

Proceeding Series of the Brazilian Society of Computational and Applied Mathematics

Uma base esparsa para o preconditionador Separador no método dos Pontos Interiores

Cecilia Orellana Castro¹

Departamento de Matemática Aplicada, IMECC-UNICAMP, Campinas, SP

Aurelio Ribeiro L. Oliveira²

Departamento de Matemática Aplicada, IMECC-UNICAMP, Campinas, SP

Resumo O preconditionador Separador (PS) foi desenvolvido especialmente para reduzir o mal condicionamento dos sistemas lineares oriundos das últimas iterações dos métodos de Pontos Interiores (MPI). Ele precisa de uma base que é uma submatriz não singular da matriz de restrições do problema, esta base depende fortemente da iteração corrente do MPI pois induz uma ordenação das colunas da matriz de restrições que pode ser aproveitada para melhorar o desempenho deste preconditionador. Propõe-se um novo critério para a escolha da base amparado num resultado que mostra que o número de condição é uniformemente limitado por uma quantidade que independe da iteração do MPI. Por outro lado, uma base esparsa economiza memória no cálculo do PS, o que implica em menor tempo computacional usado para detectar colunas linearmente independentes. Assim, nesta abordagem procurou-se um PS com base esparsa sem deixar de lado o bom condicionamento. A implementação desta nova abordagem mostrou resultados competitivos.

Palavras-chave. Método de Pontos Interiores, Preconditionador Separador, Número de condição limitado.

1 Introdução

Os MPIS do tipo primal-dual tornaram-se uma importante ferramenta para resolver problemas de programação linear (PL) de grande porte devido a suas interessantes propriedades teóricas e computacionais, destacando entre estas, sua complexidade polinomial e o fato de que não é necessário um ponto inicial factível pois tanto a factibilidade quanto a otimalidade são obtidas progressivamente.

O maior esforço computacional do MPI é o cálculo da direção de busca que é obtida de sistemas lineares que, embora apresentem um alto grau de esparsidade, nas últimas iterações do MPI tornam-se muito mal condicionados pois a condição de complementariedade deve ser satisfeita. Atualmente a direção de busca é alternativamente obtida da solução de dois sistemas; o sistema Aumentado e o sistema de Equações Normais. Foi

¹o.cecilita@hotmail.com

²aurelio@ime.unicamp.br

provado em [5] que o número de condição do sistema de Equações Normais sem preconditionamento é da ordem $\mathcal{O}(\mu^{-2})$, onde μ denota o gap de dualidade do problema de PL. Assim, o estudo de implementações eficientes que acelerem as últimas iterações do MPI é uma interessante linha de pesquisa.

Para evitar armazenamento excessivo e cálculos dificultem a resolução de problemas de grande porte usam-se métodos iterativos adequadamente preconditionados pois desta maneira apenas produtos do tipo *matriz-vetor* serão necessários. Além disso, um preconditionador esparso e que diminua significativamente o número de condição determinará a eficiência do método, por esta razão, o preconditionamento é feito em duas fases: na primeira fase, usa-se o preconditionador Fatoração Controlada de Cholesky (FCC) veja [2]; na segunda fase, o PS proposto em [7]. Os autores do PS e, posteriormente os seus colaboradores [4] fizeram ordenações baseadas em heurísticas, algumas delas bem sucedidas, porém ainda existem problemas não resolvidos e outros cuja solução demanda muito tempo, isto acontece porque a escolha das colunas linearmente independentes da base passa por uma fatoração LU cara ou porque a base escolhida não fornece um preconditionador que diminua consideravelmente o número de condição.

O objetivo deste trabalho é fazer um estudo do número de condição do sistema de Equações Normais preconditionado pelo PS e, a partir desta informação ordenar as colunas da matriz de restrições do problema para construir uma base esparsa que forneça um número de condição limitado. O resultado teórico dos autores apresentado no Teorema 4.1 está baseado em [6]. Além disso, acrescentou-se uma condição à mudança de fase proposta em [4]. Foi feita uma implementação desta nova abordagem combinando com o novo critério de mudança de fase para ser comparada com a versão atualmente utilizada, veja [4]. Os testes computacionais mostraram que a nova proposta foi mais eficiente e robusta.

2 Direção de busca no método de Pontos Interiores

Neste trabalho considera-se o par primal-dual do problema de PL canalizado dado por:

$$\begin{aligned} \text{(P)} \quad & \left\{ \min c^T x \quad \text{s. a} \quad Ax = b, \quad x + s = u \quad x, s \geq 0 \right\} \text{ e} \\ \text{(D)} \quad & \left\{ \max b^T y - u^T w \quad \text{s. a} \quad A^T y - w + z = c \quad w, z \geq 0 \quad y \in \mathbb{R}^m \right\} \end{aligned}$$

Onde $x, s, w \in \mathbb{R}^n$ e A é uma matriz de tamanho $m \times n$ que será considerada de posto completo. A direção de busca numa iteração do MPI primal-dual é encontrada aplicando o método de Newton nas condições de otimalidade do seguinte problema:

$$\text{(P')} \quad \left\{ \min c^T x - \mu \sum_{i=1}^n \log x_i - \mu \sum_{i=1}^n \log s_i \quad \text{s. a} \quad Ax = b, \quad x + s = u \quad x, s > 0 \right\}.$$

O problema (P') é obtido aplicando a penalidade barreira logarítmica nas restrições de não negatividade de (P). Assim, se $r_b = b - Ax$, $r_u = u - x - s$, $r_c = c + w - z - A^T y$, $r_1 = \mu e - XZe$, $r_2 = \mu e - SWe$ $X = \text{diag}(x_1, \dots, x_n)$, $Z = \text{diag}(z_1, \dots, z_n)$, $S = \text{diag}(s_1, \dots, s_n)$, $W = \text{diag}(w_1, \dots, w_n)$ e $e^T = (1, \dots, 1) \in \mathbb{R}^n$, a direção de busca $\Delta X = (\Delta x, \Delta s, \Delta y, \Delta z, \Delta w)^T$ é o vetor solução de (1)

$$\begin{pmatrix} A & 0 & 0 & 0 & 0 \\ I_n & I_n & 0 & 0 & 0 \\ 0 & 0 & A^T & -I_n & I_n \\ Z & 0 & 0 & X & 0 \\ 0 & W & 0 & 0 & S \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta s \\ \Delta y \\ \Delta z \\ \Delta w \end{pmatrix} = \begin{pmatrix} r_b \\ r_u \\ r_c \\ r_1 \\ r_2 \end{pmatrix}, \quad (1)$$

Se $D^{-1} = X^{-1}Z + S^{-1}W$, $r = r_c + X^{-1}r_1 - S^{-1}r_2 + S^{-1}Wr_u$ e $h = r_b$, o sistema (1) é reduzido à formulação conhecida como Sistema de Equações Normais (2) cuja matriz é simétrica e definida positiva de tamanho m , dada por:

$$ADA^T \Delta y = h + ADr. \quad (2)$$

A direção de busca é obtida pelo método dos Gradientes Conjugados Precondicionado (GCP) pois trabalhamos com problemas de grande porte.

3 Precondicionador Separador aplicado à matriz do Sistema de Equações Normais

A construção do PS está baseada na condição de complementariedade de um problema de PL, neste caso: $x_i z_i = 0$ e $s_i w_i = 0$ para todo $i = 1, \dots, n$. Observe que as componentes da matriz diagonal D que aparece em (2) são dadas por $d_i = (z_i x_i^{-1} + w_i s_i^{-1})^{-1}$, isto implica que em cada iteração do MPI a matriz D se altera, particularmente, próximo à uma solução ótima, pela não negatividade das variáveis, existirão índices $j \in \{1, \dots, n\}$ tais que $d_j \rightarrow 0$ ou $d_j \rightarrow \infty$. Esta característica é justificativa do sucesso do precondicionador nas últimas iterações do MPI e a motivação da condição de mudança de fase acrescentada neste trabalho. Em cada iteração do MPI considere uma ordenação $d_{\sigma(1)} \geq \dots \geq d_{\sigma(m)} \geq \dots \geq d_{\sigma(n)}$ onde σ é uma permutação de $\{1, \dots, n\}$, esta ordem se altera de iteração a iteração. Denote os conjuntos $\mathcal{B} = \{\sigma(1), \dots, \sigma(m)\}$ e $\mathcal{N} = \{\sigma(m+1), \dots, \sigma(n)\}$, se as colunas das matrizes A e D são reordenadas de acordo com σ , então a matriz de (2) fica:

$$ADA^T = A_{\mathcal{B}} D_{\mathcal{B}} A_{\mathcal{B}}^T + A_{\mathcal{N}} D_{\mathcal{N}} A_{\mathcal{N}}^T. \quad (3)$$

Se a submatriz $A_{\mathcal{B}}$ fosse não singular, o PS para as Equações Normais é dado pela matriz $P = D_{\mathcal{B}}^{-1/2} A_{\mathcal{B}}^{-1}$, e a matriz (3) precondicionada por P é $P(ADA^T)P^T = I_m + WW^T$ com $W = D_{\mathcal{B}}^{-1/2} A_{\mathcal{B}}^{-1} A_{\mathcal{N}} D_{\mathcal{N}}^{1/2}$. A situação ideal acontece quando $D_{\mathcal{B}}^{-1/2} \rightarrow 0$ e $D_{\mathcal{N}}^{1/2} \rightarrow 0$ implicando que $W \rightarrow 0$ e, portanto, $P(ADA^T)P^T \approx I_m$. Porém nada garante que $A_{\mathcal{B}}$ seja não singular e mesmo sendo assim, nem todo d_j com $j \in \mathcal{B}$ é um valor grande. De fato, próximo a uma solução ótima existem pelo menos $n - m$ valores próximos de zero, isto implica que no máximo existirão m valores não pequenos. Por outro lado, para fazer uma análise espectral da matriz precondicionada suponha que (λ, v) seja um autovetor de $I + WW^T$, isto é, $v + WW^T v = \lambda v$. Multiplicando por v^T observa-se que $|\lambda| \geq 1$, ou seja $\kappa(P(ADA^T)P^T) = \frac{\lambda_{\max}}{\lambda_{\min}} \leq \lambda_{\max}$. Observe que,

$$\lambda_{\max}(PAD^T P^T) = \|PAD^{1/2}\|_2^2 \leq \|PAD^{1/2}\|_F^2 = \sum_{i=1}^n d_j \|PA_j\|_2^2, \quad (4)$$

assim, usaremos (4) para encontrar um limitante do número de condição $\kappa(P(ADA^T)P^T)$. Por outro lado, encontrar colunas linearmente independentes de A para formar a base do PS pode demandar muito uso tempo, pois este trabalho é feito usando uma fatoração LU da matriz A , um pivô nulo ou próximo de zero indicará que a coluna correspondente a este pivô é linearmente dependente. Quando o preenchimento é excessivo a fatoração LU é reiniciada salvando as colunas linearmente independentes encontradas.

4 A nova proposta

A partir das observações apresentadas na seção anterior, surge a ideia de criar uma ordenação que considere simultaneamente bom condicionamento e esparsidade para a base do PS. Para isso denote como $nnz(A_j)$ = número de elementos não nulos na coluna A_j , para $j = 1, \dots, n$, observe que; $1 \leq nnz(A_j) \leq m$ para toda coluna A_j de A , porém em problemas esparsos $nnz(A_j) \ll m$. Defina $k_j = d_j^{1/2}/nnz(A_j)$ e faça uma ordenação decrescente dos k_j . Com essa ordem propõe-se o seguinte algoritmo.

Entrada A matriz de restrições $A \in \mathbb{R}^{m \times n}$ de posto m e a matriz diagonal D .

Saída Os conjuntos de índices básicos $\mathcal{B} = \{b_1, \dots, b_m\}$ e não básicos $\mathcal{N} = \{1, \dots, n\} \setminus \mathcal{B}$.

1. Obter a permutação σ do conjunto $\{1, \dots, n\}$ tal que: $k_{\sigma(1)} \geq k_{\sigma(2)} \geq \dots \geq k_{\sigma(n)}$;

2. Defina $\mathcal{B} = \emptyset$, $i = 1$, $k = 0$;

3. **Enquanto** $|\mathcal{B}| < m$

Se $A_{\sigma(i)}$ é linearmente independente a $\{A_j : j \in \mathcal{B}\}$

então $\mathcal{B} = \mathcal{B} \cup \{\sigma(i)\}$; $k = k + 1$; $b_k = \sigma(i)$;

Fim-se $i = i + 1$.

Fim-enquanto

A ordenação decrescente dos k_j 's é motivada pela seguinte razão: se duas colunas A_{j_1} e A_{j_2} satisfazem $nnz(A_{j_1}) \leq nnz(A_{j_2})$, ou seja A_{j_1} é mais esparsa que A_{j_2} , então, $1/nnz(A_{j_1}) \geq 1/nnz(A_{j_2})$, logo a coluna A_{j_1} terá prioridade sobre A_{j_2} se $d_{j_1} \geq d_{j_2}$. Assim, enquanto os valores $d_j^{1/2}$ serão usados no Teorema 4.1 buscando um bom condicionamento, as quantidades $nnz(A_j)$ procuram dar prioridade às colunas mais esparsas. O algoritmo acima e a prova do Teorema 4.1 estão baseados em [6] acrescentando uma condição que leve em conta a esparsidade da matriz A . Para simplificar a notação considera-se a permutação σ como sendo a permutação identidade, além disso, denota-se $A_{\mathcal{B}}$ simplesmente por B .

Teorema 4.1. *Suponha que os índices básicos \mathcal{B} e não básicos \mathcal{N} do PS sejam obtidos pelo algoritmo acima. Então:*

$$1. d_j^{1/2} \|D_{\mathcal{B}}^{-1/2} B^{-1} A_j\| = 1 \quad \text{para } j \in \mathcal{B};$$

$$2. d_j^{1/2} \|D_{\mathcal{B}}^{-1/2} B^{-1} A_j\| \leq nnz(A_j) \|B^{-1} A_j\| \quad \text{para } j \in \{1, \dots, n\} \setminus \mathcal{B}$$

Além disso, se $K = \max\{nnz(A_j) : j = 1, \dots, n\}$, então: $\kappa(PADA^T P^T) \leq nK^2 \|B^{-1} A\|^2$.

Prova Consideram-se duas situações:

Caso 1. Se $j \in \mathcal{B}$, então $B^{-1}A_j = e_j$ onde e_j é o j -ésimo vetor canônico de \mathbb{R}^m ; logo, $d_j^{1/2} \|D_B^{-1/2} B^{-1}A_j\| = d_j^{1/2} \|D_B^{-1/2} e_j\| = d_j^{1/2} \|d_j^{-1/2} e_j\| = 1$.

Caso 2. Se $j \notin \mathcal{B}$ são considerados dois casos:

Caso 2.1. A coluna A_j não foi considerada para entrar na base de acordo ao algoritmo acima, isto é, $j > b_i$, logo $k_{b_i} \geq k_j$ para todo $b_i \in \mathcal{B}$. Seja $d_0^{1/2} = \min\{d_{b_i}^{1/2} : b_i \in \mathcal{B}\}$, então se $k_0 = d_0^{1/2} / \text{nnz}(A_0)$ tem-se que $k_0 \geq k_{b_m} \geq k_j$; logo:

$$d_j^{1/2} \|D_B^{-1/2} B^{-1}A_j\| \leq \frac{d_j^{1/2} \|B^{-1}A_j\|}{\min\{d_{b_i}^{1/2} : b_i \in \mathcal{B}\}} = \frac{k_j \text{nnz}(A_j)}{k_0 \text{nnz}(A_0)} \|B^{-1}A_j\| \leq \text{nnz}(A_j) \|B^{-1}A_j\|. \quad (5)$$

Caso 2.2. A coluna A_j foi candidata para ser r -ésima coluna de B , porém A_j resultou ser linearmente dependente às colunas $A_{b_1}, A_{b_2}, \dots, A_{b_{r-1}}$; isto é, $A_j = B[u, 0]^T$, para $u \in \mathbb{R}^{r-1}$. Observe que $k_{b_i} \geq k_j$ para $i = 1, \dots, r-1$, além disso $\|u\| = \|B^{-1}A_j\|$.

Se $d_0^{1/2} = \min\{d_{b_1}^{1/2}, \dots, d_{b_{r-1}}^{1/2}\}$ e $k_0 = d_0^{1/2} / \text{nnz}(A_0)$ então $k_0 \geq k_{b_{r-1}} \geq k_j$, logo:

$$d_j^{1/2} \|D_B^{-1/2} B^{-1}A_j\| = d_j^{1/2} \left(\sum_{i=1}^{r-1} d_{b_i}^{-1} u_i^2 \right)^{1/2} \leq \frac{k_j \text{nnz}(A_j)}{k_0 \text{nnz}(A_0)} \|B^{-1}A_j\| \leq \text{nnz}(A_j) \|B^{-1}A_j\|. \quad (6)$$

Usando (4) tem-se que $\lambda_{\max} = \|D_B^{-1/2} B^{-1}AD^{1/2}\|_2^2 \leq \sum_{j=1}^n d_j \|D_B^{-1/2} B^{-1}A_j\|_2^2$, além disso, usando as desigualdades obtidas em (5) e (6) obtem-se:

$$\lambda_{\max} (PADA^T P^T) \leq K^2 \sum_{j=1}^n \|B^{-1}A_j\|^2 = K^2 \|B^{-1}A\|_F^2 \leq mK^2 \|B^{-1}A\|^2.$$

Por outro lado, $\lambda_{\min} (PADA^T P^T) \geq 1$, assim $\kappa(PADA^T P^T) \leq mK^2 \|B^{-1}A\|^2$. ■

Observe que o limitante do número de condição da matriz preconditionada depende apenas dos dados do problema e não da iteração do MPI.

5 Novo critério de Mudança de Fase

A mudança de fase para o preconditionamento híbrido proposta em [4] visa aproveitar o preconditionador FCC durante a maior quantidade de iterações do MPI possível. Na construção do FCC o preenchimento em cada coluna do preconditionador é controlado por um parâmetro η , onde η é o número de entradas extras por coluna. A mudança de fase é feita quando o valor de η supera um limite permitido dado por um valor η_{\max} . Observou-se que em vários problemas o PS já obtém bom desempenho antes de atingir o $\eta_{\max} = 10$, assim foi acrescentada uma condição adicional na mudança de fase que melhorou consideravelmente o tempo computacional em boa parte dos problemas testados. A condição acrescentada considera uma ordenação decrescente dos elementos da diagonal: $d_1 \geq d_2 \geq \dots \geq d_m \geq \dots \geq d_n$, e define $\delta = s_1 - s_2$, onde $s_1 = \sum_{i=1}^m d_i$ e $s_2 = \sum_{i=m+1}^n d_i$. A mudança de fase proposta é a seguinte:

Se o número de iterações do método GCP numa iteração do MPI é maior que $m/5$ então deve-se verificar se $\eta \leq \eta_{\max}$ e $\delta < 1.5 \times 10^6$, ou seja, verifica-se que o preconditionador FCC ainda não requer muita memória e o PS ainda não deverá obter um bom desempenho

pois a matriz D não apresenta diferenças abruptas nos elementos da sua diagonal. Se a condição for verdadeira, η é aumentado em 10. Caso contrário, isto é, se $\eta > \eta_{\max}$ ou $\delta \geq 1.5 \times 10^6$, a mudança de fase é realizada.

6 Resultados Numéricos

Tabela 1: Número de iterações , Tempo

Prob	Iterações		Tempo		Tamanho	
	PS_M	PS_C	PS_M	PS_C	Linha	Coluna
cre-a	27	27	7,36	6,34	2989	6692
cre-b	43	43	49,50	37,65	5328	36382
cre-c	27	27	5,78	5,44	2370	5412
cre-d	42	42	33,36	31,27	4094	28601
ex05	39	39	6,49	6,52	832	7805
ex09	45	50	58,23	64,91	1821	18184
ken11	23	22	10,31	10,08	9964	16740
ken13	29	29	103,22	59,13	22365	36561
ken18	41	40	1246,82	844,12	78538	128434
chr22b	29	29	19,84	21,75	5587	10417
chr25a	29	29	45,68	48,01	8149	15325
scr15	24	24	8,12	7,45	2234	6210
scr20	21	21	66,54	66,62	5079	15980
rou20	24	24	915,36	703,08	7359	37640
nug12	–	20	–	71,56	3192	8856
qap8	10	10	0,78	0,75	742	1632
qap12	–	20	–	82,94	2794	8856
ste36a	37	37	10178,34	9443,21	27683	131076
25fv47	29	28	1,90	2,10	788	1843
maros	40	20	2,44	0,79	655	1437
nesm	31	31	1,69	0,80	654	2922
BL	38	38	19,11	15,72	5729	12462
NL	41	41	35,87	32,09	6665	14680
stocfor2	21	21	1,21	0,89	1980	2868
stocfor3	32	32	91,56	84,12	15362	22228

Os experimentos numéricos foram realizados utilizando a versão modificada do PCx, veja [3], nesta versão, o método direto usado para a solução dos sistemas lineares foi substituído pelo método dos GCP com uma abordagem híbrida proposta em [1]. Os testes realizados comparam os resultados das abordagens PS_M e PS_C , sendo PS_M a abordagem com ordenação da base B proposta em [4], e PS_C a abordagem com base B obtida pelo algoritmo da seção 4. Na abordagem PS_M usou-se a troca de fase proposta em [4] e na abordagem PS_C usou-se a troca de fase da seção 5. A base B do PS muda quando $8 * n_g \geq m$, onde n_g denota o número de iterações do método dos GCP numa iteração do MPI. Os problemas utilizados são de domínio público extraídos das bibliotecas NETLIB, QAP e KENNINGTON. As últimas colunas da tabela indicam o número de linhas e de colunas dos problemas pré-processados pelo PCx. Para avaliar nossa proposta comparamos

o número total de iterações do MPI e o tempo em que cada problema foi resolvido.

Foram testados 25 problemas. As diferenças mais notáveis foram marcadas em negrito. O símbolo – indica que o problema não foi resolvido. Observa-se que em 5 problemas o número de iterações do MPI foi reduzido, de fato, os problemas **nug12** e **qap12** não foram resolvidos pela abordagem PS_M . Já no problema **ex-09** o número de iterações com a abordagem PS_C aumentou. Com respeito ao tempo usado para resolver os problemas, a abordagem PS_C obteve melhor desempenho em 14 problemas e desempenho inferior em apenas em 3.

7 Conclusões

Os testes numéricos mostram que a nova proposta é mais eficiente e robusta devido à nova mudança de fase. Além disso o cálculo da base foi acelerado porque o número de colunas linearmente independentes encontradas antes do primeiro reinício da fatoração LU foi superior na abordagem PS_C .

Agradecimentos

Este trabalho contou com o apoio financeiro da FAPESP e do CNPq.

Referências

- [1] S. Bocanegra, F. F. Campos, and A. R. L. Oliveira. Using a hybrid preconditioner for solving large-scale linear systems arising from interior point methods. *Computational Optimization and Applications*, 36(2-3):149–164, 2007.
- [2] F. F. Campos. *Analysis of conjugate gradients-type methods for solving linear equations*. PhD thesis, University of Oxford, 1995.
- [3] J. Czyzyk, S. Mehrotra, M. Wagner, and S. J. Wright. Pcx: an interior-point code for linear programming. *Optimization Methods and Software*, 11(1):397–430, 1999.
- [4] M. I. V. Fontova, A. R. L. Oliveira, and F. F. Campos. A note on hybrid preconditioners for large-scale normal equations arising from interior-point methods. *Optimization Methods and Software*, 25(2):321–332, 2010.
- [5] J. Gondzio. Matrix-free interior point method. *Computacional Optimization and Applications*, 51:457–480, 2012.
- [6] R. D. C. Monteiro, J. W. O’Neal, and T. Tsuchiya. Uniform boundedness of a preconditioned normal matrix used in interior-point methods. *SIAM Journal on Optimization*, 15(1):96–100, 2004.
- [7] 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.