Proceeding Series of the Brazilian Society of Applied and Computational Mathematics, v. 6, n. 1, 2018.

Trabalho apresentado no XXXVII CNMAC, S.J. dos Campos - SP, 2017.

Proceeding Series of the Brazilian Society of Computational and Applied Mathematics

Usando grupos no cálculo do precondicionador separador aplicado aos métodos de pontos interiores

Luciana Casacio[1]
Departamento Acadêmico de Matemática, UTFPR, Cornélio Procópio, PR
Aurelio R. L. Oliveira[2]
Departamento de Matemática, Estatística e Computação Científica, UNICAMP, Campinas, SP
Christiano Lyra[3]
Faculdade de Engenharia Elétrica e de Computação, UNICAMP, Campinas, SP

Resumo. Métodos iterativos precondicionados são utilizados para solução dos sistemas lineares dos métodos de pontos interiores com o objetivo final de resolver problemas de otimização linear de grande porte. Durante as iterações dos métodos de pontos interiores, a matriz de coeficientes se torna mal condicionada, ocasionando instabilidade numérica e dificuldades em encontrar a solução, principalmente quando métodos iterativos são utilizados. Assim, a escolha do precondicionador é essencial para o sucesso da abordagem. O trabalho propõe alterações na construção do precondicionador separador; o conceito de grupos e um novo critério de ordenamento das colunas que preserva a estrutura esparsa da matriz de coeficientes original são adotados. Resultados teóricos mostram que a matriz do novo precondicionador separador tem o número de condição limitado. Os estudos de caso mostram que a abordagem é promissora na solução de problemas de otimização linear de grande porte.

Palavras-chave. Métodos de Pontos Interiores, Precondicionador Separador, Métodos Iterativos, Programação Linear.

1 Introdução

Os métodos de pontos interiores (MPI) têm sido amplamente utilizados para a solução de problemas de otimização linear [10]. Suas propriedades teóricas e bons desempenhos computacionais têm motivado o interesse da comunidade científica, principalmente para a solução de problemas de grande porte. Cada iteração dos MPI envolve a solução de um ou mais sistemas lineares, que quase sempre possuem dimensões elevadas e alto grau de esparsidade. Resolver esses sistemas é a parte que demanda maior esforço computacional de cada iteração dos MPI.

Existem várias abordagens para resolver esses sistemas lineares. A maioria das implementações utilizam a fatoração de Cholesky [9]. No entanto, no decorrer das iterações

[1]lucianacasacio@utfpr.edu.br
[2]aurelio@ime.unicamp.br
[3]chrlyra@densis.fee.unicamp.br

dos MPI, a matriz de coeficientes se torna mal condicionada e densa, o que tem motivado o estudo de estratégias que contornem o problema do mau condicionamento. Os métodos iterativos são escolhas atrativas para solução desses sistemas e o uso de bons precondicionadores é essencial para o sucesso dos métodos.

Oliveira e Sorensen [12] mostraram que todo precondicionador para o sistema de equações normais pode gerar um precondicionador para o sistema aumentado, mas o contrário não é verdadeiro. Isso motivou o desenvolvimento de precondicionadores para tais sistemas [1,2,6]. No entanto, com o sistema de equações normais, é possível utilizar o método dos gradientes conjugados, uma vez que a matriz de coeficientes é simétrica e definida positiva. Assim, precondicionadores para o sistema de equações normais ganharam espaço, especialmente para solução de problemas de programação linear por MPI [3].

O precondicionador separador proposto por Oliveira e Sorensen [12] apresenta bons resultados nas iterações finais dos MPI, quando o sistema está muito mal condicionado. Este trabalho apresenta uma nova abordagem para a construção do precondicionador separador que melhora as características de esparsidade dos sistemas de equações através de um novo critério de ordenamento das colunas.

2 Sistemas lineares dos métodos de pontos interiores

Durante os métodos de pontos interiores, é necessário resolver o sistema linear

$$\begin{bmatrix} A & 0 & 0 \\ 0 & A^T & I \\ Z & 0 & X \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix} = \begin{bmatrix} r_p \\ r_d \\ r_c \end{bmatrix}, \tag{1}$$

onde $r_p = b - Ax$, $r_d = c - A^T y - z$ e $r_c = \mu e - XZe$.

Eliminando $\Delta z = X^{-1}(r_c - Z\Delta x)$ da segunda equação de (1), temos:

$$\underbrace{\begin{bmatrix} -D^{-1} & A^T \\ A & 0 \end{bmatrix}}_{\mathcal{A}} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} g \\ r_p \end{bmatrix}, \tag{2}$$

onde $D^{-1} = X^{-1}Z$ and $g = r_d - X^{-1}r_c$. $\mathcal{A}$ é conhecido como sistema aumentado; a matriz é simétrica e indefinida.

Eliminando Δx de (2), o sistema é reduzido ao sistema de equações normais:

$$\underbrace{ADA^T}_{\mathcal{S}} \Delta y = ADg + r_p. \tag{3}$$

A matriz $\mathcal{S} = ADA^T$ é chamada de complemento de Schur, é simétrica e definida positiva. Normalmente a fatoração de Cholesky é aplicada para resolver o sistema. No entanto, durante o processo de pivoteamento, a matriz de coeficientes perde sua estrutura esparsa e frequentemente ocorrem restrições de tempo de processamento e limitação de memória. Devido à isso, os métodos iterativos se tornam uma alternativa atraente e a escolha de um precondicionador eficiente é essencial para o sucesso da abordagem.

DOI: 10.5540/03.2018.006.01.0457

Durante as iterações dos MPI, somente a matriz diagonal D se altera a cada iteração. Além disso, D é altamente mal condicionada, em especial quando os MPI se aproximam da solução ótima.

O objetivo deste estudo é melhorar o condicionamento do sistema. Especificamente, o trabalho estende os resultados apresentados em [7] para o precondicionador separador [12]. O objetivo final do trabalho é obter uma implementação simples, robusta e com tempo computacional adequado.

3 Precondicionadores

O precondicionador separador, proposto por Oliveira e Sorensen, 2005 [12] foi originalmente criado para o sistema aumentado dos MPI. No entanto, pode ser facilmente estendido para o sistema de equações normais. Seu bom desempenho nas iterações finais dos MPI, onde a maioria dos precondicionadores perdem a eficiência, é uma característica desejável.

Seja $A = [B \quad N]P$ onde P é a matriz de permutação e B é não singular. Considere também $PDP^T = \begin{bmatrix} D_B & 0 \\ 0 & D_N \end{bmatrix}$. Sem perda de generalidade assuma $P = I$. Assim, $ADA^T = BD_BB^T + ND_NN^T$.

Agora, multiplicando por $D_B^{-\frac{1}{2}}B^{-1}$ e pós-multiplicando pela sua transposta

$$T = D_B^{-\frac{1}{2}}B^{-1}(ADA^T)B^{-T}D_B^{-\frac{1}{2}} = I + WW^T,$$

onde $W = D_B^{-\frac{1}{2}}B^{-1}ND_N^{\frac{1}{2}}$.

Próximo da solução, pelo menos $n - m$ entradas de D são pequenas. Assim, com uma escolha adequada das colunas de B, as diagonais de D_B^{-1} e D_N passam a ter elementos muito pequenos próximo à solução. Nessa situação, W aproxima-se da matriz nula, a matriz precondicionada aproxima-se da matriz identidade.

Embora a ideia seja simples, o cálculo de B demanda grande esforço computacional [8]. Por outro lado, a mesma matriz B pode ser reusada por mais de uma iteração, não acrescentando custo computacional a cada iteração.

Os autores sugerem a escolha das m primeiras colunas linearmente independentes com a menor norma-1 de AD. Em [13], essa heurística foi aprimorada e resultados melhores foram obtidos utilizando-se a norma-2.

A maneira mais simples de calcular B é através da fatoração LU, trabalhando com atualização atrasada pois quando uma coluna linearmente dependente aparece, ela é eliminada da fatoração e o método prossegue com as próximas colunas, ordenadas conforme a heurística. No entanto, o processo pode levar ao número excessivo de fill-in. Uma estratégia para evitar o problema consiste em interromper a fatoração quando preenchimento permitido for ultrapassado. Nesse caso, as colunas são reordenadas e a fatoração é reiniciada.

A próxima seção apresenta a estratégia desenvolvida a fim de reduzir o custo computacional no cálculo da matriz B.

DOI: 10.5540/03.2018.006.01.0457

3.1 Novo critério de ordenamento

Baseados em [11] e [7], este trabalho propõe reordenar as colunas da matriz A em ordem decrescente $h_j = d_{jj}^{1/2}\|a_j\|$ e dividir as colunas ordenadas em grupos de acordo com um critério de reordenamento.

O processo se inicia ordenando $h_{\delta(1)} \geq h_{\delta(2)} \geq \ldots \geq h_{\delta(n)}$, onde δ é a permutação de $\{1, \ldots n\}$. Para simplificar a notação vamos considerar δ a matriz identidade. Então, particiona-se os escalares h_j em k grupos $G_1, \ldots, G_k$, contendo $n_1, \ldots, n_k$ elementos em cada grupo, tal que

$$\min_{h_j \in G_p} h_j > \max_{h_j \in G_{p+1}} h_j, \quad \text{for } p = 1, \ldots, k-1.$$

Seja $J_i = \{j : h_j \in G_i\}$ o conjunto dos índices dentro de cada grupo. Então, as a_j colunas de A são divididas em k grupos $\{a_j : j \in J_i\}$. A ordem das colunas dentro de cada grupo é arbitrária e pode seguir algum critério escolhido. Neste trabalho, o critério adotado foi a esparsidade. Para compor a base B são selecionadas as m primeiras colunas linearmente independentes após todos os reordenamentos.

Teorema 3.1. *Suponha que B seja a base encontrada e $R = D_B^{-\frac{1}{2}} B^{-1}$ o precondicionador separador. Então*

$$cond(RADA^T R^T) \leq mnC^2\|B^{-1}\|^2, \quad onde$$

$$C = \max\{c_1, \ldots, c_k\},\ c_i = \frac{\max\{h_j : j \in J_i\}}{\min\{h_j : j \in J_i\}},\ i = 1, \ldots, k.$$

A demonstração do Teorema pode ser encontrada em [4].

O Teorema 3.1 mostra que o número de condição da matriz precondicionada considerando a separação das colunas em grupos é limitado. Além disso, esse valor não é influenciado nem pelo número de iterações do método iterativo, nem pelo critério de ordenamento dentro de cada grupo.

4 Estudos de casos

Todos os experimentos foram realizados utilizando o código PCx [5] em linguagem de programação C. A rotina para solução do sistema linear (3) que originalmente era resolvida pela fatoração de Cholesky, foi alterada para o método do gradiente conjugado precondicionado pelo precondicionador separador, adotando o novo critério de ordenamento das colunas de B, proposto na Seção 3.1.

Testes computacionais sobre a quantidade de grupos indicaram que $\sqrt{m}/4$ é um bom número, onde m é o número de linhas do problema. Essa escolha pode ser explicada pelo fato que se houver muitos grupos, eles serão compostos de poucas colunas e o critério de esparsidade será perdido. Por outro lado, se houver um número pequeno de grupos, o método dos gradientes conjugados requererá muitas iterações para convergir, fazendo a convergência dos métodos de pontos interiores lenta. Para divisão dos grupos utiliza-se os resultados do Teorema 3.1. Após o cálculo de todos os $c_i = \frac{h_j}{h_{j+1}}$, os maiores c_i definem os

limites de cada grupo. Dentro de cada grupo, as colunas são ordenadas pela mais esparsa até a menos esparsa. Depois disso é calculada a matriz B.

A nova abordagem do precondicionador separador foi testada em alguns problemas de grande porte. A Tabela 1 mostra os dados dos problemas. Na primeira coluna está o nome do problema; as colunas seguintes mostram a biblioteca, o número de linhas, colunas e o número de elementos não nulos, respectivamente.

Tabela 1: Dados dos problemas

—**Problema**	**Biblioteca**	**Linhas**	**Colunas**	**Elementos não nulos**
Ken-11	KENNINGTON	11548	18203	43161
Ken-13	KENNINGTON	23393	37420	84909
Ken-18	KENNINGTON	105128	154699	512719
els19	QAP	4350	13186	50882
chr25a	QAP	8149	15325	53725
scr15	QAP	2234	6210	24060
scr20	QAP	5079	15980	61780
scsd8-2b-64	STOCHLP	5130	35910	112770
scsd8-2c-64	STOCHLP	5130	35910	112770
scsd8-2r-432	STOCHLP	8650	60550	190210
Pds-30	MISC	49144	157845	338852
Pds-40	MISC	64265	214385	457538
qap12	NETLIB	2794	8856	33528
qap15	NETLIB	5698	22275	85470
nug08	MISC	742	1632	5936
nug12	MISC	2794	8856	33528
nug15	MISC	5698	22275	85470

Uma comparação das soluções dos problemas utilizando o precondicionador separador antes e após a modificação é apresentada na Tabela 2. A primeira coluna apresenta o nome do problema. As duas colunas seguintes têm o total de iterações dos MPI e o tempo computacional, em segundos, utilizando o precondicionador separador na versão original. As colunas quatro e cinco, mostram o total de iterações dos MPI e o tempo computacional, em segundos, utilizando o precondicionador separador na versão modificada. A última coluna apresenta a razão da melhora do tempo computacional quando comparadas as duas versões. O precondicionador separador foi utilizado em todas as iterações dos MPI.

Observe que o total de iterações dos MPI é alterado em alguns problemas. No entanto, o tempo computacional foi reduzido em todos os casos.

5 Conclusões

Este trabalho retomou a linha das contribuições de [12], que propuseram uma nova classe de precondicionadores para solução de sistemas de equações normais por métodos iterativos.

A nova abordagem propõe a separação das colunas da matriz precondicionadora em grupos e adotando a esparsidade como critério de reordenamento das colunas, obtém-se o

DOI: 10.5540/03.2018.006.01.0457

Tabela 2: Comparação da performance dos MPI utilizando as diferentes versões do precondicionador separador.

	Versão Original		Versão Modificada		
Problema	MPI	tempo(s)	MPI	tempo(s)	Razão das diferenças nos tempos
Ken-11	22	26.28	22	24.56	93.46
Ken-13	28	180.15	28	122.09	67.77
Ken-18	35	2168.39	35	1426.70	65.80
els19	31	120.07	31	97.25	80.99
chr25a	28	32.54	28	30.08	92.44
scr15	**24**	**16.09**	**29**	**13.85**	86.08
scr20	**22**	**193.42**	**24**	**15.83**	8.18
scsd8-2b-64	7	3.98	7	2.90	72.86
scsd8-2c-64	7	3.09	7	2.45	79.29
scsd8-2r-432	**18**	**22.13**	**9**	**9.49**	42.88
Pds-30	72	4786.30	72	3563.12	74.44
Pds-40	77	10575.72	77	8332.40	78.79
qap12	21	373.76	21	347.43	92.96
qap15	**23**	**5310.72**	**24**	**5183.37**	97.60
nug08	9	1.21	9	1.20	99.17
nug12	20	369.00	20	317.11	85.94
nug15	**23**	**5271.29**	**26**	**4146.33**	78.66

precondicionador separador o mais esparso possível.

Resultados teóricos mostram que, utilizando a partição em grupos, o número de condição da matriz precondicionada é limitado, independente do critério de reordenamento adotado. Experimentos computacionais mostram que para problemas de grande porte, o tempo computacional é reduzido. Com esses resultados é possível afirmar que a abordagem é especialmente atraente para solução de problemas de grande porte.

Investigações adicionais à esse trabalho visam estudos sobre uma abordagem precondicionadora híbrida em duas fases. Na primeira fase, um precondicionador que apresente bom desempenho nas iterações iniciais dos MPI é utilizado. Quando esse precondicionador perde sua eficiência a segunda fase é ativada e o precondicionador separador na sua versão modificada é utilizado até o final das iterações dos MPI.

Referências

[1] G. Al-Jeiroudi, J. Gondzio, and J. Hall. Preconditioning indefinite systems in interior point methods for large scale linear optimisation. *Optimization Methods & Software*, 23(3):345–363, 2008.

[2] L. Bergamaschi, J. Gondzio, M. Venturin, and G. Zilli. Inexact constraint preconditioners for linear systems arising in interior point methods. *Computational Optimization and Applications*, 36(1–2):137–147, 2007.

[3] S. Bocanegra, F. Campos, and A. Oliveira. Using a hybrid preconditioner for solving large-scale linear systems arising from interior point methods. *Special issue of Computational Optimization and Applications*, 36(2/3):149–164, 2007.

[4] L. Casacio. *Aperfeiçoamento de precondicionadores para a solução de sistemas lineares dos métodos de pontos interiores*. PhD thesis, Universidade Estadual de Campinas, Campinas – SP, Fevereiro, 2015.

[5] J. Czyzyk, S. Mehrotra, M. Wagner, and S. J. Wright. PCx an interior point code for linear programming. *Optimization Methods & Software*, 11-2(1-4):397–430, 1999.

[6] H. S. Dollar, N. I. M. Gould, W. H. A. Schilders, and A. J. Wathen. Using constraint preconditioners with regularized saddle-point problems. *Special issue of Computational Optimization and Applications*, 36(2/3):249–270, 2007.

[7] M. D. Dražić, R. P. Lazović, and V. V. Kovačević-Vujčić. Sparsity preserving preconditioners for linear systems in interior-point methods. *Computational Optimization and Applications*, 61 (3):557–570, 2015. DOI: 10.1007/s10589-015-9735-7.

[8] C. T. L. S. Ghidini, A. R. L. Oliveira, and D. C. Sorensen. Computing a hybrid preconditioner approach to solve the linear systems arising from interior point methods for linear programming using the gradient conjugate method. *Annals of Management Science*, 3:43–64, 2014.

[9] G. H. Golub and C. F. Van Loan. *Matrix Computations Third Edition*. The Johns Hopkins University Press, Baltimore, Maryland, 1996.

[10] J. Gondzio. Interior point methods 25 years later. *European Journal of Operational Research*, 218:587–601, 2012.

[11] R. D. Monteiro, J. W. O'Neil, and T. Tsuchiya. Uniform boundedness of a preconditioned normal matrix used in interior point methods. *SIAM Journal of Optimization*, 15(1):96–100, 2005. http://dx.doi.org/10.1137/S1052623403426398.

[12] A. R. L. Oliveira and D. C. Sorensen. A new class of preconditioners for large-scale linear systems from interior point methods for linear programming. *Linear Algebra and Its Applications*, 394:1–24, 2005.

[13] M. I. Velazco, A. R. L. Oliveira, and F. F. Campos. A note on hybrid preconditions for large scale normal equations arising from interior-point methods. *Optimization Methods and Software*, 25:321–332, 2010.

