Proceeding Series of the Brazilian Society of Computational and Applied Mathematics, v. 8, n. 1, 2021.

Trabalho apresentado no XL CNMAC, Evento Virtual - Co-organizado pela Universidade do Mato Grosso do Sul (UFMS).

Proceeding Series of the Brazilian Society of Computational and Applied Mathematics

Um método de pontos interiores para resolução de problemas lineares discretos mal-postos

Emídio Santos Portilho Júnior[1]
CECE/UNIOESTE, Foz do Iguaçu, PR
Aurelio Ribeiro Leite de Oliveira[2]
IMECC/UNICAMP, Campinas, PR

Resumo. Dada a importância e a dificuldade em se obter resultados satisfatórios via métodos diretos para solução de problemas lineares discretos mal-postos oriundos da discretização de problemas inversos lineares. Neste trabalho, nós retomamos o método de pontos interiores do tipo Preditor-Corretor apresentado em [5] que aproxima o problema de regularização de Tikhonov por um problema de programação quadrática através de uma formulação Primal-Dual com barreira logarítmica. Este método Preditor-Corretor nos leva a sistemas de equações normais que são resolvidos pelo método dos gradientes conjugados precondicinado com o precondiconador separador. Neste trabalho, a fim de reduzir o número de iterações de pontos interiores e do método dos gradientes conjugados precondicionado [8], propomos a utilização do precondicionador Fatoração Controlada de Cholesky [1].

Palavras-chave. Regularização de Tikhonov, Programação Quadrática, Métodos de Pontos Interiores.

1 Introdução

A discretização de um problema inverso comumente fornece um sistema linear de equações

$$Ax = b,\ A \in \mathbb{R}^{m\times n},\ x \in \mathbb{R}^n,\ b \in \mathbb{R}^m \tag{1}$$

Quando os valores singulares da matriz dos coeficientes deste sistema se acumulam próximos à origem e decaem gradualmente a zero, isso torna a matriz severamente mal-condicionada. Tais sistemas são frequentemente chamados de problemas lineares discretos e mal-postos [6].

O método de regularização de Tikhonov é um dos mais antigos e mais populares métodos de regularização. Este método aproxima o sistema linear (1) pelo sistema regularizado

$$(A^TA + \alpha^2 I)x = A^Tb, \tag{2}$$

onde $\alpha \geq 0$ é o parâmetro de regularização que determina a quantidade de regularização e I é o operador identidade.

Em muitos problemas, a matriz A tem muitos valores singulares pequenos, o que acaba acarretando um mal condicionamento da matriz A e consequentemente de A^TA, visto que os autovalores de A^TA são os valores singulares de A elevados ao quadrado. Além disso, em geral o vetor b

[1]emidio.portilho@gmail.com.
[2]aurelio@ime.unicamp.br.

DOI: 10.5540/03.2021.008.01.0497

está contaminado por erros de medidas (ruídos). Neste trabalho, assumiremos que os erros estão restritos ao lado direito do sistema (1), isto é, dado b podemos escrever

$$b = \overline{b} + \mathbf{e} \qquad \overline{b} = A\overline{x},$$

onde $\overline{b}$ representa os dados exatos não perturbados, $\overline{x} = A^{\dagger}\overline{b}$ representa a solução exata e o vetor $\mathbf{e}$ representa os erros nos dados.

A solução x de um problema mal-posto obtida via métodos diretos está com frequência associada a um valor elevado de $||x||_2$, veja [7]. Em vista disso o método apresentado por Tikhonov e Arsenin em [7] tem por objetivo obter soluções do sistema de equações lineares $Ax = b$ com norma pequena, do que é proposto resolver o problema de minimização:

$$\min_{x} ||x||_2, \text{ sujeito a } \quad ||b - Ax||_2 \quad \leq \mathbf{T}_0 \tag{3}$$

onde $\mathbf{T}_0$ é o valor máximo da norma do resíduo que estamos dispostos a aceitar. Sendo assim, estamos diante de um problema de minimização com restrições de desigualdade.

No trabalho [5], o problema de minimização (3) foi aproximado por um problema de programação quadrática através de uma formulação Primal-Dual que deu origem a um método Preditor-Corretor capaz de obter uma solução para o problema de minimização (3). Este método Preditor-Corretor da origem a sistemas de equações normais que são resolvidos via método dos gradientes conjugados precondicinado com o precondiconador separador. Neste trabalho, a fim de reduzir o número de iterações de pontos interiores e do método dos gradientes conjugados precondicionado [8], propomos a utilização do precondicionador Fatoração Controlada de Cholesky [1].

2 Métodos de Pontos Interiores

Em [5], para desenvolver o método de pontos interiores (MPI) para o problema de regularização de Tikhonov, nós aproximamos o problema de minimização (3) pelo problema (4), que procura soluções com propriedades similares as requeridas pelo problema (3):

$$\begin{cases} \min\limits_{x,u,v} & \dfrac{\tau}{2}||x||_2^2 + e^T(u+v) \\ \text{sujeito a} & Ax + u - v = b. \\ & (u,v) \geq 0 \text{ e } x \in \mathbb{R}^n. \end{cases} \tag{4}$$

Trata-se de um problema de programação quadrática com restrições lineares, onde A é uma matriz de posto completo $m \times n$, b e x são vetores colunas de dimensões apropriadas. Além disso, e é um vetor com todas as entradas iguais a 1 e u e v são variáveis não negativas. O valor $\tau > 0$ representa um parâmetro de penalização para valores grandes de $||x||_2$.

Associado ao problema primal (4), temos o problema de programação quadrática dual:

$$\begin{cases} \max\limits_{x,y,z,w} & -\dfrac{\tau}{2}||x||^2 + y^T b \\ \text{sujeito a} & \tau x - A^T y = 0 \\ & e - y - z = 0 \\ & e + y - w = 0 \\ & (z,w) \geq 0 \text{ e } y \in \mathbb{R}^m. \end{cases} \tag{5}$$

Aplicando os mesmos passos de [9] aos problemas (4) e (5) obtemos um método de pontos interiores do tipo preditor-corretor cujo a estrutura é dada pelo Algoritmo 1.

Algoritmo 1: Etapas do PCM

Dados: $(x^0, u^0, v^0, y^0, z^0, w^0)$ com $(u^0, v^0, z^0, w^0) > 0$
Resultado: Solução do PCM
para $k = 1, 2, ...$ **faça**

1. $\mu^k = \frac{u_k^T z_k + v_k^T w_k}{2n}$;
2. $\dot{q}^k = r_b^k + U^k(e - (Z^k)^{-1} r_z^k) + V^k(-e^k + (W^k)^{-1} r_w^k)$
3. $(\Theta^k)^{-1} = (Z^k)^{-1}(U^k) + (W^k)^{-1} V^k$
4. $\triangle x^{af} = (A^T \Theta^k A + \tau I_n)^{-1}(A^T \Theta^k \dot{q}^k + r_y^k)$
5. $\triangle y^{af} = \Theta^k(\dot{q}^k - A\triangle x^{af})$
6. $\triangle z^{af} = -r_z^k - \triangle y^{af}$
7. $\triangle w^{af} = \triangle y^{af} - r_w^k$
8. $\triangle u^{af} = -U^k e + (Z^k)^{-1} U^k r_z + (Z^k)^{-1} U^k \triangle y^{af}$
9. $\triangle v^{af} = -V^k e + (W^k)^{-1} V^k r_w - (W^k)^{-1} V^k \triangle y^{af}$
10. Obtenha α_u^{af}, α_v^{af}, α_z^{af} e α_w^{af}
11. $\alpha^{af} = \beta^k \min\left\{\alpha_u^{af}, \alpha_v^{af}, \alpha_z^{af}, \alpha_w^{af}\right\}$ onde $\beta^k \in (0, 1)$
12. $\mu_{af}^k = \frac{(u^k + \alpha_u^{af}\triangle u^{af})(z^k + \alpha_z^{af}\triangle z^{af}) + (v^k + \alpha_v^{af}\triangle v^{af})(w^k + \alpha_w^{af}\triangle w^{af})}{2n}$
13. $\sigma_k = \left(\dfrac{\mu_{af}^k}{\mu^k}\right)^3$
14. $\ddot{q} = (Z^k)^{-1}(-\sigma_k \mu^k e + \triangle U^{af} \triangle Z^{af} e) + (W^k)^{-1}(\sigma_k \mu^k e - \triangle V^{af} \triangle W^{af} e)$
15. $(\Theta^k)^{-1} = (Z^k)^{-1}(U^k) + (W^k)^{-1} V^k$
16. $\triangle x^{cc} = (A^T \Theta^k A + \tau I_n)^{-1}(A^T \Theta^k \ddot{q}^k)$
17. $\triangle y^{cc} = \Theta^k(\ddot{q}^k - A\triangle x^{cc})$
18. $\triangle z^{cc} = -\triangle y^{cc}$
19. $\triangle w^{cc} = \triangle y^{cc}$
20. $\triangle u^{cc} = (Z^k)^{-1}(\sigma_k \mu^k e - \triangle U^{af} \triangle Z^{af} e + U^k \triangle y^{cc})$
21. $\triangle v^{cc} = (W^k)^{-1}(\sigma_k \mu^k e - \triangle V^{af} \triangle W^{af} e - V^k \triangle y^{cc})$
22. Obtenha a direção de descida $d^k = d_k^{af} + d_k^{cc}$
23. Calcule o comprimento de passo $\alpha^k = \beta^k \min\{\rho_u, \rho_v, \rho_w, \rho_z\}$ onde:
    $\beta^k \in (0, 1)$, e $\rho_u = \min\limits_i \left\{-\dfrac{u_i^k}{\triangle u_i^k} \mid \triangle u_i^k < 0\right\}$, $\rho_v = \min\limits_i \left\{-\dfrac{v_i^k}{\triangle v_i^k} \mid \triangle v_i^k < 0\right\}$,
    $\rho_w = \min\limits_i \left\{-\dfrac{w_i^k}{\triangle w_i^k} \mid \triangle w_i^k < 0\right\}$, $\rho_z = \min\limits_i \left\{-\dfrac{z_i^k}{\triangle z_i^k} \mid \triangle z_i^k < 0\right\}$;
24. $(\mathbf{x}_{k+1}, \mathbf{y}_{k+1}) = (\mathbf{x}_k, \mathbf{y}_k) + \alpha^k d^k$

fim

Neste algoritmo valem as seguintes igualdades: $\alpha_u^{af} = \text{argmax}\left\{\alpha \in (0,1] : u^k + \alpha\triangle u^{af} > 0\right\}$, $\alpha_v^{af} = \text{argmax}\left\{\alpha \in (0,1] : v^k + \alpha\triangle v^{af} > 0\right\}$, $\alpha_z^{af} = \text{argmax}\left\{\alpha \in (0,1] : z^k + \alpha\triangle z^{af} > 0\right\}$, $\alpha_w^{af} = \text{argmax}\left\{\alpha \in (0,1] : w^k + \alpha\triangle w^{af} > 0\right\}$, $Z^k = diag(z_1^k, z_2^k, \ldots, z_m^k)$, $U^k = diag(u_1^k, u_2^k, \ldots, u_m^k)$, $W^k = diag(w_1^k, w_2^k, \ldots, w_m^k)$, $V^k = diag(v_1^k, v_2^k, \ldots, v_m^k)$, $r_b^k = b - Ax^k - u^k + v^k$, $r_y^k = A^T y^k - \tau x^k$, $r_z^k = z^k + y^k - e$ e $r_w^k = w^k - y^k - e$.

Na etapa 4 do Algoritmo 1 nos deparamos com sistemas de equações normais. Para resolver tais sistemas utilizamos o Método dos Gradientes Conjugados Precondicionado (MGCP) [8].

Nós implementamos duas versões do Algoritmo 1, a primeira versão utilizando o precondicionador separador desenvolvido em [3]. Vamos nos referir a esta combinação como MPCLU. A segunda versão foi implementada utilizando a Fatoração Controlada de Cholesky proposta por [1]. A esta combinação iremos nos referir como MPCFCC.

Mais detalhes sobre as formulações MPCLU e MPCFCC podem ser encontras em [4].

3 Resultados Numéricos

Todos os resultados numéricos foram desenvolvidos em MATLAB R2013b com sistema operacional 64-bit Windows 10, processador Intel Core I7-8550U, 1.99 Ghz, 16 GB de memória RAM. Os códigos para calcular a discretização dos exemplos deste trabalho provêm dos pacotes disponibilizados em [2]. Consideramos que o lado direito dos sistemas está contaminado por ruídos, ou seja, $b = \bar{b} + \mathbf{e}$, em que $\mathbf{e}$ refere-se a um vetor aleatório normalizado escolhido de tal forma que $||\mathbf{e}||/||\bar{b}|| = \epsilon > 0$. Iremos nos referir ao quociente $NL = ||\mathbf{e}||/||\bar{b}||$ como nível de ruído. Para os testes numéricos assumimos $\tau = 5 \times 10^{-3}$.

A fim de comparar a eficiência do MPCFCC e do MPCLU, iremos comparar os resultados obtidos por ambos os métodos na resolução dos problemas de teste Baart, Shaw e Phillips com níveis de ruído $NL = 10^{-3}$, $NL = 10^{-4}$ e $NL = 10^{-3}$ respectivamente. Nós testamos ambos os métodos para várias dimensões dos problemas de teste. Apresentamos os resultados nas tabelas a seguir:

Tabela 1: MPCFCC aplicado à matriz de Baart.

Dimensão	It	Itgcp	Erd	Eri	t_{cpu}
250	8	148	$1,158 \times 10^{-1}$	$2,403 \times 10^{-4}$	$2,809 \times 10^{-1}$
500	9	129	$1,201 \times 10^{-1}$	$1,635 \times 10^{-4}$	$6,022 \times 10^{-1}$
1000	10	131	$7,760 \times 10^{-2}$	$5,332 \times 10^{-5}$	$2,841 \times 10^{+0}$
2000	10	131	$1,058 \times 10^{-1}$	$4,386 \times 10^{-4}$	$2,312 \times 10^{+1}$
5000	11	147	$1,101 \times 10^{-1}$	$2,134 \times 10^{-5}$	$5,811 \times 10^{+2}$

Tabela 2: MPCLU aplicado à matriz de Baart.

Dimensão	It	Itgcp	Erd	Eri	t_{cpu}
250	9	148	$1,145 \times 10^{-1}$	$2,204 \times 10^{-4}$	$4,127 \times 10^{-1}$
500	11	158	$1,300 \times 10^{-1}$	$1,657 \times 10^{-4}$	$5,723 \times 10^{-1}$
1000	10	164	$1,573 \times 10^{-1}$	$6,239 \times 10^{-5}$	$1,742 \times 10^{+0}$
2000	12	201	$7,130 \times 10^{-2}$	$3,557 \times 10^{-5}$	$8,885 \times 10^{+0}$
5000	10	165	$8,060 \times 10^{-2}$	$3,120 \times 10^{-5}$	$5,365 \times 10^{+1}$

Tabela 3: MPCFCC aplicado à matriz de Shaw.

Dimensão	It	Itgcp	Erd	Eri	t_{cpu}
250	13	332	$3,250 \times 10^{-2}$	$1,474 \times 10^{-5}$	$3,200 \times 10^{-1}$
500	13	332	$3,360 \times 10^{-2}$	$9,887 \times 10^{-6}$	$7,339 \times 10^{-1}$
1000	14	354	$3,340 \times 10^{-2}$	$8,180 \times 10^{-6}$	$3,478 \times 10^{+0}$
2000	14	343	$3,350 \times 10^{-2}$	$6,793 \times 10^{-6}$	$2,598 \times 10^{+1}$
3000	15	372	$3,310 \times 10^{-2}$	$6,273 \times 10^{-7}$	$8,920 \times 10^{+1}$

Tabela 4: MPCLU aplicado à matriz de Shaw.

Dimensão	It	Itgcp	Erd	Eri	t_{cpu}
250	12	344	$3,170 \times 10^{-2}$	$1,283 \times 10^{-5}$	$4,074 \times 10^{-1}$
500	12	337	$3,450 \times 10^{-2}$	$1,144 \times 10^{-5}$	$8,760 \times 10^{-1}$
1000	15	396	$3,230 \times 10^{-2}$	$5,710 \times 10^{-6}$	$4,019 \times 10^{+0}$
2000	14	356	$3,310 \times 10^{-2}$	$7,223 \times 10^{-6}$	$1,500 \times 10^{+1}$
3000	15	382	$3,360 \times 10^{-2}$	$6,546 \times 10^{-6}$	$3,363 \times 10^{+2}$

Tabela 5: MPCFCC aplicado à matriz de Phillips.

Dimensão	It	Itgcp	Erd	Eri	t_{cpu}
252	9	285	$1,180 \times 10^{-2}$	$6,052 \times 10^{-4}$	$2,707 \times 10^{-1}$
504	11	377	$1,440 \times 10^{-2}$	$5,890 \times 10^{-4}$	$6,451 \times 10^{-1}$
1024	12	417	$1,480 \times 10^{-2}$	$3,644 \times 10^{-4}$	$3,200 \times 10^{+0}$
2048	13	502	$1,210 \times 10^{-2}$	$4,740 \times 10^{-4}$	$2,557 \times 10^{+1}$
4096	12	458	$1,120 \times 10^{-2}$	$4,235 \times 10^{-4}$	$2,876 \times 10^{+2}$

Tabela 6: MPCLU aplicado à matriz de Phillips.

Dimensão	It	Itgcp	Erd	Eri	t_{cpu}
252	10	385	$1,000 \times 10^{-2}$	$3,143 \times 10^{-4}$	$3,817 \times 10^{-1}$
504	11	429	$1,250 \times 10^{-2}$	$3,563 \times 10^{-4}$	$9,546 \times 10^{-1}$
1024	12	458	$9,000 \times 10^{-3}$	$3,182 \times 10^{-4}$	$4,547 \times 10^{+0}$
2048	13	512	$1,170 \times 10^{-2}$	$3,919 \times 10^{-4}$	$2,302 \times 10^{+1}$
4096	12	466	$1,160 \times 10^{-2}$	$5,213 \times 10^{-4}$	$1,038 \times 10^{+2}$

DOI: 10.5540/03.2021.008.01.0497

A primeira coluna das tabelas nos informa a dimensão das matrizes, a segunda coluna fornece o número de iterações de métodos de pontos interiores que foram necessárias para resolver o problema, ao passo que a terceira fornece o número de iterações do MGCP necessárias durante toda a execução do método de pontos interiores em questão. Uma vez que a solução exata $\mathbf{x}$ é conhecida, as colunas Erd e Eri representam respectivamente os erros relativos cometidos: Erd=$||\mathbf{x}^k - \mathbf{x}||_2/||\mathbf{x}||_2$ e Eri=$||A\mathbf{x}^k - b||_2/||b||_2$. Por fim, t_{cpu} representa o tempo em segundos demandados pelo método para resolução do problema.

Para o problema de Baart as iterações de MPI e iterações de MGCP do método MPCFCC se mostrou competitivo com o método MPCLU, ele também se mostrou competitivo no quesito tempo computacional, exceto para ordem $n = 5000$.

Já para o problema de teste Shaw, o número de iterações de ponto interior e de MGCP dos métodos MPCFCC e MPCLU tiveram resultados bem próximos e podemos dizer o mesmo com relação à precisão dos resultados alcançados pelos mesmos. Na maior parte dos casos, o tempo de processamento do MPCFCC obteve resultados melhores do que o método MPCLU.

Por fim, para o problema de Phillips o número iterações de MPI foram os mesmos em quase todos os casos. Mas, o número de iterações de MGCP foram menores em todos os casos. No entanto, os resultados de tempo de processamento t_{cpu} do MPCFCC se tornaram maiores do que os do MPCLU a partir da ordem $n = 2048$. Isto se deve à necessidade de atualizar o precondicionador Fatoração Controlada de Cholesky quando o mesmo perde eficiência.

4 Conclusões

Neste trabalho, o cálculo das direções de busca do método Preditor-Corretor nos levaram a sistemas de equações normais. Estes sistemas de equações normais foram resolvidos pelo Método dos Gradientes Conjugados Precondicionado. Sendo os precondicionadores utilizados o Precondicionador Separador e a Fatoração Controlada de Cholesky.

De modo geral podemos dizer que o MPCLU e o MPCFCC obtiveram ordem de precisão das soluções bastante similares. Também, o número de iterações de MPI de ambos os métodos se mostraram bem próximos para os exemplos utilizados. No entanto, na maior parte dos casos o método MPCFCC obteve êxito em diminuir o número de iterações de MGCP no decorrer das iterações de MPI.

Concluímos que embora sejam necessários mais testes, os resultados obtidos com a implementação dos métodos no software MATLAB, mostram que o MPCFCC pode ser competitivo com o MPCLU.

Referências

[1] Campos, F. F. Analysis of conjugate gradient-type methods for solving linear equations, Tese deDoutorado, University of Oxford, 1995.

[2] Hansen, P. C. Regularization tools version 4.0 for matlab 7.3. *Numer. Algorithms*, volume 46, pages 189–194, 2007. DOI: 10.1007/s11075-007-9136-9.

[3] Oliveira, A. R. L.; Sorensen, D. A new class of preconditioners for large-scale linear systems from interior point methods for linear programming, *Linear Algebra and its applications*, Elsevier, volume 394, pages 1–24, 2005. DOI: 10.1016/j.laa.2004.08.019.

[4] Portilho Jr, E. S. Métodos de pontos interiores para resolução de problemas de regularização de Tikhonov de grande porte, Tese de Doutorado, Unicamp, 2020.

[5] Portilho Jr, E. S.; Oliveira, A. R. L. Métodos de pontos interiores para resolução de problemas de regularização de Tikhonov de grande porte, *Anais do X Encontro Regional de Matemática Aplicada e Computacional do Rio Grande do Sul – ERMAC-RS*, 2020. ISBN: 978-65-5623-103-7

[6] Rezghi, R. and Hosseini, S. M. A new variant of l-curve for tikhonov regularization, *Journal of Computational and Applied Mathematics*, Elsevier, volume 231, pages 914–924, 2009. DOI: 10.1016/j.cam.2009.05.016.

[7] Tikhonov, A. N.; Aarsenin, V. Y. *Solution of ill-posed problems.* V.H. Winston & Sons, Washington DC, 1977.

[8] Trefethen, L. N.; Bau, D. *Numerical Linear Algebra.* Society for Industrial and Applied Mathematics, 1997.

[9] Zhang, Y. A primal-dual interior point approach for computing l_1 and l_∞ solutions of overdetermined linear systems, *J. Optim. Theory Appl.*, volume 77(2), pages 323–341, 1993. DOI:10.1007/BF00940715.

