

Comparação entre variantes do Método dos Gradientes Conjugados aplicados a sistemas lineares originados dos Métodos de Pontos Interiores

Alessandro Fonseca Esteves Coelho

Universidade Estadual de Campinas
Cidade Universitária Zeferino Vaz, Distrito de Barão Geraldo, 13081-970, Campinas -SP,
Brasil
e-mail: alessandro.coelho@gmail.com

Aurelio Ribeiro Leite de Oliveira

Universidade Estadual de Campinas
Cidade Universitária Zeferino Vaz, Distrito de Barão Geraldo, 13081-970, Campinas -SP,
Brasil
e-mail: aurelio@ime.unicamp.br

Marta Ines Velazco Fontova

Faculdade Campo Limpo Paulista-FACCAMP
Rua Guatemala 167, Bairro Jd. América, 13231-230, Campo Limpo Paulista - SP, Brasil
e-mail: marta.velazco@gmail.com

Resumo

Sistemas de equações normais surgem naturalmente quando utilizamos métodos de pontos interiores para encontrar uma solução de problemas de programação linear. Quando trabalhamos com problemas de grande porte, uma boa alternativa é o uso de métodos iterativos para encontrar uma solução destes sistemas. Neste trabalho avaliaremos versões do método dos gradientes conjugados preconditionado que consideram a estrutura de equações normais destes sistemas. Estas versões são comparadas com a versão padrão do método dos gradientes conjugados. Resultados numéricos, com problemas de grande porte, mostram que uma dessas versões é competitiva em relação à versão padrão.

Palavras chave. Programação Linear. Métodos de Pontos Interiores. Gradientes Conjugados.

Área: Programação Matemática.

Abstract

Systems Normal equations arrive naturally when we use interior point methods to find a solution to linear programming problems. When working with large scale systems, a good alternative is the use of iterative methods to find the solution for those systems. In this work, we review variants of the preconditioned conjugate gradient method that takes into account the structure of the normal equations of those systems. These variants are compared with the standard conjugate gradient method. Numerical results, with large scale problems, show that one of these versions is competitive with the standard one.

Keywords. Linear Programming. Interior Point Method. Conjugate Gradient.

Area: Mathematical Programming.

1 Introdução

Existem métodos de pontos interiores onde resolvemos dois sistemas de equações lineares a cada iteração. Realizando operações algébricas obtemos sistemas na forma de equações normais e estes são simétricos e definidos positivos.

A solução destes dois sistemas é o passo mais caro e que demanda maior tempo e esforço computacional nos métodos de pontos interiores. Portanto, é extremamente importante uma boa escolha do método de solução que será utilizado nessa etapa. Tipicamente, métodos diretos são utilizados para obter a solução destes sistemas, por exemplo, a fatoração de Cholesky das matrizes simétricas definidas positivas [Adler, Resende, Veiga & Karmarkar (1989), Gondzio (1995)]. Porém, limitações de tempo e memória podem impedir o seu uso. Assim, abordagens iterativas são mais eficientes quando bons preconditionadores são utilizados devido ao mal-condicionamento da matriz nas iterações finais.

Uma abordagem híbrida para a solução destes sistemas foi proposta em [Bocanegra, Campos & Oliveira (2007)]. Neste abordagem, dois preconditionadores são utilizados. O primeiro, nas iterações iniciais, é um tipo de fatoração incompleta, denominada Fatoração Controlada de Cholesky, proposta em [Campos & Birkett (1998)]. O segundo, denominado preconditionador separador, foi proposto em [Oliveira & Sorensen (2005)]. A troca de preconditionadores acontece quando a Fatoração Controlada de Cholesky perde a eficiência e torna a convergência do método iterativo mais lenta.

No trabalho de [Bocanegra et al. (2007)] o método iterativo escolhido foi o método de gradientes conjugados preconditionado. Porém, este método não considera nenhuma estrutura especial que esse sistema possa ter. Os sistemas obtidos do método de pontos interiores são sistemas de equações normais. Essa característica não é explorada quando o método de gradientes conjugados preconditionado é utilizado. Entretanto, existem versões deste mesmo método que consideram essa estrutura em sua formulação.

O objetivo deste trabalho é avaliar a eficiência de duas versões de gradientes conjugados preconditionado que consideram a estrutura de equações normais dos sistemas simétricos e definidos positivos que surgem do método de pontos interiores. Assim, estaremos trabalhando com o método preditor-corretor utilizando diferentes versões de gradientes conjugados para obter as soluções dos sistemas lineares.

2 Problema de Programação Linear

Um problema de programação linear é definido como a otimização de uma função objetivo linear dentro de uma determinada região, denominada região factível. Assim um problema de programação linear na forma padrão tem a seguinte forma [Wright (1997)]:

$$\text{minimizar } c^t x, \text{ sujeito a } Ax = b, x \geq 0,$$

onde A é uma matriz $m \times n$ de posto completo e c , b e x são vetores coluna de dimensões apropriadas. Associado a este problema temos o problema dual:

$$\text{maximizar } b^t y, \text{ sujeito a } A^t y + z = c, z \geq 0,$$

onde y é o vetor $m \times 1$ de variáveis duais e z é o vetor $n \times 1$ de variáveis de folga duais.

2.1 Método Preditor-Corretor

Neste método de pontos interiores [Wright (1997)] calculamos direções para cada variável (x, y, z) . No método preditor-corretor teremos duas direções para cada variável. De forma resumida, podemos dizer que essas direções são construídas aplicando o método de Newton

nas condições de otimalidade. 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.

A primeira direção chamamos de direção afim escala e, a segunda, direção de correção, onde esta é composta pela correção não linear e a perturbação (centragem). A perturbação é calculada na segunda direção, pois, podemos verificar a qualidade da primeira direção e calcular a perturbação de acordo com ela. Abaixo, temos a direção afim escala ($\tilde{d}x, \tilde{d}y, \tilde{d}z$):

$$\begin{cases} A\tilde{d}x = r_p \\ A^t\tilde{d}y + \tilde{d}z = r_d \\ X\tilde{d}z + Z\tilde{d}x = r_a. \end{cases}$$

A segunda direção ($\hat{d}x, \hat{d}y, \hat{d}z$) é calculada assumindo que o passo primal e o passo dual seja igual a 1, isto é, $\alpha_p = \alpha_d = 1$ e com perturbação μ mas utilizando a mesma matriz do sistema anterior. Assim:

$$\begin{cases} A\hat{d}x = 0 \\ A^t\hat{d}y + \hat{d}z = 0 \\ X\hat{d}z + Z\hat{d}x = \mu e - \tilde{D}x\tilde{D}ze. \end{cases}$$

A perturbação μ é dada por uma heurística desenvolvida por Mehrotra [Mehrotra (1992)] e calculamos da seguinte forma: $\mu = (\tilde{\gamma}/\gamma)^3(\gamma/n)$, onde $\tilde{\gamma} = (x + \tilde{\alpha}_p\tilde{d}x)^t(z + \tilde{\alpha}_d\tilde{d}z)$.

Portanto, a direção final é dada pela soma das duas direções:

$$(dx, dy, dz) = (\tilde{d}x, \tilde{d}y, \tilde{d}z) + (\hat{d}x, \hat{d}y, \hat{d}z).$$

Isto é, a direção final pode ser obtida resolvendo diretamente o sistema que construímos somando os dois últimos sistemas apresentados.

O critério de convergência que utilizaremos é baseado nas condições de otimalidade.

3 Métodos Iterativos

Existem diferentes formas de encontrar a solução de um sistema linear do tipo $Ax = b$. Podemos utilizar métodos diretos ou métodos iterativos e existem diversos para as duas abordagens.

À medida que os problemas aumentam, o custo computacional dos métodos diretos pode ficar muito alto. Uma boa alternativa para solucionar esses sistemas é o uso de métodos iterativos. Estes métodos geram uma sequência de soluções aproximadas. Dizemos que o método converge quando esta sequência converge para a solução do sistema linear.

3.1 Método dos Gradientes Conjugados Precondicionado

Uma forma de melhorar a eficiência e a robustez dos métodos iterativos consiste em utilizar precondicionadores. Um precondicionador M válido para o método de gradientes conjugados deve ser simétrico definido positivo. Atendida esta condição, podemos escrever uma versão precondicionada para método de gradientes conjugados [Saad (2003)].

Algoritmo 3.1 *GCP - Gradientes Conjugados Precondicionado*

Calcule $r_0 = b - Ax_0$, $z_0 = M^{-1}r_0$ e $p_0 = z_0$.

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

$$\begin{aligned}\alpha_j &= \frac{(r_j, z_j)}{(Ap_j, p_j)} \\ x_{j+1} &= x_j + \alpha_j p_j \\ r_{j+1} &= r_j - \alpha_j Ap_j \\ z_{j+1} &= M^{-1} r_{j+1} \\ \beta_j &= \frac{(r_{j+1}, z_{j+1})}{(r_j, z_j)} \\ p_{j+1} &= z_{j+1} + \beta_j p_j\end{aligned}$$

Fim

3.2 GCP para Equações Normais

Uma das formas de converter um sistema linear não simétrico em um sistema simétrico consiste em resolver o sistema equivalente $A^t Ax = A^t b$. Assim, estamos premultiplicando o sistema original por A^t e, quando fazemos isso, podemos aplicar o método de gradientes conjugados no novo sistema e obter o algoritmo versão NR (N de “Normal” e R de “Residual” [Saad (2003)]). Outra opção para obter um sistema simétrico é resolver $AA^t u = b$ onde $x = A^t u$. Neste caso, obtemos a versão NE (N de “Normal” e E de “Error” [Saad (2003)]) quando aplicamos o método de gradientes conjugados no novo sistema. Nos dois casos obtemos os chamados sistemas de equações normais.

Existem diferentes versões do GCP para equações normais [Saad (2003)]. Destacaremos neste trabalho apenas duas versões, ambas utilizam o preconditionador à esquerda. Denominaremos estas versões de PGCNE e PGCNR.

O PGCNR é facilmente derivado do Algoritmo 3.1. Nesta versão do método consideramos que o sistema original $Ax = b$ foi premultiplicado pela matriz A^t , isto é, aplicaremos o condicionamento ao sistema de equações normais $A^t Ax = A^t b$. Vamos denotar por $r_j = b - Ax_j$, o resíduo do sistema original, $\tilde{r}_j = A^t r_j$ o resíduo para o sistema de equações normais e, finalmente, o resíduo do sistema preconditionado por $z_j = M^{-1} \tilde{r}_j$. O escalar α_j será dado por $\alpha_j = \frac{(\tilde{r}_j, z_j)}{(A^t Ap_j, p_j)} = \frac{(\tilde{r}_j, z_j)}{(Ap_j, Ap_j)}$. Estas relações nos sugerem o uso de um vetor auxiliar $w_j = Ap_j$ no método que assume a seguinte forma:

Algoritmo 3.2 PGCNR

Calcule $r_0 = b - Ax_0$, $\tilde{r}_0 = A^t r_0$, $z_0 = M^{-1} \tilde{r}_0$, $p_0 = z_0$.

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

$$\begin{aligned}w_j &= Ap_j \\ \alpha_j &= \frac{(z_j, \tilde{r}_j)}{(w_j, w_j)} \\ x_{j+1} &= x_j + \alpha_j p_j \\ r_{j+1} &= r_j - \alpha_j w_j \\ \tilde{r}_{j+1} &= A^t r_{j+1} \\ z_{j+1} &= M^{-1} \tilde{r}_{j+1} \\ \beta_j &= \frac{(z_{j+1}, \tilde{r}_{j+1})}{(z_j, \tilde{r}_j)} \\ p_{j+1} &= z_{j+1} + \beta_j p_j\end{aligned}$$

Fim

O PGCNE é facilmente derivado do Algoritmo 3.1. Basta preconditionarmos à esquerda o sistema $AA^t u = b$, com $x = A^t u$, e aplicar o método GCP. Observe que o método será construído para a variável auxiliar u . Usaremos o vetor auxiliar $w = A^t p_j$. Temos assim a versão preconditionada à esquerda do GCNE na variável auxiliar u , isto é, o PGCNE - variável u .

Algoritmo 3.3 *PGCNE - variável u*

Calcule $r_0 = b - AA^t u_0$, $z_0 = M^{-1} r_0$, $p_0 = z_0$

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

$$\begin{aligned} w_j &= A^t p_j \\ \alpha_j &= \frac{(z_j, r_j)}{(w_j, w_j)} \\ u_{j+1} &= u_j + \alpha_j p_j \\ r_{j+1} &= r_j - \alpha_j A w_j \\ z_{j+1} &= M^{-1} r_{j+1} \\ \beta_j &= \frac{(z_{j+1}, r_{j+1})}{(z_j, r_j)} \\ p_{j+1} &= z_{j+1} + \beta_j p_j \end{aligned}$$

Fim

Podemos, usando este último método, recuperar a variável x fazendo o produto $x = A^t u$ ou escrever o método diretamente na variável original x .

Neste trabalho, estamos interessados em resolver problemas de programação linear utilizando os métodos de pontos interiores. Utilizaremos os Algoritmos 3.2 e 3.3 para resolver os sistemas lineares. Uma implementação preliminar comparando o PGCNE com o GCP foi apresentada em [Coelho, Oliveira & Velazco (2010)].

4 Precondicionadores para o Sistema de Equações Normais

A operação mais cara do método de pontos interiores preditor-corretor consiste em encontrar as direções de busca resolvendo dois sistemas lineares conforme visto na Seção 2. Estes sistemas podem ser escritos na forma de equações normais e então resolvidos por métodos diretos como, por exemplo, a fatoração de Cholesky. Algumas vezes o uso de métodos diretos pode levar a problemas de armazenamento ou tempo de processamento. Nestas situações o uso de métodos iterativos pode ser a única alternativa de solução.

O sucesso da implementação, usando métodos iterativos, depende de uma boa escolha do precondicionador. Diversos precondicionadores têm sido usados para resolver sistemas de equações normais originados do método de pontos interiores. Alguns destes precondicionadores são baseados na fatoração incompleta de Cholesky. Esta classe de precondicionadores é mais eficiente nas iterações iniciais do método de pontos interiores e perde eficiência quando o método se aproxima de uma solução. Por outro lado, próximo da convergência, o precondicionador separador proposto em [Oliveira & Sorensen (2005)], tem comportamento oposto comparado à fatoração incompleta de Cholesky, isto é, sua eficiência é melhor próximo a uma solução do problema de programação linear.

4.1 Precondicionador Híbrido

Nas iterações de pontos interiores a matriz $Z^{-1}X$ se altera muito rapidamente e, nas iterações finais, se torna muito mal-condicionada. Assim, é difícil encontrar um precondicionador que obtenha bom desempenho para todas as iterações.

Trabalharemos com um precondicionador híbrido M , utilizando o método dos gradientes conjugados, para resolver o sistema de equações normais:

$$M^{-1}(AZ^{-1}XA^t)M^{-t}\bar{y} = M^{-1}(AZ^{-1}X(r_d - X^{-1}r_a) + r_p)$$

onde $\bar{y} = M^t dy$. Desta forma, teremos duas fases no método de pontos interiores. Usaremos a fatoração controlada de Cholesky na primeira fase para construir a matriz M e o precondicionador separador para construir a matriz M na segunda fase.

4.2 Fatoração Controlada de Cholesky

Consideremos a fatoração de Cholesky $AZ^{-1}XA^t = LL^t = \tilde{L}\tilde{L}^t + R$ onde L é o fator obtido quando a fatoração é completa, \tilde{L} quando é incompleta e R é a matriz resíduo. Definindo $E = L - \tilde{L}$ temos: $\tilde{L}^{-1}(AZ^{-1}XA^t)\tilde{L}^{-t} = (\tilde{L}^{-1}L)(\tilde{L}^{-1}L)^t = (I + \tilde{L}^{-1}E)(I + \tilde{L}^{-1}E)^t$.

Podemos ver que quando a matriz \tilde{L} se aproxima da matriz L , a matriz E se aproxima da matriz nula e então $\tilde{L}^{-1}(AZ^{-1}XA^t)\tilde{L}^{-t} \approx I$. A fatoração controlada de Cholesky [Campos & Birkett (1998)] é baseada na minimização da norma de Frobenius da matriz E . Consideremos o seguinte problema:

$$\text{minimizar } \|E\|_F^2 = \sum_{j=1}^m c_j \text{ com } c_j = \sum_{i=1}^m |l_{ij} - \tilde{l}_{ij}|^2.$$

Agora podemos dividir c_j em dois somatórios:

$$c_j = \sum_{k=1}^{t_j+\eta} |l_{kj} - \tilde{l}_{kj}|^2 + \sum_{k=t_j+\eta+1}^m |l_{kj}|^2$$

onde t_j representa o número de entradas não nulas abaixo da diagonal da j -ésima coluna da matriz $AZ^{-1}XA^t$ e η representa o número de entradas extras permitidas por coluna. O primeiro somatório contém todas as $t_j + \eta$ entradas não nulas da j -ésima coluna de \tilde{L} . O segundo somatório contém apenas as entradas restantes do fator completo L que não têm entradas correspondentes em \tilde{L} . Considerando que $\tilde{l}_{ij} \approx l_{ij}$ quando $\eta \rightarrow m$ e l_{ij} não é calculado, $\|E\|_F$ é minimizado baseado em uma heurística modifica o primeiro somatório.

Aumentando η , c_j irá diminuir, pois, o primeiro somatório contém um número maior de termos. Além disso, $\|E\|_F$ é minimizado escolhendo $t_j + \eta$ que contém as maiores entradas em valor absoluto de \tilde{L} eliminando as maiores entradas correspondentes em L .

4.3 Precondicionador Separador

O preconditionador separador foi proposto para sistemas aumentados originados do método de pontos interiores [Oliveira & Sorensen (2005)] para problemas de programação linear. Esse preconditionador é uma generalização do proposto por [Resende & Veiga (1993)] em um contexto de problemas de fluxo de custo mínimo em rede. A principal característica deste preconditionador é que ele trabalha bem próximo a uma solução do problema de programação linear onde este já está muito mal-condicionado. Porém, se usado nas iterações iniciais, o método não converge para vários problemas de programação linear.

Usaremos a versão do preconditionador separador para o sistema de equações normais [Oliveira & Sorensen (2005)] conforme descrito a seguir.

Seja $A = [BN]P$ onde P é uma matriz de permutação tal que B é não singular. Fazendo $D = Z^{-1}X$ temos: $ADA^t = BD_B B^t + ND_N N^t$. Multiplicando por $D_B^{-\frac{1}{2}} B^{-1}$ e pós-multiplicando pela transposta chegamos a

$$T = D_B^{-\frac{1}{2}} B^{-1} (ADA^t) B^{-t} D_B^{-\frac{1}{2}} = I + WW^t \text{ onde } W = D_B^{-\frac{1}{2}} B^{-1} N D_N^{\frac{1}{2}}.$$

Observe que a matriz preconditionada é definida positiva e tem autovalores maiores ou iguais a um, isto é, não tem autovalores próximos de zero.

Próximo a uma solução do problema de programação linear pelo menos $n - m$ entradas da matriz D são pequenas. Assim, com uma escolha adequada das colunas de B , a diagonal das matrizes D_B^{-1} e D_N são muito pequenas próximas a solução. Nesta situação a matriz W se aproxima da matriz nula e, então, a matriz T se aproxima da matriz identidade.

4.4 Mudança de Fase

A mudança de fase é um aspecto muito importante para o bom desempenho desta abordagem. Usaremos a heurística para a mudança de fase desenvolvida em [Velazco, Oliveira & Campos (2010)]. Assim, se o número de iterações necessárias para a convergência do método de gradientes conjugados é maior que $m/5$, o parâmetro η na fatoração controlada de Cholesky é incrementado em $\eta = \eta + 10$. A mudança de preconditionador acontece quando η ultrapassa um valor máximo pré fixado.

4.5 Métodos de Pontos Interiores e Métodos Iterativos

O passo mais caro de uma iteração de pontos interiores é a solução do sistema linear:

$$(ADA^t)dy = r_p + A(Dr_d - Z^{-1}r_a). \quad (1)$$

Para resolver o sistema (1) vamos compará-lo com o seguinte sistema de equações normais apresentado na Seção 3: $GG^t u = b$ com $x = G^t u$.

Vamos comparar o sistema (1) com o sistema de equações normais na variável u , isto é, $GG^t u = b$. Isto se justifica, pois, como queremos obter o valor do vetor dy no sistema (1), este será dado pelo vetor u no sistema $GG^t u = b$. Notemos também que a matriz G é exatamente a matriz $AD^{\frac{1}{2}}$ no sistema (1) e o vetor b é o vetor $r_p + A(Dr_d - Z^{-1}r_a)$.

Outra forma de se resolver o sistema (1) é compará-lo com o segundo tipo de equações normais apresentado na Seção 3: $G^t Gx = G^t b$.

Notemos que a matriz G será substituída por $D^{-\frac{1}{2}}A^t$ e manteremos a variável x mas esta indicará o valor de dy . O vetor $r_p + A(D^{-1}r_d - Z^{-1}r_a)$ do sistema (1) é representado por $G^t b$ no sistema $G^t Gx = G^t b$. Portanto o lado direito do sistema (1) é o resultado do produto de $AD^{-\frac{1}{2}}$ pelo vetor b . Precisamos recuperar o vetor b do sistema (1).

Desta forma: $AD^{-\frac{1}{2}}b = r_p + A(Dr_d - Z^{-1}r_a) = b - Ax + A(D^{-1}r_d - Z^{-1}r_a)$.
 Fazendo $\tilde{b} = A^t(AA^t)^{-1}b$ temos:

$$A\tilde{b} - Ax + A(D^{-1}r_d - Z^{-1}r_a) = AD^{-1}(D\tilde{b} - Dx + r_d - DZ^{-1}r_a).$$

Portanto $b = D^{-\frac{1}{2}}(D\tilde{b} - Dx + r_d - DZ^{-1}r_a)$ para o sistema (1).

Após estas observações, podemos aplicar os Algoritmos 3.2 e 3.3 para resolver o sistema (1), utilizando o preconditionador híbrido proposto na Seção 4.1.

5 Experimentos Numéricos

Os experimentos numéricos foram realizados utilizando o PCx [Czyzyk, Mehrotra, Wagner & Wright (1999)]. O PCx foi implementado utilizando o método preditor-corretor com múltiplas correções e, para a resolução dos sistemas lineares que surgem do método de pontos interiores, um método direto foi adotado. Os testes deste trabalho foram realizados numa versão modificada do PCx onde as múltiplas correções foram eliminadas e o método direto foi substituído pelo método dos gradientes conjugados preconditionado 3.1. Todos os testes foram realizados em um processador Intel Core 2 Duo 2.2GHz com 2Gb de memória, em ambiente Linux usando os compiladores gcc e gfortran.

5.1 Problemas Teste

Os problemas utilizados são de domínio público extraídos de diversas bibliotecas: NETLIB, STOCHPL¹, MISC². Os modelos QAP são da coleção QAPLIB [Burkard, Karisch & Rendl

¹Stochastic LP test sets n.d.

²Miscellaneous LP models n.d., Mittelmann - LP models n.d.

(1991)]. A Tabela 1 indica o número de elementos não nulos dos problemas testados e a qual biblioteca pertence cada problema.

Problema	Não Nulos	Biblioteca	Problema	Não Nulos	Biblioteca
share1b	1148	NETLIB	scsd8-2c-64	112770	STOCHLP
share2b	777	NETLIB	scsd8-2r-432	190210	STOCHLP
sctap1	1802	NETLIB	stocfor3	62960	NETLIB
sctap2	7052	NETLIB	els19	50882	QAP
sctap3	9383	NETLIB	chr22b	36520	QAP
scsd1	2388	NETLIB	scr15	24060	QAP
scsd6	4316	NETLIB	scr20	61780	QAP
scsd8	8584	NETLIB	rou20	152980	QAP
ship04s	2875	NETLIB	ste36a	512640	QAP
ship04l	4290	NETLIB	ste36b	512640	QAP
ship12s	4297	NETLIB	qap12	33528	NETLIB
ship12l	9254	NETLIB	qap15	85470	NETLIB
seba	4102	NETLIB	nug12	33528	QAP
pilot4	5001	NETLIB	nug15	85470	QAP
pilotwe	8841	NETLIB	pds-10	103725	MISC
pilotnov	11934	NETLIB	pds-20	226494	MISC
scfxm2	4919	NETLIB	pds-30	333260	MISC
scfxm3	7381	NETLIB	pds-40	457538	MISC
fit1p	9868	NETLIB	pds-50	581152	MISC
fit2p	50284	NETLIB	pds-60	709178	MISC
fit1d	13427	NETLIB	pds-70	822526	MISC
fit2d	129042	NETLIB	pds-80	916852	MISC
25fv47	10538	NETLIB	pds-90	1002902	MISC
scsd8-2b-64	112770	STOCHLP	pds-100	1096002	MISC

Tabela 1: Dados dos Problemas Teste

5.2 Resultados Computacionais

Para avaliar a eficiência dos métodos sugeridos comparamos o total de iterações de pontos interiores, o total de iterações do método de gradientes conjugados e o tempo total para resolver os problemas. Os métodos iterativos testados são duas versões de gradientes conjugados descritos no Capítulo 3, isto é, os Algoritmos 3.2 e 3.3 foram comparados ao Algoritmo 3.1. Destacamos que os Algoritmos 3.2 e 3.3 estão escritos para um sistema de equações normais qualquer. Assim, as versões implementadas no trabalho, foram escritas para os sistemas de equações normais que surgem do método de pontos interiores utilizado.

Na Tabela 2, apresentamos apenas os problemas que a versão testada, versão PGCNR, encontrou o valor ótimo e comparamos com a versão PGC. Nesta tabela encontramos o total de iterações realizadas no método de pontos interiores, o total de iterações realizadas pelo método iterativo, isto é, o método de gradientes conjugados utilizado, e o tempo total, em segundos, de convergência de cada problema. O total de iterações de gradientes conjugados foi dado pela soma do número de iterações realizadas para o cálculo de cada uma das duas direções do método de pontos interiores e assim foi feito para todas as iterações de pontos interiores. As iterações de pontos interiores estarão indicadas entre parênteses. A última linha da tabela mostra a média aritmética das iterações de pontos interiores, das iterações de gradientes conjugados e do tempo.

Na grande parte dos problemas da Tabela 2 a versão PGC se mostrou totalmente superior, isto é, convergiu em menor tempo, com menor número de iterações de pontos interiores e, também, de gradientes conjugados. Apesar disso, para os problemas *ship04l*, *fit1p* e *fit2p* a versão PGCNR foi superior no total de iterações de pontos interiores.

Os resultados apresentados na Tabela 2 foram construídos a partir dos problemas que convergiram para a versão PGCNR. Assim, todos os outros problemas apresentados na

Problema	Iterações do PGC	Iterações do PGCNR	Tempo - PGC	Tempo - PGCNR
share1b	45 (19)	273 (19)	0,04	0,06
share2b	142 (17)	143 (17)	0,02	0,02
sctap1	362 (16)	585 (21)	0,08	0,11
sctap2	93 (14)	188 (20)	0,16	0,27
sctap3	191 (16)	607 (24)	0,31	0,64
scsd1	146 (9)	163 (12)	0,03	0,03
scsd6	263 (13)	263 (13)	0,05	0,06
scsd8	272 (12)	272 (12)	0,13	0,17
ship04s	495 (15)	544 (18)	0,15	0,23
ship04l	609 (19)	630 (18)	0,36	0,32
ship12s	885 (14)	907 (13)	0,31	0,33
ship12l	1133 (17)	1194 (17)	1,01	1,23
seba	5091 (16)	6162 (18)	2,48	5,28
pilot4	1022 (48)	2084 (48)	0,97	1,42
pilotwe	2419 (48)	2429 (49)	1,58	1,63
pilotnov	3675 (18)	3673 (18)	2,13	2,40
scfxm2	884 (20)	868 (21)	0,49	0,52
scfxm3	1348 (20)	1376 (22)	0,90	0,99
fit1p	37074 (43)	36242 (40)	56,46	49,98
fit2p	171213 (48)	170759 (47)	2558,26	2863,63
fit1d	58 (19)	58 (19)	0,08	0,10
fit2d	56 (25)	56 (25)	0,98	1,03
25fv47	4185 (26)	4630 (28)	4,19	5,02
scsd8-2b-64	386 (7)	878 (8)	3,39	6,06
scsd8-2c-64	382 (7)	5864 (10)	4,04	20,00
scsd8-2r-432	6519 (18)	11397 (22)	39,28	180,96
pds-10	6872 (47)	8851 (49)	63,36	77,26
pds-30	31129 (73)	70946 (79)	741,93	1748,39
pds-40	43516 (79)	127153 (84)	1463,64	4813,55
média	11050,51(25,62)	15834,31(27,27)	170,57	337,29

Tabela 2: Total de Iterações de Gradientes Conjugados NR e Pontos Interiores - Tempo

Tabela 1 não atingiram seu valor ótimo para a versão PGCNR.

A principal motivação para testarmos estas duas versões de gradientes conjugados se baseia no fato de elas considerarem a estrutura de equações normais do sistema, aproveitando assim sua estrutura natural, no caso, os sistemas apresentados para o cálculo das direções do método de pontos interiores. Porém, para a versão PGCNR, necessitamos de um esforço extra, isto é, recuperar o lado direito do sistema como descrito na Seção 4.5. Portanto, apesar da versão PGCNR aproveitar a estrutura de equações normais, esta versão fica muito comprometida, pois, necessita recuperar um vetor para a sua utilização.

A versão PGCNE não necessita deste esforço extra. Esta versão aproveita a estrutura de equações normais do sistema sem a necessidade de recuperar nenhum vetor. Os problemas testados com a versão PGCNE atingiram o valor ótimo para praticamente todos os problemas. Isto mostra uma maior competitividade da versão PGCNE com a versão PGC.

Visto que praticamente todos os problemas apresentados na Tabela 1 atingem o valor ótimo, selecionamos os problemas de maior porte para apresentar os resultados e destacamos que nos problemas omitidos os dois métodos se comportam praticamente da mesma forma quando comparamos as iterações de pontos interiores, gradientes conjugados e o tempo total. A Tabela 3, utilizando o mesmo formato da Tabela 2, mostra os resultados obtidos para os problemas selecionados.

Observamos que os métodos comparados se comportam de forma muito parecida em parte dos problemas testados. Apesar disso ocorrem diferenças consideráveis entre os métodos. Variações no tempo e no total de iterações de gradientes conjugados ocorreram em todos os problemas e, em alguns casos, até mesmo no total de iterações de pontos interiores. Destacamos os problemas *ste36b* e *nug15*. O primeiro atingiu a solução ótima

Problema	Iterações do PGC	Iterações do PGCNE	Tempo - PGC	Tempo - PGCNE
stocfor3	69507 (32)	69828 (32)	338,54	345,72
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
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
média	66933,59 (47,5)	77606,81(48,13)	3831,67	5512,82

(-) significa que o método falhou.

Tabela 3: Total de Iterações de Gradientes Conjugados e Pontos Interiores NE - Tempo

apenas quando foi usado o método de gradientes conjugados preconditionado que considera a estrutura de equações normais, isto é, o PGCNE. Entretanto, o segundo problema atingiu a solução ótima apenas quando o método de gradientes conjugados preconditionado não considera a estrutura de equações normais do sistema, isto é, o PGC.

Existem grandes diferenças nos resultados apresentados para as versões testadas. Ocorreram variações em todos os itens avaliados, ou seja, até mesmo no total de iterações de pontos interiores. Isto é importante destacar, pois, a cada iteração de pontos interiores resolvemos dois sistemas lineares e encontrar esta solução é a etapa que exige um grande esforço computacional. Portanto sempre buscamos reduzir o total de iterações de pontos interiores, sobre esse ponto de vista, a versão PGCNE é a opção indicada.

Como dissemos anteriormente, a motivação para testar essas duas versões de gradientes conjugados baseava-se em aproveitar a estrutura de equações normais do sistema de equações lineares. A versão PGCNE aproveita essa estrutura sem custos adicionais e obteve resultados semelhantes quando comparada com a versão encontrada no PCx modificado, isto é, a versão PGC. Porém, a versão PGCNR, apesar de também aproveitar a estrutura de equações normais do sistema, não obteve resultados tão satisfatórios. Isto ocorreu porque esta versão necessitou recuperar o lado direito do sistema. Com isso acumulamos muitos erros numéricos e isto custou um pior desempenho para muitos problemas e, para os problemas de maior porte, não conseguimos obter a solução ótima.

Enfim, para os problemas testados, a versão PGCNE se mostrou extremamente competitiva com a versão PGC.

6 Conclusões

Os métodos de pontos interiores são uma boa opção quando trabalhamos com problemas de grande porte. O ponto crítico destes métodos consiste na solução de dois sistemas de equações lineares que surgem a cada iteração. A matriz dos coeficientes deste sistema

muda a cada iteraçao dos métodos de pontos interiores, o que significa que a cada iteraçao temos uma nova matriz para trabalhar. Assim, estratégias eficientes devem ser adotadas para a soluçao destes sistemas que consiste no passo mais caro do método. Restriçoes de tempo e memória impedem, muitas vezes, o uso de métodos diretos. Portanto, o uso de métodos iterativos pode ser a única opçao em alguns casos. Contudo, o sucesso do método iterativo adotado, depende de um bom condicionador.

Neste trabalho, apresentamos resultados no contexto descrito. Necessitavamos resolver problemas de programação linear e, para isso, utilizamos um método de pontos interiores no qual as soluçoes dos sistemas que surgem foram obtidas utilizando um método iterativo condicionado. Avaliamos a eficiência de diferentes versoes do método de gradientes conjugados condicionado utilizando um condicionador híbrido.

A principal motivaçao para testar diferentes versoes de gradientes conjugados condicionado vem da estrutura natural que os sistemas obtidos no método preditor-corretor aparecem, isto é, sistemas de equações normais. Motivados por esse fato, o uso de versoes do método iterativo que consideram essa estrutura em sua formulaçao foram avaliadas. Estudos anteriores já haviam sido realizados com uma versao de gradientes conjugados condicionado mas, neste caso, a versao não considerava a estrutura de equações normais.

Para avaliar a eficiência de cada uma das versoes consideramos o total de iteraçoes do método de pontos interiores, de gradientes conjugados e o tempo de processamento total. Inicialmente, testamos os problemas selecionados para a versao de gradientes conjugados que foi nossa base para comparaçao, PGC. O próximo passo foi implementar e testar as outras versoes, PGCNE e PGCNR.

A implementaçao da versao PGCNE no código PCx ocorreu naturalmente visto que a estrutura do método se adequava muito bem aos sistemas encontrados. Os resultados obtidos com essa versao foram muito semelhantes aos da versao PGC. Em poucos casos ocorreram variaçoes no total de iteraçoes de pontos interiores. De maneira geral, o tempo e total de iteraçoes de gradientes conjugados também não apresentaram grandes diferenças. Portanto, a versao PGCNE obteve bom desempenho e competitividade com a versao PGC.

A versao PGCNR, já apresentou dificuldades teóricas antes da implementaçao. Foram necessárias maiores adequaçoes para ser incorporada aos métodos de pontos interiores. Isto ocorreu, pois, a versao PGCNR considera que o sistema de equações normais é obtido a partir de um sistema original que foi premultiplicado pela matriz transposta dos coeficientes do sistema. Assim, o lado direito do sistema também será premultiplicado por esta mesma matriz. Este foi o ponto que necessitou de maiores ajustes para a implementaçao, uma vez que, o sistema gerado pelo método de pontos interiores, já se encontrava nesta forma. Portanto, foi necessário modificar o lado direito destes sistemas, isto é, recuperar um vetor que havia sido premultiplicado por uma matriz. Essas adaptaçoes envolvem muitas operaçoes, até mesmo a resoluçao adicional de um sistema. Com isso, acumulamos erros numéricos e assim, o método ficou prejudicado perdendo eficiência. Esta perda de eficiência foi confirmada nos testes realizados. Entre os problemas testados, 19 não atingiram uma soluçao ótima sendo a maioria problemas de grande porte. Assim, observamos muitas variaçoes no total de iteraçoes dos métodos de pontos interiores, e de gradientes conjugados e, também, no tempo total de convergência.

Quando utilizamos a versao PGCNR não encontramos a soluçao ótima, para os problemas de maior dimensao, indicando um desempenho inferior deste método em relaçao ao PGC e ao PGCNE. Ainda assim, em alguns problemas, o PGCNR obteve um bom desempenho. Portanto, o seu uso pode ser mais adequado em outras aplicaçoes, por exemplo, aplicaçoes onde não seja necessário recuperar o lado direito do sistema linear.

Neste trabalho, testamos métodos iterativos que trabalham com uma das duas formulaçoes algébricas mais comuns dos sistemas obtidos do método de pontos interiores, isto

é, sistemas de equações normais onde a matriz dos coeficientes é simétrica e definida positiva. Em trabalhos futuros, podemos avaliar métodos iterativos que trabalhem com a outra formulação destes sistemas, o sistema aumentado e, neste caso, a matriz dos coeficientes é simétrica e indefinida.

Agradecimentos

Este trabalho contou com o apoio financeiro do Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq).

Referências

- Adler, I., Resende, M. G. C., Veiga, G. & Karmarkar, N. (1989). An implementation of Karmarkar's algorithm for linear programming, *Mathematical Programming* **44**: 297–335.
- Bocanegra, S., Campos, F. F. & Oliveira, A. R. L. (2007). Using a hybrid preconditioner for solving large-scale linear systems arising from interior point methods, *Computational Optimization and Applications* **36**(1–2): 149–164.
- Burkard, R. S., Karisch, S. & Rendl, F. (1991). QAPLIB - a quadratic assignment problem library, *European Journal of Operations Research* **55**: 115–119.
- Campos, F. F. & Birkett, N. R. C. (1998). 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 J. Sci. Comput.* **19**(1): 126–138.
- Coelho, A. F. E., Oliveira, A. R. L. & Velazco, M. I. (2010). Solução de sistemas lineares de grande porte usando uma variante do método dos gradientes conjugados, *Anais do XLII Simpósio Brasileiro de Pesquisa Operacional, em CD-ROM* pp. 2445–2456.
- Czyzyk, J., Mehrotra, S., Wagner, M. & Wright, S. J. (1999). PCx an interior point code for linear programming, *Optimization Methods & Software* **11-2**(1-4): 397–430.
- Gondzio, J. (1995). HOPDM (Version 2.12) – A fast LP solver based on a primal–dual interior point method, *European Journal of Operational Research* **85**: 221–225.
- Mehrotra, S. (1992). On the implementation of a primal-dual interior point method, *SIAM Journal on Optimization* **2**(4): 575–601.
- Oliveira, A. R. L. & Sorensen, D. C. (2005). 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.
- Resende, M. G. C. & Veiga, G. (1993). An efficient implementation of a network interior point method, *DIMACS Series in Discrete Mathematics and Theoretical Computer Science* **12**: 299–348.
- Saad, Y. (2003). *Iterative Methods for Sparse Linear Systems*, SIAM Publications, SIAM, Philadelphia, PA, USA.
- Velazco, M. I., Oliveira, A. R. L. & Campos, F. F. (2010). A note on hybrid preconditions for large scale normal equations arising from interior-point methods, *Optimization Methods and Software* **25**: 321–332.



Wright, S. J. (1997). *Primal–Dual Interior–Point Methods*, SIAM Publications, SIAM, Philadelphia, PA, USA.