El método de gradiente conjugado proyectado precondicionado (PPGC) en optimización con restricciones

Leobardo Valera [*] Miguel Argáez [**] Brígida Molina [***]

6 de diciembre de 2011

Resumen

En este trabajo se propone utilizar el algoritmo gradiente conjugado proyectado precondidionado (PPCG) para resolver un problema de minimización cuadrática con restricciones de igualdad. Para acelerar la velocidad de convergencia del método de gradiente conjugado proyectado (PCG), se plantea precondicionar el sistema KKT usando la factorización incompleta de Cholesky del Hessiano de la función objetivo. Finalmente se muestran algunos resultados numéricos que se obtienen al resolver varios problemas de la librería CUTEr, usando el algoritmo PCG y su versión precondicionada PPCG, los cuales muestran la efectividad del algoritmo.

1. Introducción

Problemas de optimización lineal con restricciones de igualdad surgen en muchas y variadas aplicaciones entre otras en economía, en la solución de

[*]Escuela de Matemáticas, Universidad Metropolitana, lvalera@unimet.edu.ve

[**]Department of Mathematical Sciences University of Texas at El Paso, mar@math.utep.edu

[***]Centro de Cálculo Cienttífico y Tecnológico (CCCT), Escuela de Computación, Facultad de Ciencias, Universidad Central de Venezuela. brigida.molina@ciens.ucv.ve Facultad de Ciencias, Universidad Central de Venezuela. brigida.molina@ciens.ucv.ve

ecuaciones diferenciales parciales para materiales incomprensibles, en la recuperación de señales de entrada. Un problema de optimización cuadrática con restricciones de igualdad tiene la forma:

$$\begin{array}{ll} \text{minimizar} & \frac{1}{2}x^T G x + g^T x \\ \text{sujeto a} & Ax = b \end{array} \tag{1}$$

donde $G \in \mathbb{R}^{n\times n}$ es una matriz simétrica, $A \in \mathbb{R}^{m\times n}$ con $m < n$, es una matriz de rango completo , $g \in \mathbb{R}^n$, $b \in \mathbb{R}^m$ son vectores dados, $x \in \mathbb{R}^n$ es el vector a determinar. Las condiciones necesarias y suficientes para resolver (1) son A sea de rango completo y G sea definida positiva sobre el núcleo de A. Resolver (1) es equivalente a resolver el siguiente sistema de ecuaciones lineales, llamado las condiciones Karush-Kuhn-Tucker (KKT)

$$\begin{pmatrix} G & A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} -g \\ b \end{pmatrix}, \tag{2}$$

donde $y \in \mathbb{R}^m$ es el multiplicador de Lagrange [12] .

Este tipo de sistemas aparecen en muchas investigaciones en ciencia e ingeniería. Resolver este problema eficientemente es la línea de investigación actual [5, 6, 7, 13, 14, 15, 16]. Benzi et al. en [5] presenta un estudio exhaustivo de este problema.

Los métodos para resolver (2) se pueden clasificar en dos grupos: El primero usa un método iterativo sobre todo el sistema como por ejemplo: Gradiente Conjugado (**CG**), Mínimo Residuo (**MINRES**), entre otros. El segundo reduce el sistema de una manera equivalente a un sistema de menor dimensión, entre ellos está el **Shur Complement**, y el Gradiente Conjugado Proyectado, **PCG** [12] entre otros.

Uno de los propositos de este artículo es divulgar los trabajos de investigación del Profesor Argaez en el área **PCG**, y presentar nuevas líneas de investigación donde se resuelva el sistema eficientemente sin la necesidad del almacenarlo [1, 2, 3, 4]. La idea fundamental del PCG es conseguir una solución particular para las restricciones lineales y luego buscar la solución del problema moviendose en una variedad afín determinada por la solución particular y el subespacio generado por el núcleo de A.

La solución particular es determinada por la solución de norma mínima, la cual esta en el espacio fila de la matriz A. Luego el problema se reduce a encontrar la solución de un sistema lineal positivo definida sobre el núcleo de A, este sistema lineal es llamada ecuación proyectada. Para resolver la ecuación proyectada aplicamos el algoritmo **CG** el cual nos garantiza la solución en $n - m$ iteraciones trabajando en aritmética infinita.

El algoritmo tiene la propiedad que cada iteración satisface las restricciones de igualdad del problema (1), la sucesión de aproximaciones a la solución, y la sucesión de residuos son estrictamente crecientes y estrictamente decrecientes en norma euclideana, respectivamente. Estas propiedades serían las propiedades que tendríamos en el caso que el sistema (2) fuera difinido positivo, desafortunadamente el sistema no lo es, por lo tanto el uso del **CG** no garantiza una solución del problema.

En [4] se demuestra que un buen precondicionador para la información de segundo orden del problema produce un buen precondicionador para el algoritmo PCG, esto es, basta con encontrar una buena aproximación de G.

En este trabajo proponemos seleccionar, para resolver el problema (2), al método PPCG usando como precondicionador a la factorización incompleta de Cholesky del Hessiano de la función objetivo.

El artículo esta organizado de la siguiente manera: en la sección 2 se hace una breve revisión de como encontrar la solución de un problema de optimización cuadrática con restricciones lineales de forma directa; posteriormente, en esa misma sección, se plantea el mismo problema pero ahora en función de un operador proyección sobre el espacio nulo de A para finalizar la sección con la descripción del método gradiente conjugado proyectado y se muestra el algoritmo (**PCG**) [4]. En la sección 3 se presenta el algoritmo gradiente conjugado proyectado precondicionado. En la sección 4 se presentan los resultados numéricos al resolver problemas de la librería CUTEr usando el algoritmo **PCG** es sus versiones sin precondicionar y precondicionado. En la sección 5 se proponen algunas líneas de investigación que se pueden estudiar. Las conclusiones se hacen en la sección 6.

2. Gradiente Conjugado Proyectado (PCG)

El objetivo de esta sección es presentar el método Gradiente Conjugado Proyectado para resolver problemas de minimización cuadrática con restricciones de igualdad propuesto en [4].

Consideremos el problema (2). Si A es de rango completo, entonces las columnas de A^T son linealmente independientes y generan un subespacio de dimensión m de $\mathbb{R}^n$. Si consideramos ahora Z, una base del espacio nulo de A, dicho subespacio tendrá dimensión $n-m$ y $[A^T Z]$ forma una base de $\mathbb{R}^n$.

Supongamos ahora que x es la solución de (1), entonces,

$$x = A^T x^* + Zw. \tag{3}$$

De la ecuación (3) podemos concluir que x es la suma de dos vectores; además, multiplicando a la expresión anterior por A, se tiene que:

$$b = Ax = A(A^T x^*) + A(Zw) = A(A^T x^*) + (AZ)w = A(A^T)x^* \tag{4}$$

es decir, $x_p = A^T x^*$ es una solución particular de $Ax = b$ y $x_h = Zw \in N(A)$, donde $N(A)$ es el espacio nulo de A. Así, la solución x puede expresarse como:

$$x = x_p + x_h. \tag{5}$$

Por otra parte, como A es de rango completo con $m < n$, se tiene que AA^T es simétrica definida positiva, por lo tanto, es invertible y así de la expresión (4) se obtiene que:

$$\begin{aligned} x^* &= (AA^T)^{-1}b \\ x_p &= A^T(AA^T)^{-1}b \\ x_h &= (I - A^T(AA^T)^{-1}A)w \quad \text{para } w \in \mathbb{R}^n. \end{aligned}$$

Observe que $P = I - A^T(AA^T)^{-1}A$, es una proyección ortogonal sobre el espacio nulo de A.

Una vez que se halla x_p, se puede resolver el sistema aumentado (2) en términos de x_h e y. En efecto, sustituyendo (5), junto con las expresiones obtenidas para x_p y x_h en (2) y realizando algunas manipulaciones algebraícas, el sistema aumentado se reduce a la siguiente ecuación:

$$Gx_h + A^T y = -(g + Gx_p). \tag{6}$$

Resolviendo para y, se obtiene que:

$$y = -(AA^T)^{-1}Ag - (AA^T)^{-1}AGx_p - (AA^T)^{-1}AGx_h. \tag{7}$$

Luego, sustituyendo (7) en (6) se deduce la siguiente ecuación proyectada:

$$PGx_h = -P(Gx_p + g)$$

donde $P = I - A^T(AA^T)^{-1}A$. Finalmente, usando P en la expresión anterior, ésta puede ser reescrita como:

$$PGPw = -P(Gx_p + g) \tag{8}$$

donde $Pw = x_h$ para algun $w \in \mathbb{R}^n$.

En [4] se demuestra que la ecuación proyectada (8) tiene una única solución en el espacio nulo de A y ésta es la solución de norma mínima; además, como G es simétrica definida positiva sobre el espacio nulo de A, se propone obtener dicha solución a usando el método de gradiente conjugado [11] para el sistema (8) con iterado inicial cualquier vector en el espacio nulo de A. Una de las ventajas de este procedimiento es que las soluciones aproximadas, las direcciones y los vectores residuales permanecen en el espacio nulo de A. Más aún, en aritmética exacta, el algoritmo encuentra la solución en a lo sumo $n - m$ iteraciones puesto que PGP tiene máximo $n - m$ autovalores diferentes de cero.

El algoritmo **Gradiente Conjugado Proyectado (PCG)** se presenta en **Algoritmo 1**.

3. Precondicionamiento

Un procedimiento para acelerar la velocidad de convergencia del método iterativo **PCG** es mejorar la distribución de los autovalores de la matriz

Algoritmo 1 Algoritmo Gradiente Conjugado Proyectado. **PCG**

Entrada: $G, A, b, g, \varepsilon, maxiter$

Salida: x, y

$x = A^T(AA^T)^{-1}b$

$f = Gx + g$

$y = -(AA^T)^{-1}Af$

$r = -(f + A^T y)$

si $\|r\| < \varepsilon$ **entonces**

  devolver x, y

  alto, terminar programa

fin si

$d = r$

$\beta_n = r^T r$

para $i = 1$ hasta $maxiter$ **hacer**

  $\alpha_d = d^T G d$

  $x = x + \frac{\beta_n}{\alpha_d} d$

  $f = Gx + g$

  $y = -(AA^T)^{-1}Af$

  $r = -(f + A^T y)$

  si $\|r\| < \varepsilon$ **entonces**

    devolver x, y

    alto, terminar programa

  fin si

  $\beta_d = \beta_n$

  $\beta_n = r^T r$

  $d = r + \frac{\beta_n}{\beta_d} d$

fin para

devolver x, y

PGP o reducir su número de condición [8, 9]. Además, los precondicionadores deben ser seleccionados de tal manera que se explote la estructura y la esparcidad de las matrices involucradas en el problema.

Argaéz [4] demostró que PMP es un buen precondicionador para el sistema (8) si M es una buena aproximación para la matriz G. La demostración se basó en aproximar a la inversa Moore-Penrose de la matriz PGP del sistema (8) por la inversa de Moore-Penrose para la matriz PMP. Definiendo $N = M - G$, se demuestra que mientras mayor reducción se logre en $\|N\|$ se mejora la calidad del precondicionador PMP para el sistema (8) y esto también implica una mejora de calidad de M como un preconditioner para G; por lo tanto, las propiedades y estructura de G son importantes para la elección de precondicionadores adecuados para el algoritmo de **PCG**.

El algoritmo Gradiente Conjugado Proyectado Precondicionado (**PPCG**) se presenta en **Algoritmo 2**.

En [4], se proponen como aproximaciones para G las siguientes:

1. $M = D + \tau$, donde D es la matriz diagonal de G y τ es un escalar dado.

2. $M = \frac{1}{\omega(2-\omega)}(D + \omega L)D^{-1}(D + \omega L^T)$, donde D y L son la matriz diagonal y la parte triangular estricta de G, respectivamente. El escalar ω es el paraámetro de sobrerelajación del método de SOR y debe ser seleccionado de tal manera que $0 < \omega < 2$.

Aún cuando los resultados obtenidos con estos dos precondicionadores en [4] fueron bastante satisfactorios, ellos son bastante básicos. En este trabajo proponemos utilizar como precondicionador para G a la matriz $M = \tilde{L}\tilde{L}^T$, donde $\tilde{L}$ es el factor de Cholesky incompleto para la matriz G con el mismo patrón de esparcidad que la parte inferior de G.

4. Resultados Numericos

Los experimentos numéricos fueron implementados en una computadora Intel(R) Core(TM) i3-2330M CPU @ 2.20GHz 2.20 GHz, con 2.70 GB de memoria Ram. Los algoritmos fueron implementados en MatLab R2008b. Se usó como criterio de parada $|Gx + A^T y + g| < 1e - 6$ y el tiempo es medido en segundos. Los problemas seleccionados para las pruebas fueron obtenidos de la colección CUTEr [10].

En la tabla (1) se comparan los tiempos de convergencia y el número de iteraciones para algunos problemas de la colección CUTEr. El precondicionador utilizado para el algoritmo (2) fue $M = \tilde{L}\tilde{L}^T$, donde $\tilde{L}$ es el factor de

Algoritmo 2 Algoritmo Gradiente Conjugado Proyectado Precondicionado. **PPCG**

Entrada: $G, A, b, g, \varepsilon, maxiter, M$

Salida: x, y

$x = A^T(AA^T)^{-1}b$
$f = Gx + g$
$y = -(AA^T)^{-1}Af$
$r = -(f + A^Ty)$
si $||r|| < \varepsilon$ **entonces**
  devolver x, y
  alto, terminar programa
fin si
Resolver para w, $Mw = r$
$r_{prec} = Pw$
$d = r_{prec}$
$\beta_n = r^T r_{prec}$
para $i = 1$ hasta $maxiter$ **hacer**
  $\alpha_d = d^T G d$
  $x = x + \frac{\beta_n}{\alpha_d} d$
  $f = Gx + g$
  $y = -(AA^T)^{-1}Af$
  $r = -(f + A^Ty)$
  si $||r|| < \varepsilon$ **entonces**
    devolver x, y
    alto, terminar programa
  fin si
  Resolver para w, $Mw = r$
  $r_{prec} = Pw$
  $\beta_d = \beta_n$
  $\beta_n = r^T r_{prec}$
  $d = r + \frac{\beta_n}{\beta_d} d$
fin para
devolver x, y

			PCG		PPCG		
Problema	m	n	it	t seg.	it	t seg.	$\|\|Gx + A^T y + g\|\|$
DUAL1	1	85	141	0.044	2	0.004	8.24e-07
DUAL2	1	96	66	0.024	2	0.004	6.55e-07
DUAL3	1	111	56	0.022	2	0.005	7.06e-07
DUAL4	1	75	29	0.011	2	0.003	7.52e-07
GENHS28	4900	5000	100	0.434	3	0.025	2.97e-11
MOSARQP1	700	2500	12	0.013	5	0.010	4.50e-07
MOSARQP2	600	900	17	0.018	6	0.011	7.38e-07
STCQP2	2052	4097	111	0.306	44	0.169	9.39e-07

Cuadro 1: Tamaño de los bloques, número de iteraciones requeridas por PCG y PPCG para satisfacer el criterio de parada, tiempo de CPU y norma de los residuales $||Gx + A^T y + g||$ para los diferentes problemas de prueba

Cholesky incompleto para G con con el mismo patron de esparcidad que la parte inferior de G. En los resultados mostrados en la tabla (1) se observa que se reduce de CPU desde un 24 % para el problema MOSARQP1 hasta un 91 % para el problema DUAL1, cuando se resuelve el sistema usando el algoritmo **PPCG** en lugar de **PCG**.

5. Líneas de Investigación

A pesar de que el **PCG** tiene una buena rata de convergencia, $n - m$ iteraciones, tiene el inconveniente de almacenar la matriz asociada a las restricciones de igualdad. En algunas aplicaciones el tamaño de la matriz A es tan grande que supera la capacidad de almacenamiento del hardware o la matriz A no se conoce explícitamente, como en el caso de procesamiento de señales. Por lo tanto proponemos resolver los problemas de mínimos cuadrados asociados al problema usando métodos iterativos, como por ejemplo, **CG**. Como el almacenamiento de A ya no es un obstaculo, podemos usar $(PMP)^\dagger$ como precondicionador con una matriz M los más cercana a G.

6. Conclusiones

Basados en los resultados numéricos presentados en la tabla (1) observamos que precondicionando con una proyección ortogonal usando una factorización incompleta de Choleski de G, del algoritmo **(PPCG)** mejora de manera significativa los resultados obtenidos del algoritmo **(PCG)**.

Referencias

[1] M. Argaez, H. Klie, M. Rame, and R. A. Tapia, *An interior point Krylov-orthogonal projection method for nonlinear programming.* Talk presented at the Fifth SIAM Conference on Optimization, Victoria, Canada, 1996.

[2] M. Argaez, H. Klie, M. Rame, and R. A. Tapia, *An interior point Krylov-orthogonal projection method for nonlinear programming.* TR97-16, Computational and Applied Mathematics, Rice University, Houston Texas, 1997.

[3] M. Argaez, *Exact and Inexact Newton Linesearch Interior-Point Algorithms for Nonlinear Programming Problems.* PHD Thesis, Computational and Applied Mathematics, Rice University, Houston Texas, 1997.

[4] M. Argáez, *A projected conjugate gradient algorithm for KKT systems*, Tech. Rep.TR01-2010. Mathematical Sciences Dept. University of Texas at El Paso.

[5] M. Benzi, G. H. Golub and J. Liesen, *Numerical solution of saddle point problems*, Acta Numerica, Vol. 14, pp. 1-137, 2005.

[6] D. P. Bertsekas, *Nonlinear Programming*, Athena Scientific, 2th, 1999.

[7] Z. Cao, *Fast Uzawa algorithm for generalized saddle point problems*, Appl. Numer. Math., Vol. 46, No. 2, pp. 157-171, 2003.

[8] D. De Cecchis, H. López and B. Molina, *Técnicas de Precondicionamiento de los Métodos de Krylov para la Discretización del Problema Stokes Generalizado*, Proceedings VIII International Congress on Numerical Methods in Engineering and Applied Sciences, pp. TM1-TM7, 2006.

[9] D. De Cecchis, H. López and B. Molina, *FGMRES Preconditioning by Symmetric/skew-symmetric Decomposition of Generalized Stokes Problems*, Mathematics and Computers in Simulation, Vol. 79, No. 6, pp. 1862-1877, 2009.

[10] N. I. M. Gould, D. Orban and Ph. L. Toint, *CUTEr (and SifDec), a Constrained and Unconstrained Testing Environment*, Revisited, Tech. Report TR/PA/01/04, CERFACS, Toulouse, France, 2001.

[11] M.R. Hestenes and E.L. Stiefel, *Methods of conjugate gradients for solving linear systems*, J. Res. Nat. Bur. Stand., Vol. 49, pp. 409-436, 1952.

[12] N.I.M. Gould, M.E. Hribar, and Nocedal. *On the solution of equality constrained quadratic programing problems arising in optimization.* SIAM J. Sci Computing, Vol 23, 4, pp. 1375-1394, 2001.

[13] Y. Saad, *Iterative Methods for Sparse Linear Systems*, SIAM, 2nd ed., Philadelphia, PA, 2003.

[14] H. Uzawa, *Iterative methods for concave programming*, Studies in Linear and Nonlinear Programming, K. J. Arrow and L. Hurwicz and H. Uzawa (Eds), Stanford University Press, pp. 154-165, Stanford, CA, 1958.

[15] M. H. Wright, *Interior point methods for constrained optimization*, Acta Numerica, Vol. 1, Cambridge University Press, pp. 341-407, 1992.

[16] S. J. Wright, *Primal Dual Interior Point Methods*, SIAM, Philadelphia, PA, 1997. N. I. M. GOULD, D. ORBAN, AND PH. L. TOINT, CUTEr (and SifDec), a Constrained and Unconstrained Testing Environment, Revisited, Tech. Report TR/PA/01/04, CERFACS, Toulouse, France, 2001.