Tema
Tendências em Matemática Aplicada e Computacional, **15**, N. 3 (2014), 311-322

www.scielo.br/tema
doi: 10.5540/tema.2014.015.03.0311

Variantes do Método dos Gradientes Conjugados aplicados a sistemas lineares originados dos Métodos de Pontos Interiores

A.F.E. COELHO[1*], A.R.L. OLIVEIRA[1]
e M.I.V. FONTOVA[2]

Recebido em 6 abril, 2013 / Aceito em 17 dezembro, 2014

RESUMO. Neste trabalho, comparamos duas versões precondicionadas do método dos gradientes conjugados. Essas versões diferem da versão clássica, pois consideram que o sistema linear é um sistema de equações normais. Os sistemas lineares que iremos resolver surgem do cálculo das direções dos métodos de pontos interiores. A determinação desta direção consiste no passo de maior esforço computacional e, quando trabalhamos com sistemas de grande porte, o uso de métodos diretos pode ser inviável. Portanto, uma opção é utilizar métodos iterativos precondicionados. Assim, o desempenho de duas versões do método dos gradientes conjugados precondicionado é comparado à versão clássica que já foi utilizada, neste mesmo contexto, em trabalhos anteriores. Resultados numéricos mostram que uma dessas versões é competitiva em relação à versão clássica.

Palavras-chave: métodos de pontos interiores, método dos gradientes conjugados precondicionado.

1 INTRODUÇÃO

Desde a década de 80 os métodos de pontos interiores têm atraído grande interesse. Estes métodos são muito eficientes na solução de problemas de grande porte.

Nestes métodos, a cada iteração, são resolvidos dois sistemas lineares no cálculo das direções por métodos diretos ou iterativos. Os métodos diretos apresentam dificuldades em problemas de grande porte devido ao preenchimento da matriz. Em muitos casos, a melhor solução consiste em usar métodos iterativos. A matriz dos sistemas é simétrica e definida positiva permitindo usar o método dos gradientes conjugados. Devido ao mal-condicionamento da matriz nas iterações finais dos métodos é necessário o uso de precondicionadores. Neste trabalho, utilizamos o precondicionador híbrido proposto em [5].

*Autor correspondente: Alessandro F.E. Coelho
[1] Instituto de Matemática e Computação Científica, IMECC, UNICAMP, Cidade Universitária Zeferino Vaz, Distrito de Barão Geraldo, 13081-970 Campinas, SP, Brasil. E-mails: alessandro.coelho@gmail.com; aurelio@ime.unicamp.br
[2] Faculdade Campo Limpo Paulista, FACCAMP, Rua Guatemala, 167, Bairro Jardim América, 13231-230 Campo Limpo Paulista, SP, Brasil. E-mail: marta.velazco@gmail.com

Para que o método dos gradientes conjugados seja aplicado, não é necessário que o sistema linear seja do tipo de equações normais. Entretanto, existem versões deste método que consideram a estrutura de equações normais em sua formulação [10]. Note que os sistemas originados dos métodos de pontos interiores são sistemas de equações normais.

O objetivo deste trabalho consiste em avaliar a eficiência de duas versões de gradientes conjugados precondicionado (GCP) que consideram sistemas de equações normais. Estas versões são particularizadas para os sistemas oriundos dos métodos de pontos interiores.

1.1 Método de Pontos Interiores Preditor-Corretor

Considere o problema primal e o problema dual de programação linear na forma padrão [12]:

$$\begin{array}{llll} \text{minimizar} & c^t x, & \text{sujeito a} & Ax = b, x \geq 0, \quad \text{problema primal,} \\ \text{maximizar} & b^t y, & \text{sujeito a} & A^t y + z = c, z \geq 0, \quad \text{problema dual,} \end{array}$$

onde A é uma matriz $m \times n$ de posto completo, c e b são vetores coluna de dimensões apropriadas, x é o vetor $n \times 1$ de variáveis primais, y é o vetor $m \times 1$ de variáveis duais e z é o vetor $n \times 1$ de variáveis de folga duais.

O método preditor-corretor [12] calcula duas direções a cada iteração para as variáveis (x, y, z) de programação linear. De forma resumida, podemos dizer que essas direções são obtidas aplicando o método de Newton nas condições de otimalidade:

$$\begin{pmatrix} Ax - b \\ A^t y + z - c \\ XZe \end{pmatrix} = 0; \qquad (x, z) \geq 0.$$

Definindo $r_p = b - Ax$, $r_d = c - A^t y - z$ e $r_a = -XZe$, desejamos que estes resíduos (r_p, r_d, r_a) sejam nulos obtendo assim uma solução ótima.

2 MÉTODO DOS GRADIENTES CONJUGADOS PARA SOLUÇÃO DE SISTEMAS DE EQUAÇÕES NORMAIS E MÉTODOS DE PONTOS INTERIORES

Na solução de sistemas com matrizes simétricas e definida positiva é utilizado o método dos gradientes conjugados. Quando trabalhamos com matrizes mal-condicionadas é necessário o uso de precondicionadores para garantir a convergência do método. Em [10] são apresentadas duas versões do método dos gradientes conjugados precondicionado que consideram a estrutura de equações normais em sua formulação. Nesta seção, as mesmas serão particularizadas para as matrizes originadas dos métodos de pontos interiores.

Dado um sistema linear $Gx = d$, existem duas alternativas para se obter um sistema de equações normais:

1. Substituindo a variável x por $G^t u$ teríamos o sistema $GG^t u = d$;
2. Premultiplicando o sistema original por G^t resultando no sistema $G^t Gx = G^t d$.

A primeira versão, Gradientes Conjugados Precondicionado – Normal Error (GCPNE), é construída precondicionando o sistema $GG^t u = d$, com $x = G^t u$. A segunda versão, Gradientes Conjugados Precondicionado – Normal Residual (GCPNR), considera o sistema o $G^t Gx = G^t d$.

2.1 Métodos de Pontos Interiores

O passo mais caro de uma iteração de pontos interiores consiste na solução do sistema linear:

$$(ADA^t)dy = r_p + A(Dr_d - Z^{-1}r_a), \tag{2.1}$$

onde $D = Z^{-1}X$ é uma matriz diagonal definida positiva.

Para resolver o sistema (2.1), considere o sistema de equações normais $GG^t u = d$. Relacionando ambos os sistemas, temos as seguintes identidades: o vetor u representa as direções dy a serem calculadas, a matriz G representa a matriz $AD^{\frac{1}{2}}$ de forma que $GG^t = ADA^t$. Finalmente, o vetor d representa o lado direito do sistema linear $r_p + A(Dr_d - Z^{-1}r_a)$.

Desta forma, o método *GCPNE* para resolver o sistema (2.1) pode ser resumido como:

Algoritmo 2.1. *GCPNE na variável u – Versão para o Sistema do Método de Pontos Interiores*

Calcule $r_0 = d - ADA^t u_0$, $z_0 = M^{-1}r_0$, $p_0 = z_0$

Para $j = 0, 1, \ldots,$ *até convergir, faça:*

$$w_j = D^{\frac{1}{2}}A^t p_j$$

$$\alpha_j = \frac{\langle r_j, z_j \rangle}{\langle w_j, w_j \rangle}$$

$$u_{j+1} = u_j + \alpha_j p_j$$

$$r_{j+1} = r_j - \alpha_j AD^{\frac{1}{2}} w_j$$

$$z_{j+1} = M^{-1}r_{j+1}$$

$$\beta_j = \frac{\langle r_{j+1}, z_{j+1} \rangle}{\langle r_j, z_j \rangle}$$

$$p_{j+1} = z_{j+1} + \beta_j p_j$$

Fim

Outra forma de resolver o sistema (2.1) é compará-lo com o segundo tipo de equações normais: $G^t Gx = G^t d$. Note que a matriz G será substituída por $D^{\frac{1}{2}}A^t$ e mantemos a variável x que representa o valor de dy. O vetor $r_p + A(Dr_d - Z^{-1}r_a)$ do sistema (2.1) é representado por $G^t d$ no sistema $G^t Gx = G^t d$. Portanto, é necessário recuperar o vetor d do sistema (2.1). Logo, definindo

$$\tilde{b} = A^t(AA^t)^{-1}b,$$

temos:

$$\begin{aligned} G^t d &= AD^{\frac{1}{2}} d \\ &= r_p + A(Dr_d - Z^{-1} r_a) \\ &= b - Ax + A(Dr_d - Z^{-1} r_a) \\ &= A\tilde{b} - Ax + A(Dr_d - Z^{-1} r_a) \\ &= ADD^{-1}\tilde{b} - ADD^{-1}x + ADr_d - ADD^{-1}Z^{-1} r_a \\ &= AD(D^{-1}\tilde{b} - D^{-1}x + r_d - D^{-1}Z^{-1} r_a) \\ &= AD^{\frac{1}{2}} D^{\frac{1}{2}} (D^{-1}\tilde{b} - D^{-1}x + r_d - D^{-1}Z^{-1} r_a) \end{aligned}$$

Portanto

$$d = D^{\frac{1}{2}} (D^{-1}\tilde{b} - D^{-1}x + r_d - D^{-1}Z^{-1} r_a),$$

para o sistema (2.1).

Assim, o método *GCPNR* para resolver o sistema (2.1) pode ser resumido como:

Algoritmo 2.2. *GCPNR – Versão para o Sistema do Método de Pontos Interiores*

Calcule $r_0 = d - D^{\frac{1}{2}} A^t x_0$, $\tilde{r}_0 = AD^{\frac{1}{2}} r_0$, $z_0 = M^{-1}\tilde{r}_0$, $p_0 = z_0$.

Para $j = 0, 1, \ldots,$ *até convergir, faça:*

$$w_j = D^{\frac{1}{2}} A^t p_j$$

$$\alpha_j = \frac{\langle z_j, \tilde{r}_j \rangle}{\langle w_j, w_j \rangle}$$

$$x_{j+1} = x_j + \alpha_j p_j$$

$$r_{j+1} = r_j - \alpha_j w_j$$

$$\tilde{r}_{j+1} = AD^{\frac{1}{2}} r_{j+1}$$

$$z_{j+1} = M^{-1}\tilde{r}_{j+1}$$

$$\beta_j = \frac{\langle z_{j+1}, \tilde{r}_{j+1} \rangle}{\langle z_j, \tilde{r}_j \rangle}$$

$$p_{j+1} = z_{j+1} + \beta_j p_j$$

Fim

3 EXPERIMENTOS NUMÉRICOS

Os experimentos numéricos foram realizados utilizando o código PCx [7]. O PCx resolve problemas de otimização linear pelo método preditor-corretor com múltiplas correções. Este código é implementado na linguagem *C* exceto as rotinas para solução dos sistemas lineares desenvolvida

na linguagem *Fortran*. Os sistemas lineares que surgem ao decorrer do método são resolvidos através de uma abordagem direta por fatoração de Cholesky.

Neste trabalho, foi utilizada a versão PCx-Modificado [9, 5, 11]. Esta versão utiliza uma abordagem iterativa na solução dos sistemas lineares. Assim, a opção de múltiplas correções é desligada e a fatoração de Cholesky é substituída pela solução iterativa através do método dos gradientes conjugados precondicionado. O precondicionador utilizado é o Precondicionador Híbrido [5] com as modificações propostas em [11]. A partir desta versão modificada, as novas versões dos Gradientes Conjugados para sistemas lineares são testadas e comparadas com a mesma.

Todos os testes foram realizados em um processador Intel Core 2 Duo 2.2GHz com 2Gb de RAM, em ambiente Linux.

Os problemas testes são apresentados na Tabela 1. Estes foram selecionados dentro das classes de problemas onde a decomposição de Cholesky produz uma matriz mais densa que a matriz original [5, 9]. Os problemas foram extraídos das bibliotecas de domínio público: NETLIB [3], STOCHPL [4], MISC [1, 2] e QAPLIB [6]. Na tabela, para cada problema é descrito: o número de linhas, colunas, elementos não nulos após o pré-processamento, a densidade da matriz e biblioteca de origem.

3.1 Resultados Computacionais

A eficiência e robustez dos métodos GCPNR e GCPNE é avaliada a partir da comparação dos mesmos com o método dos gradientes conjugados precondicionado utilizado na versão PCx-modificado. Nas comparações, foram considerado:

- Total de iterações de pontos interiores (descritas entre parênteses);
- Total de iterações dos métodos de gradientes conjugados dada pela soma do número de iterações realizadas para o cálculo de cada uma das duas direções do método de pontos interiores;
- Tempo de processamento.

A Tabela 2 apresenta os resultados obtidos para os problemas testados com a versão GCPNR e com a versão clássica de GCP. A versão GCPNR convergiu apenas para cinco dos problemas testados e obteve pior desempenho em todos os itens analisados. A versão GCPNR precisa de um esforço extra para conversão do sistema linear a um sistema de equações normais, isto é, recuperar o lado direito do sistema como descrito na Seção 2. Portanto, fazendo estas operações acumulamos muito erros numéricos visto que temos o cálculo de um sistema linear extra para determinar uma matriz inversa. Os erros numéricos dificultam a convergência do método iterativo e o mesmo é interrompido antes de resolver o sistema pois excede o número máximo de iterações. Este resultado compromete o método de pontos interiores que converge para resultados incoerentes e não para um ótimo do problema. O código PCx reconhece este estado e o programa devolve a mensagem *UNKNOWN status*.

Tabela 1: Problemas testes.

Problema	Linha	Colunas	Não Nulos	Densidade	Biblioteca
scsd8-2c-64	5130	35910	112770	0,0612	STOCHLP
scsd8-2r-432	8650	60550	190210	0,0363	STOCHLP
els19	4350	13186	50882	0,0887	QAP
chr22b	5587	10417	36520	0,0627	QAP
scr15	2234	6210	24060	0,1739	QAP
scr20	5079	15980	61780	0,0761	QAP
rou20	7359	37640	152980	0,0552	QAP
ste36a	27683	131076	512640	0,0141	QAP
ste36b	27683	131076	512640	0,0141	QAP
stocfor3	16675	23541	62960	0,0160	NETLIB
qap12	2794	8856	33528	0,1355	NETLIB
qap15	5698	22275	85470	0,0673	NETLIB
nug12	3192	8856	33528	0,1355	MISC
nug15	6330	22275	85470	0,0673	MISC
pds-10	15648	48780	103725	0,0135	MISC
pds-20	32276	106180	226494	0,006608	MISC
pds-30	49944	158489	333260	0,004210	MISC
pds-40	64265	214385	457538	0,003320	MISC
pds-50	80339	272513	581152	0,002654	MISC
pds-60	96503	332862	709178	0,002207	MISC
pds-70	114944	390005	822526	0,001834	MISC
pds-80	126109	430800	916852	0,001687	MISC
pds-90	142823	475448	1002902	0,001476	MISC
pds-100	156243	514577	1096002	0,001363	MISC

Se somarmos as iterações do método iterativo realizadas nos problemas que convergiram para ambos métodos temos os seguintes resultados: o GCP realizou 88418 iterações e o GCPNR realizou 224211 iterações. O método GCPNR fez mais que o dobro de iterações para resolver os mesmos sistemas e levou quase o triplo do tempo de processamento.

A versão GCPNE mostrou-se muito mais competitiva com o método clássico de GCP como mostra a Tabela 3. O método de pontos interiores atingiu o valor ótimo para praticamente todos os problemas testados. Este método não precisa da transformação extra pois, os sistemas oriundos do método de pontos interiores são semelhantes ao sistema de equações normais para o qual o método é definido.

Tabela 2: Resultados obtidos com os métodos GCP e GCPNR.

Problema	Iterações do GCP	Iterações do GCPNR	Tempo(s) – GCP	Tempo(s) – GCPNR
scsd8-2c-64	382 (7)	5864 (10)	4,04	20,00
scsd8-2r-432	6519 (18)	11397 (22)	39,28	180,96
els19	26294 (31)	–	168,18	–
chr22b	25580 (29)	–	71,46	–
scr15	15973 (24)	–	28,81	–
scr20	36360 (21)	–	223,70	–
rou20	54326 (24)	–	3216,66	–
ste36a	312988 (38)	–	32183,05	–
ste36b	–	–	–	–
stocfor3	69507 (32)	–	338,54	–
qap12	29962 (20)	–	285,54	–
qap15	113695 (24)	–	4844,20	–
nug12	23677 (20)	–	245,79	–
nug15	91134 (23)	–	4466,58	–
pds-10	6872 (47)	8851 (49)	63,36	77,26
pds-20	52372 (60)	–	741,67	–
pds-30	31129 (73)	70946 (79)	741,93	1748,39
pds-40	43516 (79)	127153 (84)	1463,64	4813,55
pds-50	71481 (79)	–	2905,05	–
pds-60	83095 (85)	–	4177,60	–
pds-70	83050 (84)	–	5052,97	–
pds-80	80742 (83)	–	5729,80	–
pds-90	97909 (82)	–	7467,40	–
pds-100	122237 (86)	–	9829,66	–

Os métodos GCP e GCPNE se comportam de forma muito parecida, apesar de ocorrerem diferenças. Variações no total de iterações do médodo dos gradientes conjugados ocorreram em todos os problemas e, em alguns casos, até mesmo no total de iterações de pontos interiores. Destacam-se os problemas *ste36b* e *nug15*. O primeiro foi resolvido apenas usando o método GCPNE entretanto, o segundo foi resolvido apenas pelo GCP. O método GCP realizou 1388306 iterações na resolução dos sistemas lineares enquanto que o método GCPNE realizou 1338843 iterações, somando apenas as iterações dos problemas que convergiram para ambos. Quanto ao tempo de processamento, ambos apresentaram comportamento semelhante na soma total; o GCPNE levou menos tempo na solução dos problemas. Apesar do melhor desempenho do GCPNE em relação ao método clássico a diferença não é relevante considerando a quantidade de problemas testados.

Tabela 3: Resultados obtidos com os métodos GCP e GCPNE.

Problema	Iterações do GCP	Iterações do GCPNE	Tempo(s) – GCP	Tempo(s) – GCPNE
scsd8-2c-64	382 (7)	389 (7)	4,04	4,03
scsd8-2r-432	6519 (18)	6914 (18)	39,28	42,26
els19	26294 (31)	26315 (31)	168,18	172,48
chr22b	25580 (29)	25688 (29)	71,46	72,98
scr15	15973 (24)	15886 (24)	28,81	28,99
scr20	36360 (21)	35789 (21)	223,70	254,20
rou20	54326 (24)	54606 (24)	3216,66	3156,59
ste36a	312988 (38)	305066 (38)	32183,05	32256,71
ste36b	–	375810 (37)	–	41786,72
stocfor3	69507 (32)	69828 (32)	338,54	345,72
qap12	29962 (20)	30519 (20)	285,54	305,59
qap15	113695 (24)	92876 (23)	4844,20	4160,35
nug12	23677 (20)	24111 (20)	245,79	249,23
nug15	91134 (23)	–	4466,58	–
pds-10	6872 (47)	6765 (47)	62,94	63,01
pds-20	52372 (60)	53759 (60)	741,67	782,26
pds-30	31769 (74)	31324 (74)	793,66	824,34
pds-40	43516 (79)	43797 (79)	1463,64	1537,23
pds-50	71481 (79)	69471 (79)	2905,05	2959,64
pds-60	83095 (85)	82774 (85)	4177,60	4337,98
pds-70	83050 (84)	77579 (85)	5052,97	4994,45
pds-80	80742 (83)	75626 (83)	5729,80	5658,51
pds-90	97909 (82)	82992 (81)	7467,40	6841,65
pds-100	122237 (86)	126769 (87)	9829,66	10493,45

(–) significa que o método falhou.

Estes resultados também podem ser observados através do perfil de desempenho [8]. O perfil de desempenho é uma ferramenta para comparar o desempenho de s algoritmos na solução de p problemas. Neste caso, temos $s = 3$ algoritmos e $p = 24$ problemas. Usaremos como medidas de desempenho os seguintes itens: total de iterações de pontos interiores; total de iterações dos métodos dos gradientes conjugados e tempo de processamento.

A Figura 1 apresenta o gráfico utilizando como medida de desempenho o total de iterações do método de pontos interiores. Os métodos GCP e GCPNE têm a mesma eficiência e robustez mostrando-se superiores ao GCPNR que só convergiu em *20%* dos problemas testados.

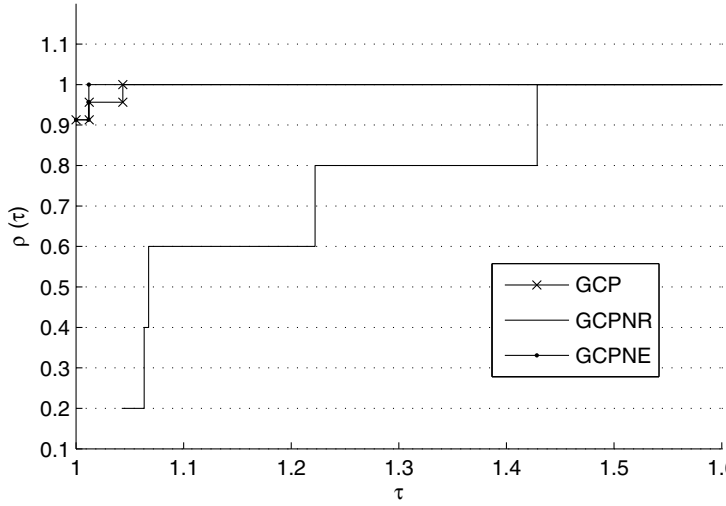

Figura 1: GCP, GCPNR e GCPNE – Perfil de desempenho: Iterações de Pontos Interiores.

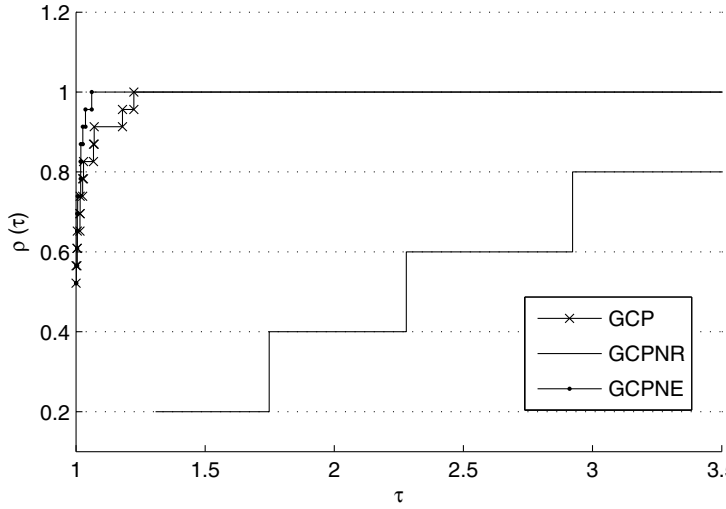

Figura 2: GCP, GCPNR e GCPNE – Perfil de desempenho: Iterações de Gradientes Conjugados.

Na Figura 2, temos os resultados a partir do total de iterações do método dos gradientes conjugados como medida de desempenho e os resultados são similares. Finalmente, a Figura 3 apresenta os mesmos resultados para o perfil de desempenho com o tempo de processamento. Podemos verificar que o método GCPNR se mostrou menos eficiente e robusto que os métodos GCP e GCPNE. Os resultados a partir do perfil de desempenho mostram que o método GCPNE é tão eficiente quanto o método clássico e seus resultados são competitivos.

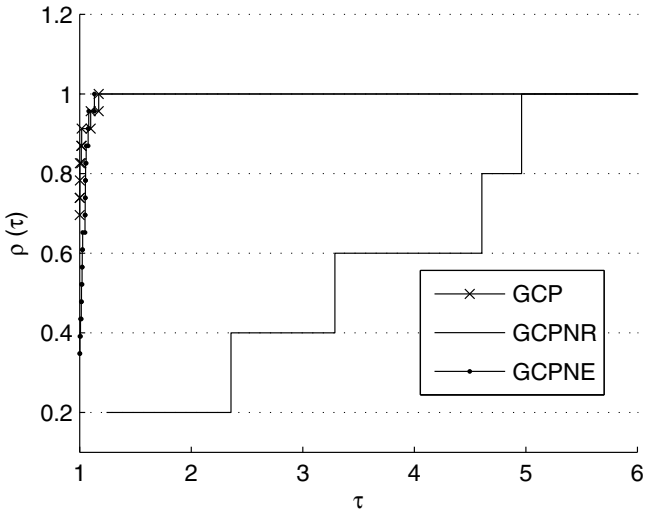

Figura 3: GCP, GCPNR e GCPNE – Perfil de desempenho: Tempo de Processamento.

4 CONCLUSÕES

Neste trabalho, avaliamos a eficiência de diferentes versões do método dos gradientes conjugados precondicionado aplicados aos sistemas lineares oriundos dos métodos de pontos interiores. A principal motivação para testar diferentes versões desses métodos vem da estrutura natural destes sistemas, na forma de equações normais.

Para avaliar a eficiência de cada uma das versões consideramos o total de iterações do método de pontos interiores e do método dos gradientes conjugados e o tempo total de processamento. Os problemas selecionados foram resolvidos utilizando a versão clássica e as duas versões que utilizam a estrutura de equações normais: GCPNE e GCPNR.

A implementação da versão GCPNE no código PCx-modificado é natural visto que a estrutura do método se adequa muito bem aos sistemas encontrados. Os resultados obtidos com essa versão foram muito semelhantes aos da versão clássica. Em poucos casos ocorreram variações no total de iterações do método de pontos interiores. De maneira geral, o tempo e total de iterações do método dos gradientes conjugados também não apresentaram grandes diferenças. Portanto, a versão GCPNE obteve bom desempenho sendo competitiva com a versão GCP.

A versão GCPNR apresentou dificuldades teóricas na sua adaptação aos sistemas lineares dos métodos de pontos interiores, pois os sistemas não eram equivalentes. O sistema de equações normais resolvido pelo GCPNR é obtido a partir da premultiplicação do sistema pela matriz transposta dos coeficientes. Para obter a equivalência entre os sistemas, foram necessárias mudanças de variáveis e cálculos adicionais no lado direito do sistema linear dos métodos de pontos interiores. Essas adaptações envolveram operações extras que acumularam erros numéricos prejudicando a eficiência desta versão.

AGRADECIMENTOS

Este trabalho contou com o apoio financeiro do Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) e da Fundação de Amparo a Pesquisa do Estado de São Paulo (FAPESP).

ABSTRACT. This work presents two preconditioned versions of the conjugate gradient method. These versions differ from the classic version; they consider the linear system in form of normal equations. It will solve linear systems that arise from search directions computation in an interior point method. Determining the direction is the step with the largest computational effort and in large scale problems the use of direct methods might not be feasible. Therefore, one option is to use preconditioned iterative methods to solve the systems. The performances of these two versions of the preconditioned conjugate gradient methods are compared with the classic version. Numerical result shows that one of these versions is competitive with the classic one.

Keywords: linear programming, interior point method, conjugate gradient.

REFERÊNCIAS

[1] Miscellaneous LP models. Hungarian Academy of Sciences OR Lab. Online at `http://www.sztaki.hu/~meszaros/public_ftp/lptestset/misc`.

[2] Mittelmann – LP models. Miscellaneous LP models collect by Hans D. Mittelmann. Online at `http://plato.asu.edu/ftp/lptestset/pds/`.

[3] NETLIB collection LP test sets. NETLIB LP repository. Online at `http://www.netlib.org/lp/data/`

[4] Stochastic LP test sets. Hungarian Academy of Sciences OR Lab. Online at `http://www.sztaki.hu/~meszaros/public_ftp/lptestset/stochlp`

[5] S. Bocanegra, F.F. Campos & A.R.L. Oliveira. Using a hybrid preconditioner for solving large-scale linear systems arising from interior point methods, *Computational Optimization and Applications*, **36**(1–2) (2007), 149–164.

[6] R.S. Burkard and S. Karisch & F. Rendl. QAPLIB – A Quadratic Assignment Problem Library. *European Journal of Operations Research*, **55** (1991), 115–119.

[7] J. Czyzyk, S. Mehrotra, M. Wagner & S.J. Wright. PCx An Interior Point Code for Linear Programming. *Opt. Methods & Soft.*, **11-2**(1-4) (1999), 397–430.

[8] E.D. Dolan & J.J. Moré. Benchmarking optimization software with performance profiles. *Mathematical Programming*, **91** (2002), 201–213.

[9] A.R.L. Oliveira & 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** (2005), 1–24.

[10] Y. Saad. “Iterative Methods for Sparse Linear Systems”. SIAM Publications, SIAM, Philadelphia, PA, USA, (1997).

[11] M.I. Velazco, A.R.L. Oliveira & F.F. Campos. A note on hybrid preconditions for large scale normal equations arising from interior-point methods. *Opt. Methods & Soft.*, **25** (2010), 321–332.

[12] S.J. Wright. “Primal–Dual Interior–Point Methods”. SIAM Publications, SIAM, Philadelphia, PA, USA, (1996).