\hat{x}_2 &= \hat{r}_2 \\ \mathbf{F}\hat{r}_1 - \mathbf{F}\mathbf{F}^\top d\hat{x}_2 - d\hat{x}_2 &= \hat{r}_2 \\ (\mathbf{I}_{n-m} + \mathbf{F}\mathbf{F}^\top) d\hat{x}_2 &= \mathbf{F}\hat{r}_1 - \hat{r}_2. \end{aligned} \quad (3.23)$$

Agora, podemos obter $d\hat{x}_2$ resolvendo o sistema (3.23) usando o método de Gradientes Conjugados Pré-condicionado. Em seguida, substituímos o valor encontrado de $d\hat{x}_2$ em (3.22) para encontramos $d\hat{x}_1$. Desta forma, teremos encontrado as soluções para o sistema (3.21). Estas soluções juntas formam o vetor $d\hat{x}$, que por sua vez é usado na relação $\mathbf{P}dx = d\hat{x}$ para obtermos dx solução de (3.20). Note que obtemos dx e dy soluções de (3.18).

Por fim, calculamos as soluções Δx e Δy para (3.16) substituindo os valores de dx e dy na primeira igualdade de (3.17)

$$\begin{aligned} \mathbf{M}^\top \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} &= \begin{bmatrix} dx \\ dy \end{bmatrix} \Rightarrow \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \mathbf{M}^{-\top} \begin{bmatrix} dx \\ dy \end{bmatrix} \\ \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} &= \begin{bmatrix} \mathbf{D}^{-\frac{1}{2}} & \mathbf{H}^\top \\ \mathbf{J}^\top & \mathbf{0} \end{bmatrix} \begin{bmatrix} dx \\ dy \end{bmatrix} = \begin{bmatrix} \mathbf{D}^{-\frac{1}{2}}dx + \mathbf{P}^\top \begin{bmatrix} \mathbf{I}_m \\ \mathbf{0} \end{bmatrix} dy \\ \mathbf{B}^{-\top} \mathbf{D}_B^{\frac{1}{2}} \begin{bmatrix} \mathbf{I}_m & \mathbf{0} \end{bmatrix} \mathbf{P}dx \end{bmatrix} \\ \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} &= \begin{bmatrix} \mathbf{D}^{-\frac{1}{2}}dx + \mathbf{P}^\top \begin{bmatrix} dy \\ \mathbf{0} \end{bmatrix} \\ \mathbf{B}^{-\top} \mathbf{D}_B^{\frac{1}{2}} d\hat{x}_1 \end{bmatrix}. \end{aligned}$$

3.4.2 Recuperação da solução: primeira estratégia nova

Para que os testes computacionais da estratégia inicial de recuperação de solução fornecessem resultados otimistas, fez-se necessário alterar os parâmetros de precisão de solução inicialmente adotados pelo método de Gradientes Conjugados Pré-condicionado, cujos valores eram 10^{-4} para a direção preditora e 10^{-8} para a corretora: foram necessários os parâmetros 10^{-8} para a direção preditora e 10^{-12} para a corretora. Tal instabilidade pode ser verificada uma vez que a matriz pré-condicionada envolve cálculos com matrizes numericamente instáveis, tais como $\mathbf{D}_N^{\frac{1}{2}}$ e $\mathbf{D}_N^{-\frac{1}{2}}$. Uma alternativa elaborada para evitar estas mudanças foi construir um novo pré-condicionador tal como aquele fornecido por Oliveira e Sorensen (2005) tomando como base a matriz (2.11b), objetivando construir a matriz

$$\mathbf{D}_N + \mathbf{N}^\top \mathbf{B}^{-\top} \mathbf{D}_B \mathbf{B}^{-1} \mathbf{N} \quad (3.24)$$

e usando a matriz (2.10), multiplicando-a à esquerda e à direita por $\mathbf{D}^{\frac{1}{2}}$. Desta forma, a matriz em (2.10) fica

$$\mathbf{D}^{\frac{1}{2}} \mathbf{P}^\top \begin{bmatrix} \mathbf{I}_m & \mathbf{D}_B^{\frac{1}{2}} \mathbf{B}^{-1} \mathbf{N} \mathbf{D}_N^{-\frac{1}{2}} \\ \mathbf{D}_N^{-\frac{1}{2}} \mathbf{N}^\top \mathbf{B}^{-\top} \mathbf{D}_B^{\frac{1}{2}} & -\mathbf{I}_{n-m} \end{bmatrix} \mathbf{P} \mathbf{D}^{\frac{1}{2}} = \mathbf{P}^\top \begin{bmatrix} \mathbf{D}_B & \mathbf{D}_B \mathbf{B}^{-1} \mathbf{N} \\ \mathbf{N}^\top \mathbf{B}^{-\top} \mathbf{D}_B & -\mathbf{D}_N \end{bmatrix} \mathbf{P} = \bar{\mathbf{W}}. \quad (3.25)$$

Observe que a matriz à direita da igualdade nos fornece $\mathbf{D}_N + \mathbf{N}^\top \mathbf{B}^{-\top} \mathbf{D}_B \mathbf{B}^{-1} \mathbf{N}$. Já a matriz à esquerda da igualdade é equivalente a

$$\mathbf{D}^{\frac{1}{2}} \mathbf{P}^\top \left(-\mathbf{I}_n + \begin{bmatrix} \mathbf{D}_B^{-\frac{1}{2}} & \mathbf{0} \\ \mathbf{0} & \mathbf{D}_N^{-\frac{1}{2}} \end{bmatrix} \begin{bmatrix} \mathbf{D}_B^{\frac{1}{2}} & \mathbf{0} \\ \mathbf{N}^\top \mathbf{B}^{-\top} \mathbf{D}_B^{\frac{1}{2}} & \mathbf{0} \end{bmatrix} + \begin{bmatrix} \mathbf{D}_B^{\frac{1}{2}} & \mathbf{D}_B^{\frac{1}{2}} \mathbf{B}^{-1} \mathbf{N} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{D}_B^{-\frac{1}{2}} & \mathbf{0} \\ \mathbf{0} & \mathbf{D}_N^{-\frac{1}{2}} \end{bmatrix} \right) \mathbf{P} \mathbf{D}^{\frac{1}{2}}$$

que, por sua vez, equivale a

$$\begin{aligned} -\mathbf{D} + \mathbf{P}^\top & \begin{bmatrix} \mathbf{D}_B^{\frac{1}{2}} & \mathbf{0} \\ \mathbf{0} & \mathbf{D}_N^{\frac{1}{2}} \end{bmatrix} \begin{bmatrix} \mathbf{D}_B^{-\frac{1}{2}} & \mathbf{0} \\ \mathbf{0} & \mathbf{D}_N^{-\frac{1}{2}} \end{bmatrix} \begin{bmatrix} \mathbf{D}_B^{\frac{1}{2}} & \mathbf{0} \\ \mathbf{N}^\top \mathbf{B}^{-\top} \mathbf{D}_B^{\frac{1}{2}} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{D}_B^{\frac{1}{2}} & \mathbf{0} \\ \mathbf{0} & \mathbf{D}_N^{\frac{1}{2}} \end{bmatrix} \mathbf{P} + \\ & + \mathbf{P}^\top \begin{bmatrix} \mathbf{D}_B^{\frac{1}{2}} & \mathbf{0} \\ \mathbf{0} & \mathbf{D}_N^{\frac{1}{2}} \end{bmatrix} \begin{bmatrix} \mathbf{D}_B^{\frac{1}{2}} & \mathbf{D}_B^{\frac{1}{2}} \mathbf{B}^{-1} \mathbf{N} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{D}_B^{-\frac{1}{2}} & \mathbf{0} \\ \mathbf{0} & \mathbf{D}_N^{-\frac{1}{2}} \end{bmatrix} \begin{bmatrix} \mathbf{D}_B^{\frac{1}{2}} & \mathbf{0} \\ \mathbf{0} & \mathbf{D}_N^{\frac{1}{2}} \end{bmatrix} \mathbf{P}. \end{aligned}$$

A soma matricial acima pode, ainda, ser reescrita como

$$\begin{aligned} & -\mathbf{D} + \mathbf{P}^\top \begin{bmatrix} \mathbf{D}_B & \mathbf{0} \\ \mathbf{N}^\top \mathbf{B}^{-\top} \mathbf{D}_B & \mathbf{0} \end{bmatrix} \mathbf{P} + \mathbf{P}^\top \begin{bmatrix} \mathbf{D}_B & \mathbf{D}_B \mathbf{B}^{-1} \mathbf{N} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \mathbf{P} = \\ & = -\mathbf{D} + \mathbf{P}^\top \begin{bmatrix} \mathbf{I}_m \\ \mathbf{N}^\top \mathbf{B}^{-\top} \end{bmatrix} \mathbf{D}_B [\mathbf{I}_m \mathbf{0}] \mathbf{P} + \mathbf{P}^\top \begin{bmatrix} \mathbf{I}_m \\ \mathbf{0} \end{bmatrix} \mathbf{D}_B [\mathbf{I}_m \mathbf{B}^{-1} \mathbf{N}] \mathbf{P} = \\ & = -\mathbf{D} + \mathbf{P}^\top \begin{bmatrix} \mathbf{B}^\top \\ \mathbf{N}^\top \end{bmatrix} \mathbf{B}^{-\top} \mathbf{D}_B [\mathbf{I}_m \mathbf{0}] \mathbf{P} + \mathbf{P}^\top \begin{bmatrix} \mathbf{I}_m \\ \mathbf{0} \end{bmatrix} \mathbf{D}_B \mathbf{B}^{-1} [\mathbf{B} \mathbf{N}] \mathbf{P} = \\ & = -\mathbf{D} + \mathbf{A}^\top \mathbf{B}^{-\top} \mathbf{D}_B \mathbf{H} + \mathbf{H}^\top \mathbf{D}_B \mathbf{B}^{-1} \mathbf{A} = \bar{\mathbf{W}}. \end{aligned}$$

Sendo assim, se escolhermos

$$\mathbf{M}^{-1} = \begin{bmatrix} \mathbf{I}_n & \boldsymbol{\Theta} \\ \mathbf{H} & \mathbf{0} \end{bmatrix}, \quad (3.26)$$

onde $\mathbf{H} \mathbf{P}^\top = [\mathbf{I}_m \mathbf{0}]$ e $\mathbf{A} \mathbf{P}^\top = [\mathbf{B} \mathbf{N}]$, obteremos o novo sistema aumentado pré-condicionado

$$\mathbf{M}^{-1} \begin{bmatrix} -\mathbf{D} & \mathbf{A}^\top \\ \mathbf{A} & \mathbf{0} \end{bmatrix} \mathbf{M}^{-\top} = \begin{bmatrix} -\mathbf{D} + \mathbf{A}^\top \boldsymbol{\Theta}^\top + \boldsymbol{\Theta} \mathbf{A} & -\mathbf{D} \mathbf{H}^\top + \boldsymbol{\Theta} \mathbf{A} \mathbf{H}^\top \\ -\mathbf{H} \mathbf{D} + \mathbf{H} \mathbf{A}^\top \boldsymbol{\Theta}^\top & -\mathbf{H} \mathbf{D} \mathbf{H}^\top \end{bmatrix},$$

onde a escolha $\boldsymbol{\Theta} = \mathbf{H}^\top \mathbf{D}_B \mathbf{B}^{-1}$ nos leva a uma matriz pré-condicionada diagonal por blocos (tal como inicialmente obtido no trabalho de [Oliveira e Sorensen \(2005\)](#)).

A estratégia de recuperação de solução a partir da proposta desta subseção é resolver o seguinte sistema linear pré-condicionado

$$\mathbf{M}^{-1} \begin{bmatrix} -\mathbf{D} & \mathbf{A}^\top \\ \mathbf{A} & \mathbf{0} \end{bmatrix} \mathbf{M}^{-\top} \mathbf{M}^\top \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \mathbf{M}^{-1} \begin{bmatrix} r \\ r_p \end{bmatrix}, \quad (3.27)$$

onde, com as alterações abaixo

$$\mathbf{M}^\top \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} dx \\ dy \end{bmatrix} \quad \text{e} \quad \mathbf{M}^{-1} \begin{bmatrix} r \\ r_p \end{bmatrix} = \begin{bmatrix} r + \mathbf{P}^\top \begin{bmatrix} \mathbf{D}_B \mathbf{B}^{-1} r_p \\ \mathbf{0} \end{bmatrix} \\ \begin{bmatrix} \mathbf{I}_m & \mathbf{0} \end{bmatrix} \mathbf{P} r \end{bmatrix} = \begin{bmatrix} r_1 \\ r_2 \end{bmatrix}, \quad (3.28)$$

podemos reescrever o sistema (3.27) como

$$\begin{bmatrix} -\mathbf{D} + \mathbf{A}^\top \boldsymbol{\Theta}^\top + \boldsymbol{\Theta} \mathbf{A} & \mathbf{0} \\ \mathbf{0} & -\mathbf{D}_B \end{bmatrix} \begin{bmatrix} dx \\ dy \end{bmatrix} = \begin{bmatrix} r_1 \\ r_2 \end{bmatrix}. \quad (3.29)$$

Com estas alterações, obtemos a solução dy do sistema acima fazendo

$$dy = -\mathbf{D}_B^{-1}r_2 = -\mathbf{D}_B^{-1} \begin{bmatrix} \mathbf{I}_m & \mathbf{0} \end{bmatrix} \mathbf{P}r. \quad (3.30)$$

Para obtermos a solução dx devemos resolver $[-\mathbf{D} + \mathbf{A}^\top \boldsymbol{\Theta}^\top + \boldsymbol{\Theta} \mathbf{A}] dx = r_1$ que é equivalente a resolver

$$\mathbf{P}^\top \begin{bmatrix} \mathbf{D}_B & \mathbf{D}_B \mathbf{B}^{-1} \mathbf{N} \\ \mathbf{N}^\top \mathbf{B}^{-\top} \mathbf{D}_B & -\mathbf{D}_N \end{bmatrix} \mathbf{P} dx = r_1 \quad (3.31)$$

Neste caso, resolvemos um sistema equivalente, pondo $\mathbf{P}dx = d\hat{x}$ e $\mathbf{P}r_1 = \hat{r}$, obtendo

$$\begin{bmatrix} \mathbf{D}_B & \mathbf{D}_B \mathbf{B}^{-1} \mathbf{N} \\ \mathbf{N}^\top \mathbf{B}^{-\top} \mathbf{D}_B & -\mathbf{D}_N \end{bmatrix} \begin{bmatrix} d\hat{x}_1 \\ d\hat{x}_2 \end{bmatrix} = \begin{bmatrix} \hat{r}_1 \\ \hat{r}_2 \end{bmatrix}, \quad (3.32)$$

onde os vetores $d\hat{x}$ e \hat{r} foram apropriadamente particionados. Observe que

$$\hat{r} = \mathbf{P}r_1 = \mathbf{P}r + \begin{bmatrix} \mathbf{D}_B \mathbf{B}^{-1} r_p \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} \bar{r}_1 + \mathbf{D}_B \mathbf{B}^{-1} r_p \\ \bar{r}_2 \end{bmatrix} = \begin{bmatrix} \hat{r}_1 \\ \hat{r}_2 \end{bmatrix},$$

uma vez que consideramos $\bar{r} = \mathbf{P}r$ adequadamente particionado. Assim, fazemos

$$\begin{cases} \mathbf{D}_B d\hat{x}_1 + \mathbf{D}_B \mathbf{B}^{-1} \mathbf{N} d\hat{x}_2 = \hat{r}_1 \\ \mathbf{N}^\top \mathbf{B}^{-\top} \mathbf{D}_B d\hat{x}_1 - \mathbf{D}_N d\hat{x}_2 = \hat{r}_2. \end{cases}$$

Obtemos $d\hat{x}_1$ realizando os seguintes passos

$$d\hat{x}_1 = \mathbf{D}_B^{-1} \hat{r}_1 - \mathbf{B}^{-1} \mathbf{N} d\hat{x}_2 = \mathbf{D}_B^{-1} \bar{r}_1 + \mathbf{B}^{-1} (r_p - \mathbf{N} d\hat{x}_2), \quad (3.33)$$

que, por sua vez, é usado na segunda igualdade do sistema

$$\begin{aligned} \mathbf{N}^\top \mathbf{B}^{-\top} \mathbf{D}_B [\mathbf{D}_B^{-1} \bar{r}_1 + \mathbf{B}^{-1} (r_p - \mathbf{N} d\hat{x}_2)] - \mathbf{D}_N d\hat{x}_2 &= \hat{r}_2 \\ \mathbf{N}^\top \mathbf{B}^{-\top} \bar{r}_1 + \mathbf{N}^\top \mathbf{B}^{-\top} \mathbf{D}_B \mathbf{B}^{-1} r_p - \mathbf{N}^\top \mathbf{B}^{-\top} \mathbf{D}_B \mathbf{B}^{-1} \mathbf{N} d\hat{x}_2 - \mathbf{D}_N d\hat{x}_2 &= \hat{r}_2, \end{aligned}$$

resultando em

$$(\mathbf{D}_N + \mathbf{N}^\top \mathbf{B}^{-\top} \mathbf{D}_B \mathbf{B}^{-1} \mathbf{N}) d\hat{x}_2 = \mathbf{N}^\top \mathbf{B}^{-\top} (\bar{r}_1 + \mathbf{D}_B \mathbf{B}^{-1} r_p) - \bar{r}_2. \quad (3.34)$$

Com esta configuração, obtemos $d\hat{x}_2$ resolvendo o sistema (3.34) a partir do método de Gradientes Conjugados Pré-condicionado. Em seguida, substituímos o valor encontrado de $d\hat{x}_2$ em (3.33) para encontramos $d\hat{x}_1$ e montamos a solução de (3.32). Usando a relação $\mathbf{P}dx = d\hat{x}$ obtemos dx solução de (3.31), que juntamente com dy , obtido de (3.30), forma a solução para (3.29).

Para obtermos as soluções Δx e Δy fazemos, primeiramente

$$\begin{aligned}
 \mathbf{M}^\top \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} &= \begin{bmatrix} dx \\ dy \end{bmatrix} \Rightarrow \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \mathbf{M}^{-\top} \begin{bmatrix} dx \\ dy \end{bmatrix} \\
 \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} &= \begin{bmatrix} \mathbf{I}_n & \mathbf{H}^\top \\ \boldsymbol{\Theta}^\top & \mathbf{0} \end{bmatrix} \begin{bmatrix} dx \\ dy \end{bmatrix} = \begin{bmatrix} dx + \mathbf{P}^\top \begin{bmatrix} \mathbf{I}_m \\ \mathbf{0} \end{bmatrix} dy \\ \mathbf{B}^{-\top} \mathbf{D}_B \begin{bmatrix} \mathbf{I}_m & \mathbf{0} \end{bmatrix} \mathbf{P} dx \end{bmatrix} \\
 \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} &= \begin{bmatrix} \mathbf{P}^\top \begin{bmatrix} d\hat{x}_1 \\ d\hat{x}_2 \end{bmatrix} + \mathbf{P}^\top \begin{bmatrix} dy \\ \mathbf{0} \end{bmatrix} \\ \mathbf{B}^{-\top} \mathbf{D}_B d\hat{x}_1 \end{bmatrix} = \begin{bmatrix} \mathbf{P}^\top \begin{bmatrix} d\hat{x}_1 + dy \\ d\hat{x}_2 \end{bmatrix} \\ \mathbf{B}^{-\top} \mathbf{D}_B d\hat{x}_1 \end{bmatrix} \\
 \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} &= \begin{bmatrix} \mathbf{P}^\top \begin{bmatrix} \mathbf{D}_B^{-1} \bar{r}_1 + \mathbf{B}^{-1} (r_p - \mathbf{N} d\hat{x}_2) - \mathbf{D}_B^{-1} [\mathbf{I}_m \mathbf{0}] \mathbf{P} r \\ d\hat{x}_2 \end{bmatrix} \\ \mathbf{B}^{-\top} \mathbf{D}_B [\mathbf{D}_B^{-1} \bar{r}_1 + \mathbf{B}^{-1} (r_p - \mathbf{N} d\hat{x}_2)] \end{bmatrix} \\
 \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} &= \begin{bmatrix} \mathbf{P}^\top \begin{bmatrix} \mathbf{B}^{-1} (r_p - \mathbf{N} d\hat{x}_2) \\ d\hat{x}_2 \end{bmatrix} \\ \mathbf{B}^{-\top} [\bar{r}_1 + \mathbf{D}_B \mathbf{B}^{-1} (r_p - \mathbf{N} d\hat{x}_2)] \end{bmatrix} \tag{3.35}
 \end{aligned}$$

Observe que as expressões de Δx e Δy em (3.35) só dependem de $d\hat{x}_2$ que é obtido em (3.34) via método de Gradientes Conjugados Pré-Condicionado. Sendo assim, a programação fica restrita à solução do sistema em (3.34) e às expressões de Δx e Δy em (3.35).

3.4.3 Recuperação da solução: segunda estratégia nova

Podemos obter uma nova matriz pré-condicionadora tomando como base a matriz (2.11b), objetivando construir a matriz

$$\mathbf{D}_N^{-1} + \mathbf{D}_N^{-1} \mathbf{N}^\top \mathbf{B}^{-\top} \mathbf{D}_B \mathbf{B}^{-1} \mathbf{N} \mathbf{D}_N^{-1} \tag{3.36}$$

e usando a matriz (2.10), multiplicando-a à esquerda e à direita por $\mathbf{D}^{-\frac{1}{2}}$. Desta forma, a matriz em (2.10) fica

$$\mathbf{D}^{-\frac{1}{2}} \mathbf{P}^\top \begin{bmatrix} \mathbf{I}_m & \mathbf{D}_B^{\frac{1}{2}} \mathbf{B}^{-1} \mathbf{N} \mathbf{D}_N^{-\frac{1}{2}} \\ \mathbf{D}_N^{-\frac{1}{2}} \mathbf{N}^\top \mathbf{B}^{-\top} \mathbf{D}_B^{\frac{1}{2}} & -\mathbf{I}_{n-m} \end{bmatrix} \mathbf{P} \mathbf{D}^{-\frac{1}{2}} = \mathbf{P}^\top \begin{bmatrix} \mathbf{D}_B^{-1} & \mathbf{B}^{-1} \mathbf{N} \mathbf{D}_N^{-1} \\ \mathbf{D}_N^{-1} \mathbf{N}^\top \mathbf{B}^{-\top} & -\mathbf{D}_N^{-1} \end{bmatrix} \mathbf{P} = \widehat{\mathbf{W}}. \tag{3.37}$$

A partir da matriz à direita da igualdade obtemos $\mathbf{D}_N^{-1} + \mathbf{D}_N^{-1} \mathbf{N}^\top \mathbf{B}^{-\top} \mathbf{D}_B \mathbf{B}^{-1} \mathbf{N} \mathbf{D}_N^{-1}$. Já a partir da matriz à esquerda da igualdade podemos escrever

$$\mathbf{D}^{-\frac{1}{2}} \mathbf{P}^\top \left(-\mathbf{I}_n + \begin{bmatrix} \mathbf{D}_B^{-\frac{1}{2}} & \mathbf{0} \\ \mathbf{0} & \mathbf{D}_N^{-\frac{1}{2}} \end{bmatrix} \begin{bmatrix} \mathbf{D}_B^{\frac{1}{2}} & \mathbf{0} \\ \mathbf{N}^\top \mathbf{B}^{-\top} \mathbf{D}_B^{\frac{1}{2}} & \mathbf{0} \end{bmatrix} + \begin{bmatrix} \mathbf{D}_B^{\frac{1}{2}} & \mathbf{D}_B^{\frac{1}{2}} \mathbf{B}^{-1} \mathbf{N} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{D}_B^{-\frac{1}{2}} & \mathbf{0} \\ \mathbf{0} & \mathbf{D}_N^{-\frac{1}{2}} \end{bmatrix} \right) \mathbf{P} \mathbf{D}^{-\frac{1}{2}}$$

que, por sua vez, equivale a

$$\begin{aligned}
 -\mathbf{D}^{-1} &+ \mathbf{P}^\top \begin{bmatrix} \mathbf{D}_B^{-\frac{1}{2}} & \mathbf{0} \\ \mathbf{0} & \mathbf{D}_N^{-\frac{1}{2}} \end{bmatrix} \begin{bmatrix} \mathbf{D}_B^{-\frac{1}{2}} & \mathbf{0} \\ \mathbf{0} & \mathbf{D}_N^{-\frac{1}{2}} \end{bmatrix} \begin{bmatrix} \mathbf{D}_B^{\frac{1}{2}} & \mathbf{0} \\ \mathbf{N}^\top \mathbf{B}^{-\top} \mathbf{D}_B^{\frac{1}{2}} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{D}_B^{-\frac{1}{2}} & \mathbf{0} \\ \mathbf{0} & \mathbf{D}_N^{-\frac{1}{2}} \end{bmatrix} \mathbf{P} + \\
 &+ \mathbf{P}^\top \begin{bmatrix} \mathbf{D}_B^{-\frac{1}{2}} & \mathbf{0} \\ \mathbf{0} & \mathbf{D}_N^{-\frac{1}{2}} \end{bmatrix} \begin{bmatrix} \mathbf{D}_B^{\frac{1}{2}} & \mathbf{D}_B^{\frac{1}{2}} \mathbf{B}^{-1} \mathbf{N} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{D}_B^{-\frac{1}{2}} & \mathbf{0} \\ \mathbf{0} & \mathbf{D}_N^{-\frac{1}{2}} \end{bmatrix} \begin{bmatrix} \mathbf{D}_B^{-\frac{1}{2}} & \mathbf{0} \\ \mathbf{0} & \mathbf{D}_N^{-\frac{1}{2}} \end{bmatrix} \mathbf{P}.
 \end{aligned}$$

A soma acima pode, ainda, ser reescrita como

$$\begin{aligned}
 -\mathbf{D}^{-1} + \mathbf{P}^\top \begin{bmatrix} \mathbf{D}_B^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{D}_N^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{I}_m & \mathbf{0} \\ \mathbf{N}^\top \mathbf{B}^{-\top} & \mathbf{0} \end{bmatrix} \mathbf{P} + \mathbf{P}^\top \begin{bmatrix} \mathbf{I}_m & \mathbf{B}^{-1} \mathbf{N} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{D}_B^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{D}_N^{-1} \end{bmatrix} \mathbf{P} &= \\
 = -\mathbf{D}^{-1} + \mathbf{D}^{-1} \mathbf{P}^\top \begin{bmatrix} \mathbf{B}^\top \\ \mathbf{N}^\top \end{bmatrix} \mathbf{B}^{-\top} [\mathbf{I}_m \mathbf{0}] \mathbf{P} + \mathbf{P}^\top \begin{bmatrix} \mathbf{I}_m \\ \mathbf{0} \end{bmatrix} \mathbf{B}^{-1} [\mathbf{B} \mathbf{N}] \mathbf{P} \mathbf{D}^{-1} &= \\
 = -\mathbf{D}^{-1} + \mathbf{D}^{-1} \mathbf{A}^\top \mathbf{B}^{-\top} \mathbf{H} + \mathbf{H}^\top \mathbf{B}^{-1} \mathbf{A} \mathbf{D}^{-1} &= \\
 = \widehat{\mathbf{W}}. &
 \end{aligned}$$

Sendo assim, se escolhermos

$$\mathbf{M}^{-1} = \begin{bmatrix} \mathbf{D}^{-1} & \mathbf{\Gamma} \\ \mathbf{H} & \mathbf{0} \end{bmatrix}, \quad (3.38)$$

onde $\mathbf{H} \mathbf{P}^\top = [\mathbf{I}_m \mathbf{0}]$ e $\mathbf{A} \mathbf{P}^\top = [\mathbf{B} \mathbf{N}]$, obteremos o novo sistema aumentado pré-condicionado

$$\mathbf{M}^{-1} \begin{bmatrix} -\mathbf{D} & \mathbf{A}^\top \\ \mathbf{A} & \mathbf{0} \end{bmatrix} \mathbf{M}^{-\top} = \begin{bmatrix} -\mathbf{D}^{-1} + \mathbf{D}^{-1} \mathbf{A}^\top \mathbf{\Gamma}^\top + \mathbf{\Gamma} \mathbf{A} \mathbf{D}^{-1} & -\mathbf{H}^\top + \mathbf{\Gamma} \mathbf{A} \mathbf{H}^\top \\ -\mathbf{H} + \mathbf{H} \mathbf{A}^\top \mathbf{\Gamma}^\top & -\mathbf{H} \mathbf{D} \mathbf{H}^\top \end{bmatrix},$$

onde a escolha $\mathbf{\Gamma} = \mathbf{H}^\top \mathbf{B}^{-1}$ nos leva a uma matriz pré-condicionada diagonal por blocos.

A estratégia de recuperação de solução a partir da proposta desta subseção é resolver o seguinte sistema linear pré-condicionado

$$\mathbf{M}^{-1} \begin{bmatrix} -\mathbf{D} & \mathbf{A}^\top \\ \mathbf{A} & \mathbf{0} \end{bmatrix} \mathbf{M}^{-\top} \mathbf{M}^\top \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \mathbf{M}^{-1} \begin{bmatrix} r \\ r_p \end{bmatrix}. \quad (3.39)$$

Considerando $\bar{r} = \mathbf{P} r$ adequadamente particionado e as alterações abaixo

$$\mathbf{M}^\top \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} dx \\ dy \end{bmatrix} \quad \text{e} \quad \mathbf{M}^{-1} \begin{bmatrix} r \\ r_p \end{bmatrix} = \begin{bmatrix} \mathbf{P}^\top \begin{bmatrix} \mathbf{D}_B^{-1} \bar{r}_1 + \mathbf{B}^{-1} r_p \\ \mathbf{D}_N^{-1} \bar{r}_2 \\ \bar{r}_1 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} r_1 \\ r_2 \end{bmatrix}, \quad (3.40)$$

podemos reescrever o sistema (3.39) como

$$\begin{bmatrix} -\mathbf{D}^{-1} + \mathbf{\Gamma} \mathbf{A} \mathbf{D}^{-1} + \mathbf{D}^{-1} \mathbf{A}^\top \mathbf{\Gamma}^\top & \mathbf{0} \\ \mathbf{0} & -\mathbf{D}_B \end{bmatrix} \begin{bmatrix} dx \\ dy \end{bmatrix} = \begin{bmatrix} r_1 \\ r_2 \end{bmatrix}. \quad (3.41)$$

Com estas alterações, obtemos a solução dy do sistema acima fazendo

$$dy = -\mathbf{D}_B^{-1} r_2 = -\mathbf{D}_B^{-1} \bar{r}_1. \quad (3.42)$$

Prosseguindo, resta-nos resolver $[-\mathbf{D}^{-1} + \mathbf{\Gamma} \mathbf{A} \mathbf{D}^{-1} + \mathbf{D}^{-1} \mathbf{A}^\top \mathbf{\Gamma}^\top] dx = r_1$ para encontrarmos dx . Mas isto que é equivalente a resolver

$$\mathbf{P}^\top \begin{bmatrix} \mathbf{D}_B^{-1} & \mathbf{B}^{-1} \mathbf{N} \mathbf{D}_N^{-1} \\ \mathbf{D}_N^{-1} \mathbf{N}^\top \mathbf{B}^{-\top} & -\mathbf{D}_N^{-1} \end{bmatrix} \mathbf{P} dx = r_1 \quad (3.43)$$

Neste caso, resolvemos um sistema equivalente, pondo $\mathbf{P}dx = d\hat{x}$ e $\mathbf{P}r_1 = \hat{r}$, obtendo

$$\begin{bmatrix} \mathbf{D}_B^{-1} & \mathbf{B}^{-1}\mathbf{N}\mathbf{D}_N^{-1} \\ \mathbf{D}_N^{-1}\mathbf{N}^\top\mathbf{B}^{-\top} & -\mathbf{D}_N^{-1} \end{bmatrix} \begin{bmatrix} d\hat{x}_1 \\ d\hat{x}_2 \end{bmatrix} = \begin{bmatrix} \hat{r}_1 \\ \hat{r}_2 \end{bmatrix}, \quad (3.44)$$

onde os vetores $d\hat{x}$ e \hat{r} foram apropriadamente particionados. Observe que

$$\hat{r} = \mathbf{P}r_1 = \mathbf{P}\mathbf{P}^\top \begin{bmatrix} \mathbf{D}_B^{-1}\bar{r}_1 + \mathbf{B}^{-1}r_p \\ \mathbf{D}_N^{-1}\bar{r}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{D}_B^{-1}\bar{r}_1 + \mathbf{B}^{-1}r_p \\ \mathbf{D}_N^{-1}\bar{r}_2 \end{bmatrix} = \begin{bmatrix} \hat{r}_1 \\ \hat{r}_2 \end{bmatrix}.$$

Assim, fazemos

$$\begin{cases} \mathbf{D}_B^{-1}d\hat{x}_1 + \mathbf{B}^{-1}\mathbf{N}\mathbf{D}_N^{-1}d\hat{x}_2 = \mathbf{D}_B^{-1}\bar{r}_1 + \mathbf{B}^{-1}r_p \\ \mathbf{D}_N^{-1}\mathbf{N}^\top\mathbf{B}^{-\top}d\hat{x}_1 - \mathbf{D}_N^{-1}d\hat{x}_2 = \mathbf{D}_N^{-1}\bar{r}_2. \end{cases}$$

Obtemos $d\hat{x}_1$ fazendo

$$d\hat{x}_1 = \bar{r}_1 + \mathbf{D}_B\mathbf{B}^{-1}(r_p - \mathbf{N}\mathbf{D}_N^{-1}d\hat{x}_2), \quad (3.45)$$

que, substituindo na segunda igualdade do sistema, nos fornece

$$\begin{aligned} \mathbf{D}_N^{-1}\mathbf{N}^\top\mathbf{B}^{-\top} [\bar{r}_1 + \mathbf{D}_B\mathbf{B}^{-1}(r_p - \mathbf{N}\mathbf{D}_N^{-1}d\hat{x}_2)] - \mathbf{D}_N^{-1}d\hat{x}_2 &= \mathbf{D}_N^{-1}\bar{r}_2 \\ -\mathbf{D}_N^{-1}\mathbf{N}^\top\mathbf{B}^{-\top}\mathbf{D}_B\mathbf{B}^{-1}\mathbf{N}\mathbf{D}_N^{-1}d\hat{x}_2 - \mathbf{D}_N^{-1}d\hat{x}_2 + \mathbf{D}_N^{-1}\mathbf{N}^\top\mathbf{B}^{-\top} (\bar{r}_1 + \mathbf{D}_B\mathbf{B}^{-1}r_p) &= \mathbf{D}_N^{-1}\bar{r}_2, \end{aligned}$$

resultando em

$$(\mathbf{D}_N^{-1} + \mathbf{D}_N^{-1}\mathbf{N}^\top\mathbf{B}^{-\top}\mathbf{D}_B\mathbf{B}^{-1}\mathbf{N}\mathbf{D}_N^{-1}) d\hat{x}_2 = \mathbf{D}_N^{-1} [\mathbf{N}^\top\mathbf{B}^{-\top}(\bar{r}_1 + \mathbf{D}_B\mathbf{B}^{-1}r_p) - \bar{r}_2]. \quad (3.46)$$

Agora, podemos obter $d\hat{x}_2$ resolvendo o sistema (3.46) a partir do método de Gradientes Conjugados Pré-condicionado e, em seguida, substituir seu valor em (3.45) para encontramos $d\hat{x}_1$ e montarmos a solução de (3.44). Usando a relação $\mathbf{P}dx = d\hat{x}$ obtemos dx solução de (3.43), que juntamente com dy , obtido de (3.42), forma a solução para (3.41).

Para obtermos as soluções Δx e Δy realizamos os seguintes passos

$$\begin{aligned}
\mathbf{M}^\top \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} &= \begin{bmatrix} dx \\ dy \end{bmatrix} \Rightarrow \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \mathbf{M}^{-\top} \begin{bmatrix} dx \\ dy \end{bmatrix} \\
\begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} &= \begin{bmatrix} \mathbf{D}^{-1} & \mathbf{H}^\top \\ \mathbf{\Gamma}^\top & \mathbf{0} \end{bmatrix} \begin{bmatrix} dx \\ dy \end{bmatrix} = \begin{bmatrix} \mathbf{D}^{-1} \mathbf{P}^\top d\hat{x} + \mathbf{P}^\top \begin{bmatrix} \mathbf{I}_m \\ \mathbf{0} \end{bmatrix} dy \\ \mathbf{B}^{-\top} \begin{bmatrix} \mathbf{I}_m & \mathbf{0} \end{bmatrix} \mathbf{P} dx \end{bmatrix} \\
\begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} &= \begin{bmatrix} \mathbf{P}^\top \begin{bmatrix} \mathbf{D}_B^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{D}_N^{-1} \end{bmatrix} \begin{bmatrix} d\hat{x}_1 \\ d\hat{x}_1 \end{bmatrix} + \mathbf{P}^\top \begin{bmatrix} dy \\ \mathbf{0} \end{bmatrix} \\ \mathbf{B}^{-\top} d\hat{x}_1 \end{bmatrix} \\
\begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} &= \begin{bmatrix} \mathbf{P}^\top \begin{bmatrix} \mathbf{D}_B^{-1} d\hat{x}_1 + dy \\ \mathbf{D}_N^{-1} d\hat{x}_2 \\ \mathbf{B}^{-\top} d\hat{x}_1 \end{bmatrix} \end{bmatrix} \\
\begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} &= \begin{bmatrix} \mathbf{P}^\top \begin{bmatrix} \mathbf{D}_B^{-1} [\bar{r}_1 + \mathbf{D}_B \mathbf{B}^{-1} (r_p - \mathbf{N} \mathbf{D}_N^{-1} d\hat{x}_2)] - \mathbf{D}_B^{-1} \bar{r}_1 \\ \mathbf{D}_N^{-1} d\hat{x}_2 \\ \mathbf{B}^{-\top} [\bar{r}_1 + \mathbf{D}_B \mathbf{B}^{-1} (r_p - \mathbf{N} \mathbf{D}_N^{-1} d\hat{x}_2)] \end{bmatrix} \end{bmatrix} \\
\begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} &= \begin{bmatrix} \mathbf{P}^\top \begin{bmatrix} \mathbf{B}^{-1} (r_p - \mathbf{N} \mathbf{D}_N^{-1} d\hat{x}_2) \\ \mathbf{D}_N^{-1} d\hat{x}_2 \end{bmatrix} \\ \mathbf{B}^{-\top} [\bar{r}_1 + \mathbf{D}_B \mathbf{B}^{-1} (r_p - \mathbf{N} \mathbf{D}_N^{-1} d\hat{x}_2)] \end{bmatrix} \quad (3.47)
\end{aligned}$$

Novamente, a programação para esta segunda nova estratégia de solução fica restrita à solução do sistema em (3.46) e às expressões de Δx e Δy em (3.47).

3.4.4 Recuperação da solução: terceira estratégia nova

Uma vez que foi construído o pré-condicionador (3.26), podemos usar a última matriz em (3.25) para obter a matriz

$$\mathbf{D}_B + \mathbf{D}_B \mathbf{B}^{-1} \mathbf{N} \mathbf{D}_N^{-1} \mathbf{N}^\top \mathbf{B}^{-\top} \mathbf{D}_B. \quad (3.48)$$

Neste caso, o procedimento de recuperação de solução torna-se análogo à primeira estratégia, resolvendo

$$(\mathbf{D}_B + \mathbf{D}_B \mathbf{B}^{-1} \mathbf{N} \mathbf{D}_N^{-1} \mathbf{N}^\top \mathbf{B}^{-\top} \mathbf{D}_B) d\hat{x}_1 = \bar{r}_1 + \mathbf{D}_B \mathbf{B}^{-1} [r_p + \mathbf{N} \mathbf{D}_N^{-1} \bar{r}_2], \quad (3.49)$$

para encontrar

$$\begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} \mathbf{P}^\top \begin{bmatrix} d\hat{x}_1 - \mathbf{D}_B^{-1} \bar{r}_1 \\ \mathbf{D}_N^{-1} (\mathbf{N}^\top \mathbf{B}^{-\top} d\hat{x}_1 - \bar{r}_2) \\ \mathbf{B}^{-\top} \mathbf{D}_B d\hat{x}_1 \end{bmatrix} \end{bmatrix}. \quad (3.50)$$

3.4.5 Recuperação da solução: quarta estratégia nova

Assim como na terceira nova estratégia, adotamos o pré-condicionador (3.38) e usamos a última matriz de (3.37) para obter a matriz

$$\mathbf{D}_B^{-1} + \mathbf{B}^{-1}\mathbf{N}\mathbf{D}_N^{-1}\mathbf{N}^\top\mathbf{B}^{-\top}. \quad (3.51)$$

Assim, o procedimento de recuperação de solução torna-se análogo à segunda estratégia, resolvendo

$$(\mathbf{D}_B^{-1} + \mathbf{B}^{-1}\mathbf{N}\mathbf{D}_N^{-1}\mathbf{N}^\top\mathbf{B}^{-\top}) d\hat{x}_1 = \mathbf{D}_B^{-1}\bar{r}_1 + \mathbf{B}^{-1}[\mathbf{N}\mathbf{D}_N^{-1}\bar{r}_2 + r_p], \quad (3.52)$$

para encontrar

$$\begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} \mathbf{P}^\top \begin{bmatrix} \mathbf{D}_B^{-1}(d\hat{x}_1 - \bar{r}_1) \\ \mathbf{D}_N^{-1}(\mathbf{N}^\top\mathbf{B}^{-\top}d\hat{x}_1 - \bar{r}_2) \\ \mathbf{B}^{-\top}d\hat{x}_1 \end{bmatrix} \end{bmatrix}. \quad (3.53)$$

3.5 Considerações sobre a fatoração LU

Como esta classe de pré-condicionadores usa uma decomposição LU para encontrar um conjunto linearmente independente de colunas de \mathbf{A} que formam \mathbf{B} , sua implementação deve ser cuidadosa para que se torne competitiva. Sobre o cálculo da fatoração LU para o pré-condicionador, uma nova fatoração é realizada, de uma iteração do método de pontos interiores para outra, sempre que o método de Gradientes Conjugados Pré-condicionado necessitar de muitas iterações para convergir ou o resíduo do sistema linear for maior que a tolerância permitida. Além disso, uma segunda fatoração LU é calculada sempre que a primeira for muito densa. A seguir, descrevemos sucintamente as técnicas consideradas, técnicas estas que já se encontram presentes na versão utilizada (modificada) do PCx.

3.5.1 Obtendo colunas linearmente independentes

Uma vez obtido o ordenamento da matriz \mathbf{A} dado pela rotina apresentada na subseção 2.5.4 e juntamente com a rotina apresentada pela subseção 3.3.2 para obter colunas linearmente independentes, obtemos um conjunto de colunas de \mathbf{A} que formam \mathbf{B} .

A execução da fatoração LU é interrompida quando ocorre preenchimento excessivo. Neste caso, reordena-se o conjunto obtido de colunas linearmente independentes pelo número de entradas não nulas e reinicia-se a execução da fatoração. O processo se repete até que m colunas sejam obtidas. Isto evita o preenchimento excessivo na fatoração (GHIDINI; OLIVEIRA; SORENSEN, 2014).

Uma segunda fatoração LU (que usa técnicas sofisticadas para fatoração LU esparsa eficiente) é aplicada ao conjunto de m colunas obtidas da primeira fatoração. Com

isso, melhorias significativas são observadas, como mostra Ghidini, Oliveira e Sorensen (2014, p. 54).

3.5.2 Dependência e independência simbólica

A motivação é encontrar o único conjunto de m colunas linearmente independentes que preserva um ordenamento de colunas previamente fornecido. O intuito, então, é desenvolver estratégias que indiquem quando uma coluna pode ser ignorada na fatoração, economizando esforço computacional por estudar o ordenamento das colunas antes de efetuar a fatoração.

Colunas *simbolicamente dependentes* são colunas que são linearmente dependentes em estrutura para todos os valores numéricos de suas entradas não nulas. A ideia é encontrar um conjunto de colunas, digamos k colunas, com entradas não nulas em, no máximo, $k - 1$ linhas (GHIDINI; OLIVEIRA; SORENSEN, 2014).

Em matrizes quadradas, este procedimento é semelhante a permutar entradas não nulas na diagonal, que por sua vez é semelhante a encontrar uma correspondência de um grafo bipartido onde as linhas e colunas formam o conjunto de vértices e as arestas são representadas pelas entradas não nulas. Se uma entrada não nula não pode ser designada para a diagonal durante o processo de correspondência para uma dada coluna, tal coluna é simbolicamente dependente. Em matrizes retangulares o procedimento está associado a encontrar colunas independentes da matriz que fornece uma fatoração LU esparsa. As colunas são reordenadas pelo grau e aplica-se o algoritmo de correspondência para obter um conjunto de colunas-chave, que não são simbolicamente dependentes (GHIDINI; OLIVEIRA; SORENSEN, 2014).

Como o número de colunas linearmente independentes antes da k -ésima coluna-chave na matriz é no máximo $k - 1$, é possível acelerar a fatoração LU sempre que encontrarmos $k - 1$ colunas numericamente independentes localizadas antes da k -ésima coluna-chave. Então, podemos ignorar todas as colunas a partir da atual para a k -ésima coluna-chave e isto pode ser feito para todas as colunas-chave. Observe que não podemos simplesmente ignorar as colunas que não são colunas-chave porque as colunas-chave podem ser linearmente dependentes entre si (GHIDINI; OLIVEIRA; SORENSEN, 2014).

Colunas *simbolicamente independentes* são colunas que são linearmente independentes em estrutura para todos os valores numéricos de suas entradas não nulas. Uma poderosa estratégia consiste em mover colunas simbolicamente independentes para o início da lista ordenada, uma vez que estas colunas estão necessariamente indo para a fatoração. Então estas colunas podem ser posteriormente reordenadas para evitar preenchimento excessivo na fatoração LU. Observe que colunas simbolicamente dependentes podem ser ignoradas neste passo. Assim, podemos considerar apenas as colunas-chave fornecidas pelo

algoritmo de correspondência, que são candidatas a serem simbolicamente independentes (GHIDINI; OLIVEIRA; SORENSEN, 2014).

A heurística seguinte foi usada para encontrar colunas simbolicamente dependentes. Dizemos que a coluna j é a primeira entrada coluna de linha i se j contém a primeira entrada diferente de zero na linha i do conjunto ordenado.

Uma coluna j é simbolicamente independente, dado um conjunto ordenado, se ocorrer um dos seguintes casos:

1. Coluna j é a primeira entrada coluna de ao menos uma linha;
2. Coluna j é a segunda entrada coluna de uma linha i e a primeira entrada coluna de linha i é também primeira entrada coluna para ao menos outra linha não presente na coluna j .

Estas regras não garantem que todas as colunas simbolicamente dependentes sejam encontradas (GHIDINI; OLIVEIRA; SORENSEN, 2014).

4 Experimentos Computacionais e Análise dos Resultados

Baseado na versão do PCx proposta por Velazco, Oliveira e Campos (2010) (aqui denominada PCx_MOD) e orientado pelas estratégias de solução descritas nas subseções 3.4.2 a 3.4.5, foram desenvolvidas quatro versões para comparação: PCx_F3, PCx_F4, PCx_A3 e PCx_A4, respectivamente. Outras versões (PCx_F1, PCx_F2, PCx_A1 e PCx_A2) também foram desenvolvidas, porém foram descartadas da análise de resultados por apresentarem mais instabilidade numérica.

Para todas as versões, foram mantidos os parâmetros e critérios definidos na seção 3.3. O critério de troca de fases descrito na seção 3.2, bem como a heurística descrita na subseção 2.5.4, para obtenção de colunas de \mathbf{A} que formam \mathbf{B} , foram adotados. A estratégia de múltiplas correções de centralidade proposta por Gondzio (1996) foi desabilitada em todas as versões analisadas. O critério de parada para o método de Gradientes Conjugados Pré-condicionado foi baseado na norma Euclideana do resíduo do sistema linear: para as direções preditora e corretora na fase 1 a regra foi interromper a execução assim que o resíduo ficar menor que 10^{-4} . Assim que o *gap* de dualidade fica menor que 10^{-5} ou é detectada troca de fases, o critério muda para 10^{-8} . O número máximo de iterações para o método de Gradientes Conjugados Pré-condicionado é dado pela dimensão do sistema a ser resolvido. A linguagem de programação envolvida é C, com exceção das rotinas que realizam a Fatoração Controlada de Cholesky, que estão programadas em FORTRAN.

Os testes foram realizados em um computador equipado com processador Intel® Core™ i7 920 de 2,66GHz de frequência (modo turbo de 2,93GHz) e 8MB de cache, além de 8GB de memória RAM. O sistema operacional da máquina é Linux Kubuntu 15.10 e foi utilizado compilador GCC (GNU Compiler Collection disponível em <<https://gcc.gnu.org/>>). Neste trabalho, foram considerados 35 problemas selecionados de:

1. KENNINGTON (<<http://www.netlib.org/lp/data/kennington/>>);
2. MESZAROS (<http://www.sztaki.hu/~meszaros/public_ftp/lptestset/misc/>);
3. NETLIB (<<http://www.netlib.org/lp/data/>>);
4. QAPLIB (<<http://anjos.mgi.polymtl.ca/qaplib/>>);
5. RAIL (<<http://plato.asu.edu/ftp/lptestset/rail/>>).

Tabela 1 – Dados gerais dos problemas abordados.

Problema	Coleção	Linhas (m)	Colunas (n)	$n - m$	$nnz(\mathbf{A})$	$\bar{\eta}$	η_0
cre-b	KENNINGTON	5.328	36.382	31.054	112.233	10	-38
cre-d	KENNINGTON	4.094	28.601	24.507	86.704	10	-38
ken11	KENNINGTON	9.964	16.740	6.776	38.157	10	23
ken13	KENNINGTON	22.365	36.561	14.196	82.191	50	23
ken18	KENNINGTON	78.538	128.434	49.896	297.886	10	23
aa03	MESZAROS	690	8.572	7.882	60.898	-20	-82
air06	MESZAROS	690	8.572	7.882	60.898	-20	-82
aircraft	MESZAROS	3.754	7.517	3.763	20.267	-140	-398
bas1lp	MESZAROS	5.410	9.824	4.414	587.771	-150	-260
baxter_mat	MESZAROS	23.871	30.122	6.251	106.586	-50	-42
dbic1	MESZAROS	34.205	174.457	140.252	819.746	-20	-51
lpl1	MESZAROS	34.037	89.383	55.346	263.081	90	25
nemsemm1	MESZAROS	3.230	41.048	37.818	584.436	0	-73
nemsemm2	MESZAROS	4.526	26.754	22.228	91.127	100	28
nemspmm1	MESZAROS	2.227	7.145	4.918	43.606	420	-42
nemspmm2.in	MESZAROS	2.081	7.944	5.863	60.235	300	-43
nsir2	MESZAROS	4.450	10.054	5.604	154.935	-70	-73
nsir1	MESZAROS	4.406	10.010	5.604	143.247	-50	-69
nw14	MESZAROS	73	123.409	123.336	904.910	50	-46
pcb3000	MESZAROS	3.852	7.532	3.680	56.422	70	-31
seymour	MESZAROS	4.827	6.082	1.255	38.259	-100	-146
slp-tsk	MESZAROS	2.861	3.347	486	72.465	-1.140	-1.267
pilot	NETLIB	1.368	4.543	3.175	41.879	200	-63
pilots	NETLIB	1.368	4.543	3.175	41.879	200	-63
pilot87	NETLIB	1.971	6.373	4.402	72.163	90	-79
stocfor3	NETLIB	15.362	22.228	6.866	62.960	20	26
chr22b	QAPLIB	5.587	10.417	4.830	36.520	-40	-47
chr25a	QAPLIB	8.149	15.325	7.176	53.725	-30	-50
els19	QAPLIB	4.350	13.186	8.836	50.882	-30	-51
scr20	QAPLIB	5.079	15.980	10.901	61.780	-50	-52
scr15	QAPLIB	2.234	6.210	3.976	24.060	-10	-46
rou20	QAPLIB	7.359	37.640	30.281	152.980	0	-68
rail507	RAIL	507	62.689	62.182	404.727	0	-69
rail516	RAIL	516	47.827	47.311	315.412	0	-75
rail582	RAIL	582	54.906	54.324	394.113	0	-74

Fonte: Produzida pelo autor.

A [Tabela 1](#) apresenta os dados dos problemas abordados. A quantidade de linhas e colunas, assim como o número de elementos não nulos da matriz de coeficientes ($nnz(\mathbf{A})$), se referem à formulação resultante da etapa de pré-processamento (discutida na [subseção 3.3.2](#)). O parâmetro $\bar{\eta}$ (fornecido como dado de entrada) controla a troca de pré-condicionadores durante a execução do PCx: calculado η_0 e com base na heurística apresentada na [seção 3.2](#), atualiza-se o valor de η até o limite $\bar{\eta}$; se uma nova atualização for necessária e já houver atingido o valor $\bar{\eta}$, então ocorre a mudança de pré-condicionadores. Testes prévios foram realizados com valores pré-estabelecidos para $\bar{\eta}$ e foi escolhido aquele que promovesse a troca de pré-condicionadores.

4.1 Apresentação dos resultados

Durante os primeiros testes computacionais com as versões produzidas, verificou-se que cada problema considerado possui um domínio de possíveis valores para o parâmetro η na FCC. Além disso, existem problemas que não trocam de fases, independentemente do valor fornecido para o parâmetro $\bar{\eta}$, fazendo com que o problema seja descartado da análise de resultados (uma vez que toda a discussão deste trabalho faz parte da fase 2). Estes têm sido obstáculos na solução dos problemas, pois adotar um único valor de $\bar{\eta}$ para todos os problemas pode fazer com que não haja troca de pré-condicionadores, uma vez que o parâmetro escolhido pode estar fora do domínio.

O pré-condicionador da FCC é obtido em função da memória disponível no computador. Sendo assim, variações no consumo de memória durante a execução da fase 1 fazem com que resultados distintos sejam observados para um mesmo problema. Os resultados também podem variar com a escolha feita para obter \mathbf{B} de \mathbf{A} , visto que a matriz \mathbf{B} se altera em determinadas condições durante a execução da fase 2. Face a estas dificuldades, foram considerados apenas os problemas que trocam de fases durante a execução do programa e foram considerados diferentes valores do parâmetro $\bar{\eta}$ para cada problema avaliado. Além disso, considerou-se realizar cinco execuções das versões do PCx para a lista de problemas escolhida, afim de tomar a média dos valores observados.

A [Tabela 2](#) e a [Tabela 3](#) apresentam resultados obtidos a partir da execução pelas versões PCx_MOD, PCx_F3, PCx_F4, PCx_A3 e PCx_A4 da lista de problemas selecionados. As tabelas apresentam a quantidade de iterações do método de pontos interiores (It), o tempo total gasto para resolver o problema (para todas as tabelas deste capítulo, o tempo é medido em segundos) e o *status* exibido na tela (Stt) após finalizado o processo de solução do problema, conforme descrito na [subseção 3.3.5](#) (na tabela, O indica que o programa terminou com *status* “ótimo”, U indica *status* “desconhecido”, I significa “inactivo” e S significa “subótimo”). As tabelas apresentam, ainda, a iteração onde houve a troca de pré-condicionadores (Trc).

Outra informação que também foi considerada para comparação neste trabalho é o número de iterações do método Gradientes Conjugados pré-condicionado para as soluções dos sistemas lineares das direções preditor e corretor. Para isso, o critério utilizado foi comparar o total de iterações de gradientes conjugados (direções preditor e corretor) para todas as iterações em que o pré-condicionador separador esteve em uso, para um dado problema. A [Tabela 4](#) e a [Tabela 5](#) mostram os resultados obtidos para as cinco versões. Nelas, são exibidos o *status* final do problema em questão, a quantidade de iterações necessária para o método de pontos interiores atingir tal *status* (MPI) e a quantidade de iterações do método de Gradientes Conjugados pré-condicionado (PGC).

Por fim, a quantidade de vezes que uma nova matriz \mathbf{B} foi calculada, após a

troca de pré-condicionadores, bem como a iteração em que tal mudança ocorre, também é uma informação usada para comparação entre as versões testadas neste trabalho. O cálculo de uma nova matriz \mathbf{B} pode ser uma etapa consideravelmente cara, do ponto de vista computacional, uma vez que é feita uma nova fatoração LU. Sendo assim, se uma quantidade menor de cálculos para obtenção de uma nova matriz \mathbf{B} é realizada, o custo computacional pode ser reduzido, refletindo em uma possível queda no tempo computacional gasto para resolver o problema de programação linear. A [Tabela 6](#) e a [Tabela 7](#) apresentam esta informação exibindo a quantidade de vezes em que tal matriz foi calculada (nB), a iteração onde houve a troca de pré-condicionadores (Trc) e a(s) iteração(ões) onde ocorreu o cálculo de uma nova matriz \mathbf{B} .

Tabela 2 – Resultados gerais para as versões PCx_MOD, PCx_F3 e PCx_F4.

Problema	PCx_MOD				PCx_F3				PCx_F4			
	It	Tempo	Stt	Trc	It	Tempo	Stt	Trc	It	Tempo	Stt	Trc
cre-b	43	68,02	O	39	43	79,21	O	39	43	67,85	O	39
cre-d	42	52,51	O	39	42	55,33	O	39	45	56,24	O	39
ken11	22	9,76	O	20	22	10,06	O	20	23	11,01	O	20
ken13	30	529,54	O	28	30	484,68	O	28	31	589,31	U	28
ken18	39	786,51	O	29	38	551,91	O	29	40	608,87	O	29
aa03	23	10,05	U	8	23	9,50	O	8	52	11,67	U	8
air06	23	10,05	U	8	23	9,48	O	8	52	11,66	U	8
aircraft	40	76,04	U	34	48	68,24	O	34	40	65,38	U	34
basllp	17	521,62	O	12	16	444,53	O	12	16	444,66	O	12
baxter_mat	34	326,30	U	15	34	135,11	O	15	62	321,05	U	15
dbic1	56	1.839,85	O	55	57	1.937,80	O	55	56	2.014,65	O	55
lpl1	73	6.212,84	O	71	73	5.804,32	O	71	73	5.810,40	O	71
nemsemm1	77	99,46	O	75	77	101,90	O	75	77	101,96	O	75
nemsemm2	43	97,61	O	42	43	87,67	O	42	43	86,46	O	42
nemspmm1	62	357,18	O	55	61	449,09	O	55	61	419,56	O	55
nemspmm2.in	46	169,19	O	43	46	151,37	O	43	46	149,57	O	43
nsir2	33	93,27	O	2	32	127,78	O	2	33	89,53	U	2
nsir1	30	70,59	O	3	26	57,21	O	3	28	56,19	O	3
nw14	48	14,73	O	40	48	17,97	O	40	49	17,00	O	40
pcb3000	29	75,15	O	27	29	65,79	O	27	29	65,74	O	27
seymour	20	34,12	O	15	20	29,23	O	15	20	33,01	O	15
slp-tsk	26	101,11	O	22	25	93,79	O	22	25	92,69	O	22
pilot	41	41,45	O	37	63	39,48	O	37	49	37,42	O	37
pilots	41	41,69	O	37	63	39,47	O	37	49	42,32	O	37
pilot87	37	50,87	O	36	37	44,94	O	36	37	44,68	O	36
stocfor3	32	137,58	O	20	33	132,50	O	20	32	121,84	O	20
chr22b	30	11,92	O	10	30	16,27	O	10	60	13,38	U	10
chr25a	28	28,48	O	11	27	31,58	O	11	28	30,09	O	11
els19	31	73,53	O	11	50	113,51	O	11	32	50,84	O	11
scr20	21	91,77	O	9	22	87,99	O	9	57	214,96	O	9
scr15	24	9,04	O	10	25	12,53	O	10	25	10,10	O	10
rou20	24	1.555,74	O	12	26	1.953,01	O	12	53	2.686,72	U	12
rail507	35	26,23	I	18	38	40,08	O	18	55	62,71	O	18
rail516	29	15,12	O	23	30	16,63	O	23	30	18,42	O	23
rail582	39	30,52	O	22	39	31,30	O	22	39	35,07	O	22

Fonte: Produzida pelo autor.

Tabela 3 – Resultados gerais para as versões PCx_MOD, PCx_A3 e PCx_A4.

Problema	PCx_MOD				PCx_A3				PCx_A4			
	It	Tempo	Stt	Trc	It	Tempo	Stt	Trc	It	Tempo	Stt	Trc
cre-b	43	68,02	O	39	43	69,37	O	39	43	69,23	O	39
cre-d	42	52,51	O	39	42	48,60	O	39	42	48,40	O	39
ken11	22	9,76	O	20	22	8,82	O	20	22	8,93	O	20
ken13	30	529,54	O	28	30	480,38	O	28	30	492,77	O	28
ken18	39	786,51	O	29	38	550,93	O	29	40	556,70	O	29
aa03	23	10,05	U	8	23	9,08	O	8	22	7,21	O	8
air06	23	10,05	U	8	23	9,08	O	8	22	7,20	O	8
aircraft	40	76,04	U	34	101	86,44	S	34	43	66,54	O	34
bas1lp	17	521,62	O	12	18	445,49	O	12	22	539,38	I	12
baxter_mat	34	326,30	U	15	34	140,75	O	15	36	216,35	O	15
dbic1	56	1.839,85	O	55	86	1.987,64	U	55	56	1.894,35	O	55
lpl1	73	6.212,84	O	71	73	5.799,26	O	71	73	5.522,71	O	71
nemsemm1	77	99,46	O	75	77	100,97	O	75	77	100,65	O	75
nemsemm2	43	97,61	O	42	43	87,48	O	42	43	87,85	O	42
nemspmm1	62	357,18	O	55	61	389,87	O	55	61	361,09	O	55
nemspmm2.in	46	169,19	O	43	46	149,67	O	43	46	152,04	O	43
nsir2	33	93,27	O	2	34	116,74	O	2	33	84,30	O	2
nsir1	30	70,59	O	3	26	56,74	O	3	30	53,00	O	3
nw14	48	14,73	O	40	48	16,89	O	40	48	15,34	O	40
pcb3000	29	75,15	O	27	29	65,88	O	27	29	65,53	O	27
seymour	20	34,12	O	15	55	34,55	U	15	21	29,34	O	15
slp-tsk	26	101,11	O	22	25	89,90	O	22	25	91,12	O	22
pilot	41	41,45	O	37	41	35,75	U	37	41	36,41	O	37
pilots	41	41,69	O	37	41	35,81	U	37	41	36,52	O	37
pilot87	37	50,87	O	36	41	45,23	U	36	37	44,99	O	36
stocfor3	32	137,58	O	20	32	122,57	O	20	32	123,67	O	20
chr22b	30	11,92	O	10	29	14,02	O	10	29	11,29	O	10
chr25a	28	28,48	O	11	59	62,35	U	11	59	60,31	U	11
els19	31	73,53	O	11	33	65,60	O	11	39	63,22	O	11
scr20	21	91,77	O	9	23	91,27	O	9	21	61,28	O	9
scr15	24	9,04	O	10	25	10,15	O	10	25	9,22	O	10
rou20	24	1.555,74	O	12	25	1.615,53	O	12	24	1.056,84	O	12
rail507	35	26,23	I	18	33	21,80	O	18	32	21,34	O	18
rail516	29	15,12	O	23	31	16,01	O	23	29	14,82	O	23
rail582	39	30,52	O	22	39	28,04	O	22	39	26,92	O	22

Fonte: Produzida pelo autor.

Tabela 4 – Total de iterações do método de Gradientes Conjugados pré-condicionado para as versões PCx_MOD, PCx_F3 e PCx_F4.

Problema	PCx_MOD			PCx_F3			PCx_F4		
	Stt	MPI	PCG	Stt	MPI	PCG	Stt	MPI	PCG
cre-b	O	43	948	O	43	727	O	43	692
cre-d	O	42	583	O	42	587	O	45	1.072
ken11	O	22	38	O	22	79	O	23	100
ken13	O	30	65	O	30	94	U	31	127
ken18	O	39	4.233	O	38	1.026	O	40	1.184
aa03	U	23	2.272	O	23	1.425	U	52	2.669
air06	U	23	2.272	O	23	1.425	U	52	2.669
aircraft	U	40	176	O	48	2.500	U	40	86
bas1lp	O	17	3.018	O	16	509	O	16	570
baxter_mat	U	34	46.648	O	34	33.459	U	62	95.600
dbic1	O	56	826	O	57	145	O	56	763
lpl1	O	73	1.742	O	73	1.418	O	73	1.966
nemsemm1	O	77	57	O	77	74	O	77	90
nemsemm2	O	43	54	O	43	96	O	43	125
nemspmm1	O	62	991	O	61	549	O	61	561
nemspmm2.in	O	46	1.123	O	46	1.094	O	46	1.282
nsir2	O	33	31.873	O	32	21.943	U	33	22.975
nsir1	O	30	15.346	O	26	9.980	O	28	10.672
nw14	O	48	164	O	48	125	O	49	162
pcb3000	O	29	548	O	29	215	O	29	171
seymour	O	20	1.259	O	20	470	O	20	470
slp-tsk	O	26	168	O	25	22	O	25	20
pilot	O	41	786	O	63	1.815	O	49	1.008
pilots	O	41	786	O	63	1.815	O	49	1.008
pilot87	O	37	110	O	37	100	O	37	75
stocfor3	O	32	15.108	O	33	13.440	O	32	13.797
chr22b	O	30	9.275	O	30	9.571	U	60	8.907
chr25a	O	28	11.418	O	27	9.331	O	28	11.035
els19	O	31	11.997	O	50	22.023	O	32	8.996
scr20	O	21	12.389	O	22	8.272	O	57	30.759
scr15	O	24	4.557	O	25	5.371	O	25	5.273
rou20	O	24	8.354	O	26	9.102	U	53	26.027
rail507	I	35	2.409	O	38	1.563	O	55	2.164
rail516	O	29	647	O	30	708	O	30	742
rail582	O	39	2.201	O	39	1.501	O	39	1.611

Fonte: Produzida pelo autor.

Tabela 5 – Total de iterações do método de Gradientes Conjugados pré-condicionado para as versões PCx_MOD, PCx_A3 e PCx_A4.

Problema	PCx_MOD			PCx_A3			PCx_A4		
	Stt	MPI	PCG	Stt	MPI	PCG	Stt	MPI	PCG
cre-b	O	43	948	O	43	1.123	O	43	532
cre-d	O	42	583	O	42	877	O	42	409
ken11	O	22	38	O	22	99	O	22	32
ken13	O	30	65	O	30	109	O	30	14
ken18	O	39	4.233	O	38	879	O	40	465
aa03	U	23	2.272	O	23	1.766	O	22	1.120
air06	U	23	2.272	O	23	1.766	O	22	1.120
aircraft	U	40	176	S	101	18.424	O	43	283
bas1lp	O	17	3.018	O	18	1.105	I	22	11.478
baxter_mat	U	34	46.648	O	34	28.171	O	36	49.192
dbic1	O	56	826	U	86	14.747	O	56	866
lpl1	O	73	1.742	O	73	1.517	O	73	1.457
nemsemm1	O	77	57	O	77	144	O	77	45
nemsemm2	O	43	54	O	43	155	O	43	67
nemspmm1	O	62	991	O	61	587	O	61	367
nemspmm2.in	O	46	1.123	O	46	938	O	46	1.074
nsir2	O	33	31.873	O	34	29.223	O	33	26.619
nsir1	O	30	15.346	O	26	13.622	O	30	13.051
nw14	O	48	164	O	48	155	O	48	127
pcb3000	O	29	548	O	29	241	O	29	192
seymour	O	20	1.259	U	55	6.082	O	21	678
slp-tsk	O	26	168	O	25	19	O	25	14
pilot	O	41	786	U	41	182	O	41	239
pilots	O	41	786	U	41	182	O	41	239
pilot87	O	37	110	U	41	350	O	37	77
stocfor3	O	32	15.108	O	32	15.579	O	32	14.412
chr22b	O	30	9.275	O	29	7.897	O	29	7.137
chr25a	O	28	11.418	U	59	26.390	U	59	32.297
els19	O	31	11.997	O	33	12.044	O	39	14.142
scr20	O	21	12.389	O	23	12.760	O	21	8.187
scr15	O	24	4.557	O	25	4.904	O	25	4.701
rou20	O	24	8.354	O	25	7.842	O	24	5.916
rail507	I	35	2.409	O	33	1.368	O	32	1.343
rail516	O	29	647	O	31	797	O	29	675
rail582	O	39	2.201	O	39	1.781	O	39	1.587

Fonte: Produzida pelo autor.

Tabela 6 – Dados de atualização da matriz **B** após a mudança de pré-condicionadores para as versões PCx_MOD, PCx_F3 e PCx_F4.

Problema	PCx_MOD			PCx_F3			PCx_F4		
	nB	Trc	IterNewB	nB	Trc	IterNewB	nB	Trc	IterNewB
cre-b	1	39	39	1	39	39	1	39	39
cre-d	1	39	39	1	39	39	1	39	39
ken11	1	20	20	1	20	20	1	20	20
ken13	1	28	28	1	28	28	1	28	28
ken18	1	29	29	1	29	29	1	29	29
aa03	4	8	8; 13; 18; 22	3	8	8; 16; 22	2	8	8; 18
air06	4	8	8; 13; 18; 22	3	8	8; 16; 22	2	8	8; 18
aircraft	1	34	34	2	34	34; 46	1	34	34
basllp	2	12	12; 16	1	12	12	1	12	12
baxter_mat	3	15	15; 23; 32	3	15	15; 27; 32	2	15	15; 27
dbic1	1	55	55	1	55	55	1	55	55
lpl1	1	71	71	1	71	71	1	71	71
nemsemm1	1	75	75	1	75	75	1	75	75
nemsemm2	1	42	42	1	42	42	1	42	42
nemspmm1	1	55	55	1	55	55	1	55	55
nemspmm2.in	2	43	43; 45	1	43	43	3	43	43; 44; 45
nsir2	9	2	2; 3; 4; 7; 16; 22; 24; 26; 30	10	2	2; 21; 23; 24; 25; 26; 27; 28; 29; 31	7	2	2; 21; 23; 24; 28; 31; 32
nsir1	4	3	3; 5; 20; 25	3	3	3; 21; 25	3	3	3; 22; 26
nw14	5	40	40; 41; 43; 45; 47	3	40	40; 44; 47	4	40	40; 43; 47; 48
pcb3000	1	27	27	1	27	27	1	27	27
seymour	2	15	15; 19	1	15	15	1	15	15
slp-tsk	1	22	22	1	22	22	1	22	22
pilot	1	37	37	1	37	37	1	37	37
pilots	1	37	37	1	37	37	1	37	37
pilot87	1	36	36	1	36	36	1	36	36
stocfor3	2	20	20; 27	2	20	20; 28	2	20	20; 28
chr22b	3	10	10; 19; 26	2	10	10; 25	2	10	10; 26
chr25a	3	11	11; 18; 23	2	11	11; 22	2	11	11; 23
els19	5	11	11; 15; 20; 24; 28	4	11	11; 18; 25; 49	3	11	11; 18; 26
scr20	4	9	9; 11; 15; 18	3	9	9; 15; 19	6	9	9; 15; 32; 39; 46; 51
scr15	4	10	10; 14; 19; 22	3	10	10; 18; 24	3	10	10; 18; 24
rou20	3	12	12; 16; 22	3	12	12; 16; 25	2	12	12; 17
rail507	8	18	18; 20; 22; 24; 27; 31; 33; 34	3	18	18; 31; 37	3	18	18; 46; 53
rail516	3	23	23; 26; 28	2	23	23; 29	2	23	23; 29
rail582	6	22	22; 25; 29; 33; 36; 38	3	22	22; 28; 35	3	22	22; 30; 36

Fonte: Produzida pelo autor.

Tabela 7 – Dados de atualização da matriz **B** após a mudança de pré-condicionadores para as versões PCx_MOD, PCx_A3 e PCx_A4.

Problema	PCx_MOD			PCx_A3			PCx_A4		
	nB	Trc	IterNewB	nB	Trc	IterNewB	nB	Trc	IterNewB
cre-b	1	39	39	1	39	39	1	39	39
cre-d	1	39	39	1	39	39	1	39	39
ken11	1	20	20	1	20	20	1	20	20
ken13	1	28	28	1	28	28	1	28	28
ken18	1	29	29	1	29	29	1	29	29
aa03	4	8	8; 13; 18; 22	3	8	8; 15; 22	2	8	8; 15
air06	4	8	8; 13; 18; 22	3	8	8; 15; 22	2	8	8; 15
aircraft	1	34	34	1	34	34	1	34	34
bas1lp	2	12	12; 16	1	12	12	1	12	12
baxter_mat	3	15	15; 23; 32	3	15	15; 26; 33	3	15	15; 28; 35
dbic1	1	55	55	1	55	55	1	55	55
lpl1	1	71	71	1	71	71	1	71	71
nemsemm1	1	75	75	1	75	75	1	75	75
nemsemm2	1	42	42	1	42	42	1	42	42
nemspmm1	1	55	55	1	55	55	1	55	55
nemspmm2.in	2	43	43; 45	1	43	43	1	43	43; 44; 45
nsir2	9	2	2; 3; 4; 7; 16; 22; 24; 26; 30	8	2	2; 3; 20; 22; 24; 30; 32; 33	9	2	2; 3; 4; 19; 22; 23; 24; 28; 32
nsir1	4	3	3; 5; 20; 25	3	3	3; 19; 23	3	3	3; 20; 25
nw14	5	40	40; 41; 43; 45; 47	5	40	40; 42; 45; 46; 47	4	40	40; 41; 43; 47
pcb3000	1	27	27	1	27	27	1	27	27
seymour	2	15	15; 19	1	15	15	1	15	15
slp-tsk	1	22	22	1	22	22	1	22	22
pilot	1	37	37	1	37	37	1	37	37
pilots	1	37	37	1	37	37	1	37	37
pilot87	1	36	36	1	36	36	1	36	36
stocfor3	2	20	20; 27	2	20	20; 28	2	20	20; 28
chr22b	3	10	10; 19; 26	2	10	10; 25	2	10	10; 25
chr25a	3	11	11; 18; 23	2	11	11; 23	2	11	11; 22
els19	5	11	11; 15; 20; 24; 28	4	11	11; 18; 25; 32	4	11	11; 18; 25; 38
scr20	4	9	9; 11; 15; 18	4	9	9; 13; 17; 22	3	9	9; 14; 18
scr15	4	10	10; 14; 19; 22	3	10	10; 18; 24	3	10	10; 18; 24
rou20	3	12	12; 16; 22	3	12	12; 16; 23	2	12	12; 17
rail507	8	18	18; 20; 22; 24; 27; 31; 33; 34	3	18	18; 23; 30	3	18	18; 22; 29
rail516	3	23	23; 26; 28	3	23	23; 29; 30	1	23	23
rail582	6	22	22; 25; 29; 33; 36; 38	3	22	22; 28; 35	3	22	22; 27; 34

Fonte: Produzida pelo autor.

4.2 Análise dos dados

Os resultados gerais apresentados nas tabelas da seção 4.1 foram resumidos em perfis de desempenho, tais como descrito em Dolan e Moré (2002), utilizando escala logarítmica para visualizar toda a atividade de desempenho das versões a partir dos perfis de desempenho. O arquivo gerador dos perfis de desempenho é o mesmo `perf.m` descrito no trabalho e que está disponível em <http://www.mcs.anl.gov/~more/cops/>.

A Tabela 2 e a Tabela 3 mostram que, para cada problema, a troca de pré-condicionadores ocorre sempre na mesma iteração para todas as versões avaliadas, uma vez que nada foi alterado no procedimento que realiza a troca de pré-condicionadores, tampouco foram feitas alterações nas rotinas que envolvem o pré-condicionador da Fatoração Controlada de Cholesky. Como as alterações se concentraram nas rotinas desenvolvidas para a fase 2, as diferenças no total de iterações do método de pontos interiores são devidas aos métodos de solução adotados (e às características apresentadas) pelo pré-condicionador desenvolvido para tal fase.

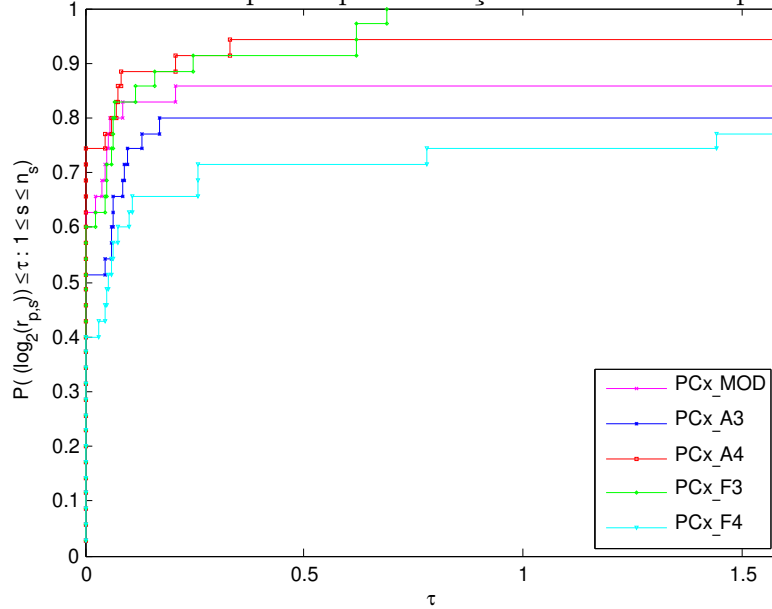
Os sistemas lineares a serem resolvidos pelas versões PCx_MOD, PCx_F3 e PCx_F4 envolvem as matrizes (2.11a), (3.24) e (3.36), respectivamente (dimensão m para a versão PCx_MOD e $n - m$ para as versões PCx_F3 e PCx_F4). Pontualmente, podemos ver que a versão PCx_F3 consegue reduzir em 1 o total de iterações do método de pontos interiores para 6 problemas, a saber `ken18`, `bas11p`, `nemspmm1`, `nsir2`, `slp-tsk` e `chr25a`, quando comparado com a versão PCx_MOD. Para os problemas `nemspmm1` e `nsir2`, o sistema linear resolvido pela versão PCx_F3 possui dimensões 4.918 e 5.604, respectivamente, enquanto que a versão PCx_MOD resolve um sistema linear menor, de dimensões 2.227 e 4.450, respectivamente. Para o problema `nsir1` a redução foi de 4 iterações e o sistema resolvido pela PCx_F3 é de dimensão 5.604, enquanto que a PCx_MOD resolve o mesmo problema usando um sistema linear menor, de dimensão 4.406. Dos 9 problemas em que a versão PCx_F3 teve desvantagem em relação à versão PCx_MOD, apenas no problema `stocfor3` que se observa um sistema linear menor a ser resolvido: dimensão 6.866, sendo que PCx_MOD resolve um sistema de dimensão 15.362. Comparando PCx_F4 com PCx_MOD, apenas em 4 problemas (`bas11p`, `nemspmm1`, `nsir1` e `slp-tsk`) houve vantagem para PCx_F4. Dos 10 problemas em que PCx_F4 obteve desvantagem no número de iterações do método de pontos interiores, em 2 problemas (`ken11` e `ken18`) o sistema linear resolvido foi menor que o da versão PCx_MOD: 6.776 e 49.896, contra 9.964 e 78.538, respectivamente.

As versões PCx_MOD, PCx_A3 e PCx_A4 resolvem os problemas utilizando um sistema linear de dimensão m , porém usando as matrizes (2.11a), (3.48) e (3.51), respectivamente. Em 4 problemas (`ken18`, `nemspmm1`, `slp-tsk` e `chr22b`) a versão PCx_A3 reduziu em 1 o total de iterações do método de pontos interiores, comparado com PCx_MOD, reduzindo em 4 iterações para o problema `nsir1`. Tal versão obteve

desvantagem em 7 problemas. Já a versão PCx_A4 obteve vantagem em 3 problemas, reduzindo em 1 o total de iterações do método de pontos interiores, enquanto obteve desvantagem em 4 problemas.

A Figura 5 apresenta o perfil de desempenho considerando a quantidade de iterações do método de pontos interiores necessária para que os problemas sejam resolvidos. Podemos perceber que, em relação à lista de problemas selecionados para teste, a versão PCx_F3 conseguiu obter sucesso na solução de todos os 35 problemas, enquanto que as demais versões não atingiram o topo do perfil por não resolver todos os problemas da lista. A figura mostra, ainda, que a eficiência apresentada pelas versões PCx_F4 e PCx_A3 é inferior à das demais versões avaliadas, se considerarmos sucesso em mais do que 50% dos problemas testados.

Figura 5 – Perfil de desempenho para iterações do método de pontos interiores

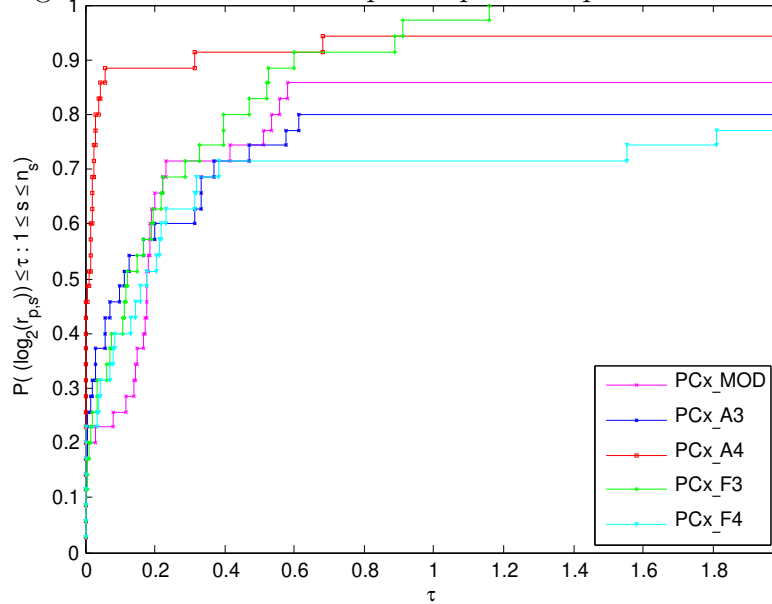


Fonte: Produzido pelo autor.

Em relação ao tempo computacional necessário para a solução dos problemas listados, a versão PCx_F3 apresenta vantagem em 15 problemas e desvantagem em outros 15. Dos 15 problemas onde foi verificada vantagem para o PCx_F3 em relação a PCx_MOD, para 8 deles o sistema linear a ser resolvido tem dimensão maior. Já dos 15 problemas onde PCx_F3 obteve desvantagem, somente 3 apresentavam um sistema linear de dimensão menor que o resolvido pela versão PCx_MOD. Quando comparamos PCx_F4 com PCx_MOD, em 14 problemas houve vantagem para PCx_F4, sendo que para 8 deles o sistema linear resolvido tem dimensão maior que para PCx_MOD. Dos 12 problemas onde houve desvantagem para PCx_F4, somente 2 forneciam um sistema linear de dimensão reduzida em relação a PCx_MOD. Para a versão PCx_A3, em relação a PCx_MOD, foi verificada vantagem em 15 problemas. Em 9 problemas a versão PCx_A3 teve desvantagem em relação a PCx_MOD. O perfil de desempenho para esta métrica

pode ser visto na [Figura 6](#). Ela mostra que a versão PCx_A4 é a mais eficiente dentre as apresentadas quando queremos resolver até 90% dos problemas. Para taxas mais apertadas, a versão PCx_F3 aparece como alternativa, sendo o mais robusto quando estendemos o tempo necessário até um fator de 2,5 da melhor versão, conseguindo resolver todos os problemas com esse tempo. As demais versões apresentam desempenho semelhante, começando a se distinguirem quando consideramos o tempo necessário a um fator de aproximadamente 1,1 da melhor versão.

Figura 6 – Perfil de desempenho para tempo total de solução



Fonte: Produzido pelo autor.

Podemos notar pela [Tabela 2](#) que para os problemas **nemspmm1**, **nsir2** e **chr25a** houve uma redução no número de iterações do método de pontos interiores para a versão PCx_F3 em relação a PCx_MOD e ao mesmo tempo um aumento no tempo computacional gasto, enquanto que a situação inversa (aumento na quantidade de iterações do método de pontos interiores e concomitante redução no tempo computacional) foi verificada para os problemas **pilot**, **pilots**, **stocfor3** e **scr20**. Para os problemas **pilot** e **pilots** o aumento no número de iterações do método de pontos interiores foi bastante significativo, sendo de 41 para PCx_MOD e 63 para PCx_F3. Somente o problema **nemspmm1** apresentou redução de iterações do método de pontos interiores com aumento do tempo computacional para PCx_F4 em relação a PCx_MOD, enquanto que nos problemas **ken18**, **pilot** e **els19** foi possível observar a situação inversa. Ainda para PCx_F4, em **cre-d** e **pilots** houve aumento no número de iterações do método de pontos interiores, mas o tempo computacional foi bastante próximo ao apresentado por PCx_MOD.

A partir da [Tabela 3](#) vemos um aumento no número de iterações do método de pontos interiores e diminuição no tempo computacional para PCx_A3 em relação a PCx_MOD nos problemas **bas1lp** e **els19**, com situação inversa para os problemas

nemspmm1 e **chr22b**. Somente para o problema **els19** pode ser observado um aumento no número de iterações concomitante à redução no tempo computacional para PCx_A4 em relação a PCx_MOD, sendo que o inverso foi observado no problema **nemspmm1**. O problema **chr22b** apresentou redução de iterações, mas o tempo computacional permaneceu próximo ao observado em PCx_MOD, quando comparado a PCx_A4.

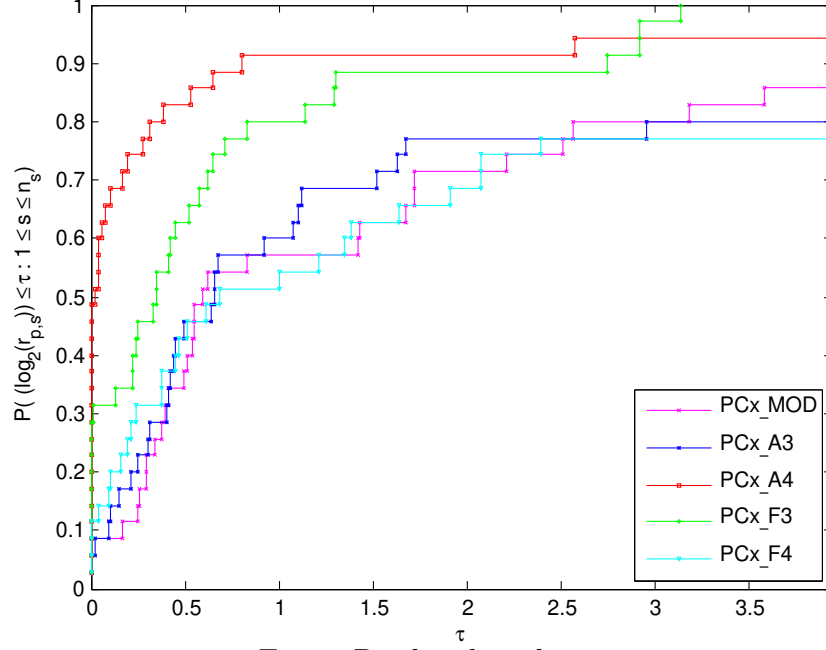
A [Tabela 4](#) mostra que a versão PCx_F3 resolveu 18 problemas com uma quantidade menor de iterações do método de Gradientes Conjugados pré-condicionado em relação a PCx_MOD, obtendo desvantagem nesta métrica para 11 problemas. Já a versão PCx_F4 resolveu 15 problemas com menos iterações do método de Gradientes Conjugados pré-condicionado do que PCx_MOD, com desvantagem em 9 problemas. Nota-se, para PCx_F3, que o sistema linear resolvido era maior para 11 dos 18 problemas com melhor resultado para esta métrica, sendo que o sistema foi menor para 3 dos 11 problemas com resultados desfavoráveis. Já para PCx_F4, o sistema maior aparecem em 8 dos 15 problemas de melhor resultado para esta métrica, sendo que o sistema menor aparece em apenas 1 dos 9 problemas desvantajosos em relação a esta métrica. Em relação a PCx_MOD, a versão PCx_F3 apresenta um aumento no número de iterações do método de pontos interiores concomitante a uma redução no número de iterações do método de Gradientes Conjugados pré-condicionado para os problemas **dbic1** e **scr20**, sendo que ambos apresentam um sistema linear de maior dimensão a ser resolvido. Ainda em relação a PCx_MOD mas considerando, agora, PCx_F4 podemos verificar o mesmo tipo de situação ocorrendo para os problemas **nw14** e **els19**, sendo que a diferença na dimensão do sistema linear a ser resolvido é bastante significativa para o problema **nw14**, pois na versão PCx_MOD a dimensão é 73, enquanto que na versão PCx_F4 a dimensão salta para 123.336.

Baseado na [Tabela 5](#) vemos que a versão PCx_A3 resolveu 13 problemas com menos iterações do método de Gradientes Conjugados pré-condicionado em relação a PCx_MOD, com desvantagem em 11 problemas. Já PCx_A4 resolveu 23 problemas com menos iterações e foi observada desvantagem em 5 problemas, comparando com PCx_MOD. Os problemas **ken18** e **seymour** apresentaram aumento no número de iterações do método de pontos interiores juntamente com uma redução nas iterações do método de Gradientes Conjugados pré-condicionado para a versão PCx_A3 em relação à versão PCx_MOD. Para a versão PCx_A4 em relação a PCx_MOD, a mesma situação foi verificada nos problemas **bas1lp** e **nsir2**.

O perfil de desempenho que considera a quantidade de iterações para o método de Gradientes Conjugados pré-condicionado é exibido na [Figura 7](#). Considerando esta métrica de desempenho e olhando para a figura, podemos ver que a eficiência das versões PCx_F3 e PCx_A4 é superior àquela apresentada pelas demais versões. PCx_A4 recebe destaque, resolvendo 50% dos problemas com o menor número de iterações, ante apro-

ximadamente 31% para PCx_F3. As demais versões apresentam desempenho bastante próximo, mesmo considerando um fator τ grande da melhor versão, nesta métrica.

Figura 7 – Perfil de desempenho para iterações de Gradientes Conjugados pré-condicionado



Fonte: Produzido pelo autor.

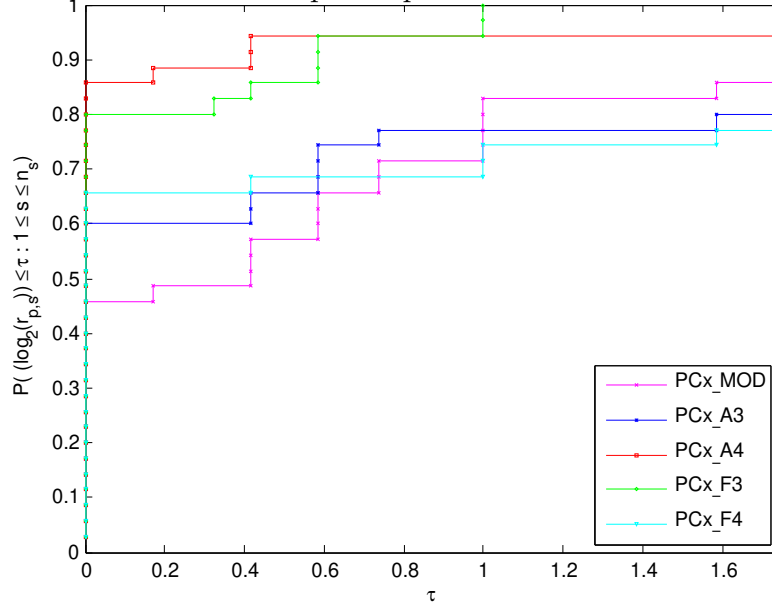
Analisando a [Tabela 6](#), observa-se que em 12 problemas a versão PCx_F3 apresenta uma redução na quantidade de cálculos para obtenção de uma nova matriz \mathbf{B} , aumento em 1 problema e para 2 problemas verificou-se um atraso nas iterações onde os cálculos são realizados, se comparado com a versão PCx_MOD. Para a versão PCx_F4, foram 9 problemas com redução no número de cálculos, 2 problemas com aumento e 1 problema com atraso nas iterações dos cálculos. Dos 12 problemas onde houve redução, para PCx_F3, 8 deles apresentam um sistema linear maior a ser resolvido em relação a PCx_MOD, enquanto que no problema onde foi observado aumento nos cálculos a dimensão do sistema é maior, e um dos 2 problemas onde se verificou atraso nas iterações dos cálculos apresenta um sistema de dimensão maior. Já para a versão PCx_F4, 6 dos 9 problemas onde houve redução apresentam um sistema linear menor, enquanto que os dois problemas que apresentaram aumento dos cálculos formam sistemas lineares maiores a serem resolvidos, e o problema que apresentou atraso nas iterações dos cálculos de uma nova \mathbf{B} forma um sistema linear menor para ser resolvido, se comparado com PCx_MOD.

Por sua vez, a [Tabela 7](#) mostra que a versão PCx_A3 resolve 8 problemas com menos cálculos de uma nova matriz \mathbf{B} do que PCx_MOD, enquanto que para outros 5 problemas há um atraso nas iterações onde tais cálculos ocorrem. A versão PCx_A4 realiza menos cálculos para 11 problemas e atrasa as iterações de tais cálculos para 2 problemas, se comparado com a versão PCx_MOD.

O perfil de desempenho que mostra a métrica da quantidade de cálculos da

uma nova matriz não singular \mathbf{B} é apresentado na Figura 8, onde se vê que a eficiência das versões PCx_F3 e PCx_A4, para esta métrica, é superior à das demais versões.

Figura 8 – Perfil de desempenho para cálculos de uma nova matriz \mathbf{B}



Fonte: Produzido pelo autor.

Para os problemas **pilot** e **pilots**, tomando como base a versão PCx_MOD, as versões PCx_F3 e PCx_F4 necessitaram de mais iterações do método de pontos interiores para solução com status “ótimo”: foram 63 iterações para ambos os problemas com PCx_F3 e 49 para PCx_F4 (também para ambos os problemas), sendo que PCx_MOD precisou de 41 iterações em ambos. Além da desvantagem nas iterações do método de pontos interiores, percebemos desvantagem no número de iterações do método de Gradientes Conjugados pré-condicionado, sendo de 1.815 para PCx_F3 em ambos os problemas e 1.008 para PCx_F4, contra 786 para PCx_MOD. Nenhuma das três versões citadas necessitou calcular mais de uma vez a matriz \mathbf{B} . Contudo, o tempo computacional permaneceu próximo: PCx_F3 resolveu **pilot** com média de 39,48 segundos e **pilots** com 39,48 segundos, PCx_F4 resolveu **pilot** com média de 37,42 segundos e **pilots** com 42,32 segundos, enquanto que PCx_MOD **pilot** com média de 41,45 segundos e **pilots** com 41,69 segundos. PCx_MOD resolve o problema **ken18** em 39 iterações do método de pontos interiores, num total de 4.233 iterações de Gradientes Conjugados pré-condicionado, gastando 786,51 segundos em média sem necessitar de mais de um cálculo para a matriz \mathbf{B} . Já a versão PCx_F4 resolve o mesmo problema em 40 iterações do método de pontos interiores mas com 1.184 iterações de Gradientes Conjugados pré-condicionado e média de 608,87 segundos sem necessitar de mais de um cálculo para a matriz \mathbf{B} . Ainda tomando como base a versão PCx_MOD, PCx_A4 resolve o problema **ken18** com uma iteração do método de pontos interiores a mais, apesar de uma significativa redução na quantidade de iterações do método de Gradientes Conjugados pré-condicionado, sem a necessidade

de calcular a matriz **B** mais de uma vez. O tempo gasto foi vantajoso em mais de 200 segundos.

5 Considerações Finais

Temos apresentado experimentos computacionais com um sistema linear alternativo para o pré-condicionador separador para a solução de problemas de programação linear usando o método preditor-corretor de [Mehrotra \(1992\)](#).

Tal como apresentado por [Oliveira e Sorensen \(2005\)](#), o pré-condicionador separador tem sua construção focada para o sistema aumentado, evitando a formação do complemento de Schur. A construção resultou em uma matriz pré-condicionada diagonal por blocos, sendo que a matriz do bloco superior esquerdo foi reduzida a uma matriz definida positiva, que pode ser de uma das formas apresentadas nas expressões [\(2.11a\)](#) e [\(2.11b\)](#).

Os experimentos numéricos foram orientados à implementação da solução do sistema linear que envolve as matrizes [\(2.11a\)](#) e [\(2.11b\)](#), mas com uma estratégia de solução voltada para o sistema aumentado. Contudo, diferente do que acontece com a matriz [\(2.11a\)](#) nas implementações mais conhecidas que envolvem o pré-condicionador separador, não foi possível encontrar um pré-condicionador para o complemento de Schur (equações normais) que fosse equivalente à matriz [\(2.11b\)](#). Desta forma, novas rotinas de recuperação da solução foram adotadas (todas implementadas em linguagem de programação C).

Os primeiros resultados, obtidos a partir da implementação computacional da estratégia de recuperação de solução proposta na [subseção 3.4.1](#), só puderam ser, de alguma maneira, conclusivos se fossem adotadas precisões “mais apertadas” para o método de Gradientes Conjugados pré-condicionado na solução dos sistemas lineares das direções preditora e corretora (respectivamente 10^{-8} e 10^{-12}), precisões estas diferentes das adotadas pela versão PCx_MOD, usada como base nas implementações e comparações. Isto pode ser devido à instabilidade que a matriz **D** gera durante o processo de solução, tal como observado na [subseção 2.5.4](#). Alterações nesta estratégia de solução forneceram as estratégias definidas nas subseções [3.4.2](#) a [3.4.5](#), que apresentaram resultados satisfatórios com a precisão já adotada pelo PCx_MOD. Sendo assim, os resultados dos experimentos numéricos da estratégia da [subseção 3.4.1](#) foram desconsiderados das análises deste trabalho.

Os resultados dos experimentos evidenciam que PCx_F3 e PCx_A4 adotam estratégias mais promissoras que as atuais na solução dos problemas de programação linear selecionados para os testes. As quatro estratégias propostas neste trabalho realizam uma quantidade menor de atualizações da matriz **B** (ou um retardo na atualização é verificado), procedimento este que pode ser bastante caro do ponto de vista do esforço computacional. Além disso, de maneira geral, as versões que implementam as estratégias de recuperação de solução propostas necessitam de uma quantidade menor de iterações de gradientes

conjugados para a convergência das soluções dos sistemas lineares das direções preditora e corretora.

Estas características podem estar aliadas a uma melhoria no condicionamento das matrizes (3.24), (3.36), (3.48) e (3.51) em relação à matriz (2.11b) e, conseqüentemente, um melhor comportamento do método de Gradientes Conjugados pré-condicionado utilizando tais matrizes, para muitos dos problemas avaliados. Em especial, as matrizes (3.24) e (3.51) se aproximam de uma matriz diagonalmente dominante, como pode ser observado a partir da discussão das propriedades das matrizes \mathbf{D}_B , \mathbf{D}_B^{-1} , \mathbf{D}_N e \mathbf{D}_N^{-1} da subseção 2.5.4.

Tal como observado por Bocanegra (2005, p 88), determinar o momento adequado para a troca de fases tem importância vital para o bom funcionamento da estratégia híbrida de pré-condicionamento, e esta tem sido uma dificuldade enfrentada pelo autor na seleção dos problemas teste. Como a rotina de execução dos problemas envolve a declaração do parâmetro $\bar{\eta}$ e este, por sua vez, é capaz de determinar o momento da troca, encontrar um valor ótimo para cada problema pode ser uma tarefa inviável, dependendo dos problemas a serem abordados. Escolhas inconvenientes para tal parâmetro de entrada podem fazer com que a troca de fases ocorra em um momento impróprio para o pré-condicionador separador. Uma sugestão para trabalhos futuros é o desenvolvimento de uma estratégia que realize a troca de fases em um momento ideal para o pré-condicionador separador, de tal forma que o parâmetro $\bar{\eta}$ não seja necessário.

A partir da discussão da seção subseção 2.5.4, podemos ver que, nas iterações finais do método de pontos interiores, a matriz (2.10) se aproxima de uma matriz estritamente diagonalmente dominante, cujos valores da diagonal principal em módulo são todos iguais a 1. Desta forma, conforme observa Golub e Van Loan (2013, p. 614), o método de Gauss-Seidel converge para qualquer escolha do ponto inicial. Um trabalho futuro desta pesquisa é aplicar o método de Gauss-Seidel no sistema linear que envolve tal matriz e verificar a relação entre esta técnica e a redução estável proposta em (GONZALEZ-LIMA; OLIVEIRA; OLIVEIRA, 2013).

Por fim, este trabalho propôs a solução dos sistemas lineares com foco no sistema aumentado, sendo que dois novos pré-condicionadores foram desenvolvidos. Uma proposta de trabalho futuro é encontrar um pré-condicionador para o sistema de equações normais que seja equivalente aos pré-condicionadores desenvolvidos e resolver os sistemas com foco nas equações normais.

Cite online: Dantzig e Thapa (2003), Dantzig e Thapa (1997). Cite: (DANTZIG; THAPA, 2003; DANTZIG; THAPA, 1997).

Referências

- AL-JEIROUDI, G.; GONDZIO, J.; HALL, J. Preconditioning indefinite systems in interior point methods for large scale linear optimisation. *Optimization Methods and Software*, v. 23, n. 3, p. 345–363, 2008. Disponível em: <<http://dx.doi.org/10.1080/10556780701535910>>. Acesso em: 5 mar. 2016.
- AXELSSON, O. *Iterative solution methods*. Cambridge: Cambridge University Press, 1994.
- BARRETT, R.; BERRY, M.; CHAN, T.; DEMMEL, J.; DONATO, J.; DONGARRA, J.; EIJKHOUT, V.; POZO, R.; ROMINE, C.; VAN DER VORST, H. *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods*. Society for Industrial and Applied Mathematics, 1994. Disponível em: <<http://epubs.siam.org/doi/abs/10.1137/1.9781611971538>>. Acesso em: 25 ago. 2015.
- BOCANEGRA, S. *Algoritmos de Newton-Krylov preconditionados para métodos de pontos interiores*. Tese (Doutorado) — Universidade Federal de Minas Gerais, Dezembro 2005. Disponível em: <<http://www.bibliotecadigital.ufmg.br/dspace/handle/1843/RVMR-6JTN72>>. Acesso em: 14 fev. 2016.
- BOCANEGRA, S.; CAMPOS, F. F.; OLIVEIRA, A. R. L. Using a hybrid preconditioner for solving large-scale linear systems arising from interior point methods. *Computational Optimization and Applications*, v. 36, n. 2, p. 149–164, 2007. ISSN 1573-2894. Disponível em: <<http://dx.doi.org/10.1007/s10589-006-9009-5>>. Acesso em: 16 jan. 2016.
- CAMPOS, F. F.; BIRKETT, N. R. An Efficient Solver for Multi-Right-Hand-Side Linear Systems Based on the CCCG (η) Method with Applications to Implicit Time-Dependent Partial Differential Equations. *SIAM Journal on Scientific Computing*, SIAM, v. 19, n. 1, p. 126–138, 1998. Disponível em: <<http://epubs.siam.org/doi/abs/10.1137/S106482759630382X>>. Acesso em: 4 mar. 2016.
- CAMPOS FILHO, F. F. *Analysis of Conjugate Gradients-Type Methods for Solving Linear Equations*. Tese (Doutorado) — Oxford University Computing Laboratory, Oxford, 1995.
- CASTRO, C. O.; OLIVEIRA, A. R. L. de. Uma proposta para melhorar o desempenho do preconditionador separador no método de pontos interiores. In: *Anais do XLVII Simpósio Brasileiro de Pesquisa Operacional*. Porto de Galinhas, PE: Sociedade Brasileira de Pesquisa Operacional, 2015. p. 2924–2935.
- CHAVES, V. H. C. *Perspectivas históricas da Pesquisa Operacional*. Dissertação (Mestrado) — Universidade Estadual Paulista "Júlio de Mesquita Filho" — UNESP, 2011. Disponível em: <<http://acervodigital.unesp.br/handle/unesp/159499>>. Acesso em: 15 fev. 2016.
- CHVÁTAL, V. *Linear Programming*. New York: W. H. Freeman and Company, 1983. (A series of books in the mathematical sciences).
- CZYZYK, J.; MEHROTRA, S.; WAGNER, M.; WRIGHT, S. J. *PCx User Guide (Version 1.1)*. [S.l.], 1997. Disponível em: <<http://pages.cs.wisc.edu/~swright/PCx/doc/PCx-user.pdf>>. Acesso em: 28 set. 2015.

- DANTZIG, G. B.; THAPA, M. N. *Linear Programming 1: Introduction*. New York: Springer, 1997. (Springer Series in Operations Research and Financial Engineering).
- _____. *Linear Programming 2: Theory and extensions*. New York: Springer, 2003. (Springer Series in Operations Research and Financial Engineering).
- DENNIS JUNIOR, J. E.; SCHNABEL, R. B. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Philadelphia: SIAM, 1996. (Classics in Applied Mathematics, v. 16).
- DOLAN, D. E.; MORÉ, J. J. Benchmarking optimization software with performance profiles. *Mathematical Programming*, v. 91, n. 2, p. 201–213, 2002. ISSN 1436-4646. Disponível em: <<http://dx.doi.org/10.1007/s101070100263>>. Acesso em: 13 fev. 2016.
- GHIDINI, C. T. L. S.; OLIVEIRA, A. R. L.; SORENSEN, D. C. Computing a hybrid preconditioner approach to solve the linear systems arising from interior point methods for linear programming using the conjugate gradient method. *Annals of Management Science*, v. 3, n. 1, p. 43–64, maio 2014. ISSN 2161-5012.
- GOLDMAN, A. J.; TUCKER, A. W. Theory of Linear Programming. In: KUHN, H. W.; TUCKER, A. W. (Ed.). *Linear Inequalities and Related Systems*. Princeton: Princeton University Press, 1956, (Annals of Mathematical Studies, 38). p. 53–97.
- GOLUB, G. H.; VAN LOAN, C. F. *Matrix Computations*. Baltimore: The Johns Hopkins University Press, 2013. (Johns Hopkins studies in the mathematical sciences).
- GONDZIO, J. Multiple centrality corrections in a primal-dual method for linear programming. *Computational Optimization and Applications*, v. 6, n. 2, p. 137–156, 1996. ISSN 1573-2894. Disponível em: <<http://dx.doi.org/10.1007/BF00249643>>. Acesso em: 8 fev. 2016.
- _____. Interior point methods 25 years later. *European Journal of Operational Research*, v. 218, n. 3, p. 587–601, 2012. ISSN 0377-2217. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0377221711008204>>. Acesso em: 15 fev. 2016.
- GONZALEZ-LIMA, M. D.; OLIVEIRA, A. R. L.; OLIVEIRA, D. E. A robust and efficient proposal for solving linear systems arising in interior-point methods for linear programming. *Computational Optimization and Applications*, v. 56, n. 3, p. 573–597, 2013. ISSN 1573-2894. Disponível em: <<http://dx.doi.org/10.1007/s10589-013-9572-5>>.
- GREENBAUM, A. *Iterative methods for solving linear systems*. Philadelphia: SIAM, 1997. (Frontiers in applied mathematics, v. 17).
- HACKBUSCH, W. *Iterative Solution of Large Sparse Systems of Equations*. New York: Springer-Verlag, 1994. (Applied Mathematical Sciences, v. 95).
- JANSEN, B. *Interior Point Techniques in Optimization: Complementary, sensitivity and algorithms*. Dordrecht: Kluwer Academic Publishers, 1997. (Applied Optimization, v. 6).
- KARMARKAR, N. A new polynomial-time algorithm for linear programming. *Combinatorica*, v. 4, n. 4, p. 373–395, 1984. ISSN 1439-6912. Disponível em: <<http://dx.doi.org/10.1007/BF02579150>>. Acesso em: 15 fev. 2016.

- KHACHIYAN, L. Polynomial algorithms in linear programming. *USSR Computational Mathematics and Mathematical Physics*, v. 20, n. 1, p. 53–72, 1980. ISSN 0041-5553. Disponível em: <http://www.sciencedirect.com/science/article/pii/0041555380900610>. Acesso em: 15 fev. 2016.
- KLEE, V.; MINTY, G. J. How good is the simplex algorithm? In: *Inequalities, III (Proc. Third Sympos., Univ. California, Los Angeles, Calif., 1969; dedicated to the memory of Theodore S. Motzkin)*. [S.l.]: Academic Press, New York, 1972. p. 159–175.
- KOJIMA, M.; MEGIDDO, N.; NOMA, T.; YOSHISE, A. *A Unified Approach to Interior Point Algorithms for Linear Complementarity Problems*. Berlin: Springer-Verlag, 1991. (Lecture Notes in Computer Science, v. 538).
- LUEMBERGER, D. G.; YE, Y. *Linear and Nonlinear Programming*. New York: Springer, 2008. (International Series in Operations Research and Management Science, v. 228).
- MATOUSEK, J.; GÄRTNER, B. *Understanding and Using Linear Programming*. Berlin: Springer, 2007. (Universitext).
- MEHROTRA, S. On the Implementation of a Primal-Dual Interior Point Method. *SIAM Journal on Optimization*, Society for Industrial and Applied Mathematics, v. 2, n. 4, p. 575–601, 1992. ISSN 1095-7189. Disponível em: <http://dx.doi.org/10.1137/0802028>. Acesso em: 27 set. 2015.
- MEIJERINK, J. A.; VAN DER VORST, H. A. An iterative solution method for linear systems of which the coefficient matrix is a symmetric M-matrix. *Mathematics of Computation*, American Mathematical Society, v. 31, n. 137, p. 148–162, 1977. ISSN 1088-6842. Disponível em: <http://dx.doi.org/10.1090/S0025-5718-1977-0438681-4>. Acesso em: 27 ago. 2015.
- OLIVEIRA, A. R. L. *A New Class of Preconditioners for Large-Scale Linear Systems from Interior Point Methods for Linear Programming*. Tese (Doutorado) — Rice University, Houston, Abril 1997. Disponível em: <https://scholarship.rice.edu/handle/1911/19149>. Acesso em: 5 mar. 2016.
- OLIVEIRA, A. R. L.; SORENSEN, D. C. A new class of preconditioners for large-scale linear systems from interior point methods for linear programming. *Linear Algebra and its Applications*, v. 394, p. 1–24, 2005. ISSN 0024-3795. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0024379504003799>. Acesso em: 27 set. 2015.
- PULINO, P. *Álgebra Linear e suas Aplicações*. Pulinus, 2012. Disponível em: <http://www.ime.unicamp.br/~pulino>. Acesso em: 9 out 2014.
- ROOS, C.; TERLAKY, T.; VIAL, J.-P. *Interior Point Methods for Linear Optimization*. 2. ed. New York: Springer, 2005. ISBN 978-0387-26378-6.
- RUGGIERO, M. A. G.; LOPES, V. L. d. R. *Cálculo Numérico: Aspectos teóricos e computacionais*. 2. ed. São Paulo: Pearson/Makron, 2005. ISBN 85-346-0204-2.
- SAAD, Y. *Iterative methods for sparse linear systems*. 2. ed. Philadelphia: SIAM, 2003.
- VARGA, R. S. *Matrix Iterative Analysis*. Englewood Cliffs: Prentice-Hall, 1962.

VELAZCO, M. I.; OLIVEIRA, A. R. L.; CAMPOS, F. F. A Note on Hybrid Preconditioners for Large-scale Normal Equations Arising from Interior-point Methods. *Optimization Methods Software*, Taylor & Francis, Inc., Bristol, PA, USA, v. 25, n. 2, p. 321–332, abr. 2010. ISSN 1055-6788. Disponível em: <<http://dx.doi.org/10.1080/10556780902992829>>. Acesso em: 16 jan. 2016.

WATKINS, D. S. *Fundamentals of Matrix Computations*. New York: John Wiley, 2010. (Pure and Applied Mathematics).

WRIGHT, S. J. *Primal-Dual Interior-Point Methods*. Philadelphia: SIAM, 1997. ISBN 978-0-898713-82-4.

ZIEGLER, G. M. *Lectures on Polytopes*. New York: Springer, 2006. (Graduate Texts in Mathematics, v. 152).

Apêndices

APÊNDICE A – Método de Newton

Podemos dizer que o método de Newton tem sua motivação na garantia e aceleração da convergência do método de Ponto Fixo para funções reais. Considere $f: \mathbb{R} \rightarrow \mathbb{R}$ uma função real contínua no intervalo $[a, b]$, intervalo este que contém uma raiz da equação $f(x) = 0$, digamos ξ . O método de Ponto Fixo consiste em tomar uma forma equivalente $x = \phi(x)$ e, a partir de uma aproximação inicial $x^{(0)}$, gerar uma sequência de iteradas $\{x^{(k)}\}$, aproximações para ξ , usando a relação $x^{(k+1)} = \phi(x^{(k)})$, pois a função $\phi(x)$ é tal que $f(\xi) = 0$ se, e somente se, $\phi(\xi) = \xi$. A função ϕ é chamada *função de iteração* e sua forma geral é $\phi(x) = x + A(x)f(x)$, desde que em ξ , $A(\xi) \neq 0$ (RUGGIERO; LOPES, 2005).

Teorema A.1. *Seja ξ uma raiz da equação $f(x) = 0$, isolada em um intervalo $I \subseteq \mathbb{R}$ centrado em ξ . Considere $\phi(x)$ uma função de iteração para a equação $f(x) = 0$. Se*

- (i) $\phi(x)$ e $\phi'(x)$ são contínuas em I ;
- (ii) $|\phi'(x)| \leq M < 1$ para todo $x \in I$;
- (iii) $x^{(0)} \in I$,

então a sequência $\{x^{(k)}\}$, gerada pelo processo iterativo $x^{(k+1)} = \phi(x^{(k)})$, converge para ξ .

Demonstração. Vide Ruggiero e Lopes (2005, p. 58). □

Ruggiero e Lopes (2005, p. 66) observam que a convergência do método de ponto Fixo é mais rápida para pequenos valores de $|\phi'(x)|$. Assim, o método de Newton escolhe uma função de iteração tal que $\phi'(\xi) = 0$. Partindo da equação $f(x) = 0$ e da forma geral para $\phi(x)$, queremos $A(x)$ tal que $\phi'(\xi) = 0$. Temos

$$\begin{aligned} \phi(x) &= x + A(x)f(x) & \Rightarrow & \phi'(x) = 1 + A'(x)f(x) + A(x)f'(x) \\ \phi'(\xi) &= 1 + A'(\xi)f(\xi) + A(\xi)f'(\xi) & \Rightarrow & \phi'(\xi) = 1 + A(\xi)f'(\xi) \end{aligned}$$

Logo, $\phi'(\xi) = 0$ se, e somente se, $A(\xi) = -\frac{1}{f'(\xi)}$, donde vem $A(x) = -\frac{1}{f'(x)}$. Desta forma, a função de iteração utilizada pelo método de Newton é dada por $\phi(x) = x - \frac{f(x)}{f'(x)}$.

Teorema A.2. *Sejam $f(x)$, $f'(x)$ e $f''(x)$ contínuas em um intervalo I que contenha a raiz $x = \xi$ de $f(x) = 0$ e suponha que $f'(\xi) \neq 0$. Então, existe um intervalo $\bar{I} \subset I$ contendo ξ , tal que se $x^{(0)} \in \bar{I}$, a sequência $\{x^{(k)}\}$, gerada por $x^{(k+1)} = x^{(k)} - \frac{f(x^{(k)})}{f'(x^{(k)})}$, convergirá para ξ .*

Demonstração. Vide [Ruggiero e Lopes \(2005, p. 69\)](#). \square

Geometricamente, o método de Newton constrói uma linearização local no ponto atual: dado $(x^{(k)}, f(x^{(k)}))$, fazemos $L_{(k)}(x^{(k)}) = f(x^{(k)}) + f'(x)(x - x^{(k)})$, que define a reta tangente à curva no ponto dado. Encontrando o zero do modelo linear L , obtém-se $L_{(k)}(x) = 0$ se, e somente se, $x = x^{(k)} - \frac{f(x^{(k)})}{f'(x^{(k)})}$. Finalmente, pomos $x^{(k+1)} = x$.

Usando a motivação de se construir um modelo localmente linear, considere $\mathbf{F}: \mathcal{D} \subset \mathbb{R}^n \rightarrow \mathbb{R}^n$, $\mathbf{F} = (f_1, f_2, \dots, f_n)^\top$, uma função não linear definida num conjunto aberto \mathcal{D} , com derivadas contínuas nesse conjunto, e suponha que exista $x^* \in \mathcal{D}$ tal que $\mathbf{F}(x^*) = 0$. A matriz de derivadas parciais de \mathbf{F} , denominada *matriz Jacobiana* é

$$\mathbf{J}(x) = \begin{bmatrix} \nabla f_1(x)^\top \\ \nabla f_2(x)^\top \\ \vdots \\ \nabla f_n(x)^\top \end{bmatrix} = \begin{bmatrix} \frac{\partial f_1(x)}{\partial x_1} & \frac{\partial f_1(x)}{\partial x_2} & \dots & \frac{\partial f_1(x)}{\partial x_n} \\ \frac{\partial f_2(x)}{\partial x_1} & \frac{\partial f_2(x)}{\partial x_2} & \dots & \frac{\partial f_2(x)}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_n(x)}{\partial x_1} & \frac{\partial f_n(x)}{\partial x_2} & \dots & \frac{\partial f_n(x)}{\partial x_n} \end{bmatrix}.$$

Temos que, conhecida uma aproximação $x^{(k)} \in \mathcal{D}$ para $x \in \mathcal{D}$, existe $c_i \in \mathcal{D}$ tal que

$$f_i(x) \approx f_i(x^{(k)}) + \nabla f_i(c_i)^\top (x - x^{(k)}), \quad i = 1, \dots, n.$$

Aproximando $\nabla f_i(c_i)$ por $\nabla f_i(x^{(k)})$, $i = 1, \dots, n$, temos um modelo localmente linear para $f_i(x)$ em torno de $x^{(k)}$

$$f_i(x) \approx f_i(x^{(k)}) + \nabla f_i(x^{(k)})^\top (x - x^{(k)}), \quad i = 1, \dots, n.$$

Obtemos, assim, o modelo

$$\mathbf{F}(x) \approx \mathbf{L}_{(k)}(x) = \mathbf{F}(x^{(k)}) + \mathbf{J}(x^{(k)})^\top (x - x^{(k)}).$$

Para a nova aproximação, encontramos o zero da linearização

$$\mathbf{L}_{(k)}(x) = 0 \quad \Leftrightarrow \quad \mathbf{J}(x^{(k)})(x - x^{(k)}) = -\mathbf{F}(x^{(k)}),$$

e fazendo $\nu^{(k)} = (x - x^{(k)})$ solução do sistema linear

$$\mathbf{J}(x^{(k)})\nu = -\mathbf{F}(x^{(k)}),$$

a nova aproximação fica $x^{(k+1)} = x^{(k)} + \nu^{(k)}$. [Dennis Junior e Schnabel \(1996, p. 90\)](#) mostram que, sob condições adequadas para \mathbf{F} , \mathbf{J} e o ponto inicial, a sequência de iteradas gerada pelo método de Newton via $x^{(k+1)} = x^{(k)} - \mathbf{J}(x^{(k)})^{-1}\mathbf{F}(x^{(k)})$ converge quadraticamente para x^* .

APÊNDICE B – Método dos Gradientes

Conjugados Pré-condicionado

Considere $\mathbf{A} \in \mathbb{M}_n(\mathbb{R})$ uma matriz simétrica e definida positiva e $b \in \mathbb{R}^n$. Suponha que queiramos encontrar a solução x^* para o sistema linear

$$\mathbf{A}x = b. \quad (\text{B.1})$$

É possível mostrar que este problema é equivalente ao problema de minimização

$$\min_{x \in \mathbb{R}^n} J(x), \quad (\text{B.2})$$

onde o funcional $J: \mathbb{R}^n \rightarrow \mathbb{R}$ é dado por $J(x) := \frac{1}{2}\langle \mathbf{A}x, x \rangle - \langle b, x \rangle$, com $\langle \cdot, \cdot \rangle$ o produto interno usual do \mathbb{R}^n . Isto se deve ao fato de J ser convexo e, assim, seu gradiente é dado por $\nabla J(x) := \mathbf{A}x - b$ no ponto $x \in \mathbb{R}^n$. Além disso, a matriz Hessiana do funcional J no ponto $x \in \mathbb{R}^n$ é $H(x) := \mathbf{A}$ e \mathbf{A} é uma matriz definida positiva. Desta forma, se \bar{x} for uma aproximação para a solução do problema de minimização (B.2), então será uma aproximação para a solução do sistema linear (B.1) (PULINO, 2012; GOLUB; VAN LOAN, 2013).

Sejam $\|v\|_{\mathbf{A}} := \sqrt{\langle \mathbf{A}v, v \rangle}$ uma norma e $\langle u, v \rangle_{\mathbf{A}} := \langle \mathbf{A}u, v \rangle$ e um produto interno em \mathbb{R}^n . Como $J(\bar{x}) = \frac{1}{2}\langle \mathbf{A}\bar{x}, \bar{x} \rangle - \langle b, \bar{x} \rangle = \frac{1}{2}\langle \mathbf{A}(\bar{x} - x^*), (\bar{x} - x^*) \rangle - \frac{1}{2}\langle \mathbf{A}^{-1}b, b \rangle$ e $J(x^*) = -\frac{1}{2}\langle \mathbf{A}^{-1}b, b \rangle$, segue que $J(\bar{x}) = \frac{1}{2}\|\bar{x} - x^*\|_{\mathbf{A}}^2 + J(x^*)$ (GOLUB; VAN LOAN, 2013).

B.1 Método da Máxima Descida

Como o mínimo do funcional J situa-se na direção de $-\nabla J(x^{(k)}) = b - \mathbf{A}x^{(k)}$ a partir do ponto $x^{(k)}$, então, se o resíduo $r^{(k)} = b - \mathbf{A}x^{(k)} \neq 0$, existirá um parâmetro $\lambda \in \mathbb{R}$ tal que $J(x^{(k)} + \lambda r^{(k)}) < J(x^{(k)})$. Desta forma, escolhemos λ de modo a obtermos o mínimo de J na variedade linear $\mathcal{S}_k := \{z \in \mathbb{R}^n; z = x^{(k)} + \lambda r^{(k)}, \lambda \in \mathbb{R}\}$, obtendo $x^{(k+1)} = x^{(k)} + \lambda_k r^{(k)}$ tal que $J(x^{(k)}) = \min\{J(z), z \in \mathcal{S}_k\}$. Para isto, definimos $\phi(\lambda) := J(x^{(k)} + \lambda r^{(k)})$, $\lambda \in \mathbb{R}$, e fazemos $\phi'(\lambda) = 0$ para encontrar o minimizador λ_k . Temos, então, que $\lambda_k = \frac{\langle r^{(k)}, r^{(k)} \rangle}{\langle \mathbf{A}r^{(k)}, r^{(k)} \rangle}$, com $r^{(k)} \neq 0$. Usando o fato de que \mathbf{A} é uma matriz definida positiva, temos que $\lambda_k > 0$.

Conforme explica Pulino (2012, p. 538) “ $x^{(k+1)} = \text{span}\{r^{(0)}, r^{(1)}, \dots, r^{(k)}\} + x^{(0)}$ e $x^{(k)}$ é uma sequência minimizante para o funcional J , isto é, $J(x^{(0)}) > \dots > J(x^{(k)}) > J(x^{(k+1)}) > \dots > J(x^*)$, para todo $k = 0, 1, 2, 3, \dots$ ”. O autor alerta, ainda, que a

convergência se torna lenta no caso de $\kappa_2(\mathbf{A})$ ser muito grande e afirma que uma explicação algébrica para este fato é a proximidade das direções dos resíduos. O [Algoritmo 4](#) resume o método.

Algoritmo 4: Método da Máxima Descida

Entrada: $x^{(0)} \in \mathbb{R}^n$, aproximação para (x^*)

Saída: x^* solução de (B.1)

início

para $k = 0, 1, 2, \dots$ **faça**

$r^{(k)} \leftarrow b - \mathbf{A}x^{(k)}$;

$\lambda_k \leftarrow \frac{\langle r^{(k)}, r^{(k)} \rangle}{\langle \mathbf{A}r^{(k)}, r^{(k)} \rangle}$;

$x^{(k+1)} \leftarrow x^{(k)} + \lambda_k r^{(k)}$;

fim

fim

B.2 Método dos Gradientes Conjugados

Introduziremos a ideia de *espaço afim*. Dado $u \in \mathbb{R}^n$ e $\mathcal{V} \subset \mathbb{R}^n$ um subespaço, então $u + \mathcal{V} := \{x \in \mathbb{R}^n; x = u + v, v \in \mathcal{V}\}$ é um espaço afim. Note que, no método da máxima descida, o k -ésimo passo de otimização é sobre o espaço $x^{(k)} + \text{span}\{\nabla J(x^{(k)})\}$. Podemos escolher um subespaço $\text{span}\{p^{(0)}, p^{(1)}, \dots, p^{(k)}\}$ onde as direções $p^{(k)}$ são tais que tornam o processo iterativo passível de convergir em n passos, onde n é a dimensão do sistema linear (B.1) ([GOLUB; VAN LOAN, 2013](#)).

Considere o produto interno $\langle \cdot, \cdot \rangle_{\mathbf{A}}$ introduzido anteriormente, onde \mathbf{A} é a matriz hessiana do funcional J . Construiremos direções consecutivas $p^{(k)}$ e $p^{(k+1)}$ \mathbf{A} -conjugadas, isto é, $\langle p^{(k)}, p^{(k+1)} \rangle_{\mathbf{A}} = \langle \mathbf{A}p^{(k)}, p^{(k+1)} \rangle = 0$, com $p^{(0)} = -\nabla J(x^{(0)}) = r^{(0)}$. Ou seja, determinaremos direções $p^{(k+1)} = r^{(k+1)} + \beta_k p^{(k)}$ de modo que $p^{(k)}$ e $p^{(k+1)}$ sejam \mathbf{A} -conjugadas. Tem-se, então, que

$$\beta_k = -\frac{\langle \mathbf{A}r^{(k+1)}, p^{(k)} \rangle}{\langle \mathbf{A}p^{(k)}, p^{(k)} \rangle} = -\frac{\langle r^{(k+1)}, \mathbf{A}p^{(k)} \rangle}{\langle \mathbf{A}p^{(k)}, p^{(k)} \rangle} \quad (\text{B.3})$$

Esta configuração nos diz que a direção $p^{(k+1)}$ é a projeção ortogonal do resíduo $r^{(k+1)}$ sobre o subespaço gerado pela direção $p^{(k)}$ em relação ao produto interno $\langle \cdot, \cdot \rangle_{\mathbf{A}}$ ([PULINO, 2012](#)).

Notemos que

$$r^{(k+1)} = b - \mathbf{A}x^{(k+1)} = b - \mathbf{A}x^{(k)} - \lambda_k \mathbf{A}p^{(k)} = r^{(k)} - \lambda_k \mathbf{A}p^{(k)}. \quad (\text{B.4})$$

Por sua vez, segue que $\langle p^{(k)}, r^{(k)} \rangle = \langle r^{(k)}, r^{(k)} \rangle + \beta_{k-1} \langle p^{(k-1)}, r^{(k)} \rangle = \langle r^{(k)}, r^{(k)} \rangle$, uma vez que $\langle p^{(k-1)}, r^{(k)} \rangle = 0$, visto que $r^{(k)} = -\nabla J(x^{(k)})$, onde $x^{(k)}$ é o ponto de mínimo do

funcional J na direção de $p^{(k-1)}$. Assim, temos

$$\lambda_k = \frac{\langle r^{(k)}, r^{(k)} \rangle}{\langle \mathbf{A}p^{(k)}, p^{(k)} \rangle}. \quad (\text{B.5})$$

Finalmente, da Equação B.4 segue que $\mathbf{A}p^{(k)} = \frac{r^{(k)} - r^{(k+1)}}{\lambda_k}$ que substituindo na Equação B.3 e usando a Equação B.5 obtemos $\beta_k = \frac{\langle r^{(k+1)}, r^{(k+1)} \rangle}{\langle r^{(k)}, r^{(k)} \rangle}$. O método pode, então, ser resumido no Algoritmo 5 (PULINO, 2012).

Algoritmo 5: Método dos Gradientes Conjugados

Entrada: $x^{(0)} \in \mathbb{R}^n$, aproximação para (x^*)

Saída: x^* solução de (B.1)

início

$r^{(0)} = b - \mathbf{A}x^{(0)};$

$p^{(0)} = r^{(0)};$

para $k = 0, 1, 2, \dots$ **faça**

$\lambda_k \leftarrow \frac{\langle r^{(k)}, r^{(k)} \rangle}{\langle \mathbf{A}p^{(k)}, p^{(k)} \rangle};$

$x^{(k+1)} \leftarrow x^{(k)} + \lambda_k p^{(k)};$

$r^{(k+1)} \leftarrow r^{(k)} - \lambda_k \mathbf{A}p^{(k)};$

$\beta_k \leftarrow \frac{\langle r^{(k+1)}, r^{(k+1)} \rangle}{\langle r^{(k)}, r^{(k)} \rangle};$

$p^{(k+1)} \leftarrow r^{(k+1)} + \beta_k p^{(k)};$

fim

fim

Pulino (2012, p. 542) mostra que dois resíduos consecutivos são ortogonais, que $\{r^{(0)}, r^{(1)}, \dots, r^{(k)}, \dots\}$ é um conjunto ortogonal e que $\{p^{(0)}, p^{(1)}, \dots, p^{(k)}, \dots\}$ é um conjunto \mathbf{A} -conjugado, mostrando que o método de gradientes conjugados converge para a solução exata do sistema (B.1) em n passos.

B.3 Método dos Gradientes Conjugados aplicado a um sistema linear pré-condicionado

Golub e Van Loan (2013, p. 630) mostram que se $\mathbf{A} = \mathbf{I}_n + \mathbf{U}\mathbf{D}\mathbf{U}^\top$ é definida positiva com $\mathbf{U} \in \mathbb{M}_{n \times r}(\mathbb{R})$ e $\mathbf{D} \in \mathbb{M}_r(\mathbb{R})$ com $r < n$ então o método de gradientes conjugados aplicado ao sistema (B.1) converge para a solução exata do sistema linear em até $r + 1$ iterações. Desta forma, o método de gradientes conjugados converge rapidamente se $\kappa_2(\mathbf{A}) \approx 1$ ou se a matriz \mathbf{A} tiver poucos autovalores distintos. Isto nos conduz às ideias de pré-condicionamento de sistemas lineares descrito na seção 2.2.

Assim, considere o sistema (B.1) e $\mathbf{C} \in \mathbf{M}_n(\mathbb{R})$ uma matriz simétrica e definida positiva para ser o pré-condicionador de (B.1), ou seja, $\bar{\mathbf{A}} = \mathbf{C}^{-1}\mathbf{A}\mathbf{C}^{-1}$, e apliquemos o método de gradientes conjugados a $\bar{\mathbf{A}}\bar{x} = \bar{b}$, onde $\bar{x} = \mathbf{C}x$ e $\bar{b} = \mathbf{C}^{-1}b$. Note que $x^{(k)} = \mathbf{C}^{-1}\bar{x}^{(k)}$ e que $r^{(k)} = b - \mathbf{A}x^{(k)}$, então temos que $\bar{r}^{(k)} = \bar{b} - \bar{\mathbf{A}}\bar{x}^{(k)} = \mathbf{C}^{-1}(b - \mathbf{A}x^{(k)}) = \mathbf{C}^{-1}r^{(k)}$. Logo, fazendo $\mathbf{M} = \mathbf{C}^2$, obtemos

$$\begin{aligned}\lambda_k &= \frac{\langle \bar{r}^{(k)}, \bar{r}^{(k)} \rangle}{\langle \bar{\mathbf{A}}\bar{p}^{(k)}, \bar{p}^{(k)} \rangle} = \frac{\langle \mathbf{M}^{-1}r^{(k)}, r^{(k)} \rangle}{\langle \mathbf{A}(\mathbf{C}^{-1}\bar{p}^{(k)}), \mathbf{C}^{-1}\bar{p}^{(k)} \rangle}, \\ \mathbf{C}x^{(k+1)} &= \mathbf{C}x^{(k)} + \lambda_k \bar{p}^{(k)}, \\ \mathbf{C}^{-1}r^{(k+1)} &= \mathbf{C}^{-1}r^{(k)} - \lambda_k \mathbf{C}^{-1}\mathbf{A}\mathbf{C}^{-1}\bar{p}^{(k)}, \\ \beta_k &= \frac{\langle \mathbf{M}^{-1}r^{(k+1)}, r^{(k+1)} \rangle}{\langle \mathbf{M}^{-1}r^{(k)}, r^{(k)} \rangle}, \\ \bar{p}^{(k+1)} &= \mathbf{C}^{-1}r^{(k+1)} + \beta_k \bar{p}^{(k)}.\end{aligned}$$

Se definirmos $p^{(k)} = \mathbf{C}^{-1}\bar{p}^{(k)}$ e $z^{(k)} = \mathbf{M}^{-1}r^{(k)}$ obteremos, por fim, a versão pré-condicionada do método dos gradientes conjugados, descrita no [Algoritmo 6](#).

Algoritmo 6: Método dos Gradientes Conjugados Pré-condicionado

Entrada: $x^{(0)} \in \mathbb{R}^n$, aproximação para (x^*)

Saída: x^* solução de (B.1)

início

$r^{(0)} = b - \mathbf{A}x^{(0)};$

Obtenha $z^{(0)}$ de $\mathbf{M}z^{(0)} = r^{(0)};$

$p^{(0)} = z^{(0)};$

para $k = 0, 1, 2, \dots$ **faça**

$\lambda_k \leftarrow \frac{\langle z^{(k)}, r^{(k)} \rangle}{\langle \mathbf{A}p^{(k)}, p^{(k)} \rangle};$

$x^{(k+1)} \leftarrow x^{(k)} + \lambda_k p^{(k)};$

$r^{(k+1)} \leftarrow r^{(k)} - \lambda_k \mathbf{A}p^{(k)};$

Obtenha $z^{(k+1)}$ de $\mathbf{M}z^{(k+1)} = r^{(k+1)};$

$\beta_k \leftarrow \frac{\langle z^{(k+1)}, r^{(k+1)} \rangle}{\langle z^{(k)}, r^{(k)} \rangle};$

$p^{(k+1)} \leftarrow z^{(k+1)} + \beta_k p^{(k)};$

fim

fim
