

Diferentes técnicas de preconditionamiento para el método PPCG en optimización con restricciones

Leobardo Valera ¹ Miguel Argáez ² Brígida Molina ³

¹ Escuela de Matemáticas, Universidad Metropolitana, Caracas, Venezuela.
lvalera@unimet.edu.ve

² Department of mathematical Sciences at University of Texas at El paso.
margaez@utep.edu

³ Centro de Cálculo Científico y Tecnológico (CCCT), Escuela de Computación, Facultad de Ciencias, Universidad Central de Venezuela.
Brigida.molina@ciens.ucv.ve

Resumen

Problemas de optimización con restricciones de igualdad surgen en muchas y variadas aplicaciones, entre otras, en economía, en la solución de ecuaciones diferenciales parciales para materiales incomprensibles, en la recuperación de señales de entrada. Un problema de optimización cuadrática con restricciones tiene la forma:

$$\min_x q(x) = \frac{1}{2} x^T G x + g^T x \quad (1)$$

$$\text{s.a. } Ax = b \quad (2)$$

donde $G \in \mathbb{R}^{n \times n}$ es una matriz simétrica, A es una matriz de rango full con $m < n$, $x, y \in \mathbb{R}^n$ y $b \in \mathbb{R}^m$. Es bien conocido que si A es una matriz de rango full y G es simétrica positiva sobre el espacio nulo de A , es decir, $Z^T G Z$ es simétrica definida positiva donde Z es una matriz cuyas columnas son una base para el espacio nulo de A , entonces (1) tiene una única solución global y puede ser obtenida al resolver el sistema aumentado

$$\begin{pmatrix} G & A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} -g \\ b \end{pmatrix} \quad (3)$$

donde $x \in \mathbb{R}^n$ es la solución de (1) y $y \in \mathbb{R}^m$ son los multiplicadores de Lagrange.

La solución x puede ser escrita como la suma de dos vectores ortogonales y complementarios en \mathbb{R}^n

$$x = x_p + x_h$$

donde x_p está en el rango de A^T y está dado por $x_p = A^T (AA^T)^{-1}b$ y x_h es un vector en el espacio nulo de A dado por

$$x_h(w) = (I - A^T (AA^T)^{-1}A)w$$

para algún $w \in \mathbb{R}^n$. La expresión $I - A^T (AA^T)^{-1}A$ es denotado por P y es una proyección ortogonal sobre el espacio nulo de A .

Después de encontrar x_p , x_h puede ser encontrado resolviendo el sistema de ecuaciones

$$(PGP)w = -P(Gx_p + g) \quad (4)$$

para algún $w \in \mathbb{R}^n$. Dicho sistema tiene infinitas soluciones ya que PGP es singular, pero existe un único w_0 en el espacio nulo de A y es la solución de norma mínima.

En [1], se propone el método de Gradiente Conjugado Proyectado (PCG) para resolver este tipo de problemas de manera eficiente. Sin embargo, la eficiencia y robustez de las técnicas iterativas pueden ser mejoradas utilizando técnicas de preconditionamiento. Precondicionar es simplemente una forma de transformar el sistema lineal original por otro que tiene la misma solución, pero el cual es probablemente más fácil de resolver con el método iterativo propuesto. En [2], se plantea el método de Gradiente Conjugado Proyectado Precondicionado (PPCG). Sin embargo, el tipo de preconditionadores propuesto son bastante sencillos; se proponen dos tipos de preconditionadores: diagonales y los basados en SSOR.

En este presente trabajo se estudian diferentes técnicas de preconditionamiento para el método PPCG. Entre otros, se utilizan preconditionadores en bloques triangulares y preconditionadores basados en la descomposición simétrica-antisimétrica [3, 4]. Se presentan resultados numéricos preliminares para mostrar la efectividad del PPCG.

Resultados numéricos preliminares

Notación

PCG-Sin	Método Gradiente Conjugado Proyectado sin preconditionar
PCG-Diag	Método Gradiente Conjugado Proyectado preconditionado con precondicionador diagonal M_1
PCG-Sor	Método Gradiente Conjugado Proyectado preconditionado con preconditionador basado en SSOR M_2
Gmres(100)	Método GMRES con RESTART 100 sin preconditionar
Gmres(100)-diag	Método GMRES con RESTART 100 preconditionado con precondicionador en forma de bloque M_3
Gmres(100)-sor	Método GMRES con RESTART 100 preconditionado con precondicionador en forma de bloque M_4

$$M_1 = D + \tau$$

donde D es la diagonal de G y $\tau = 0,2$

$$M_2 = \frac{1}{w(2-w)} (D + wL)D^{-1}(D + wL^T)$$

donde D y L son respectivamente la diagonal y la parte triangular superior estricta de G .

El método GMRES(100) se utiliza para resolver el sistema aumentado

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

los preconditionadores que se utiliza en este caso son preconditionadores de bloques

$$M_3 = \begin{pmatrix} M_1 & A^T \\ -A & 0 \end{pmatrix}$$

$$M_4 = \begin{pmatrix} M_2 & A^T \\ -A & 0 \end{pmatrix}$$

donde M_1 y M_2 son las matrices descritas anteriormente.

En la tabla siguiente se destacan de color azul aquellos resultados favorables que se hayan obtenido con un menor número de iteraciones y/o en menor tiempo.

CVXQP3					
Método	Iteraciones	$\ Ax - b\ $	$\ Gx + A^T y + g\ $	cputime	$f(x)$
PCG-Sin	219	$2,9215 \times 10^{-7}$	$1,1665 \times 10^{-5}$	276	$4,38 \times 10^8$
PCG-Diag	220	$6,0087 \times 10^{-9}$	0,6145	366	$4,38 \times 10^8$
PCG-Sor	220	$1,8065 \times 10^{-8}$	145	555	$4,38 \times 10^8$
Gmres(100)	5000	726.06	726.67	436	$1,11 \times 10^4$
Gmres(100)-diag	191	$1,1606 \times 10^{-7}$	0.0027	193	$4,38 \times 10^8$
Gmres(100)-sor					
AUG2D					
Método	Iteraciones	$\ Ax - b\ $	$\ Gx + A^T y + g\ $	cputime	$f(x)$
PCG-Sin	13	$1,3472 \times 10^{-10}$	$1,0274 \times 10^{-5}$	0.15	$3,90 \times 10^4$
PCG-Diag	6	$5,3537 \times 10^{-12}$	$5,0012 \times 10^{-5}$	0.13	$3,90 \times 10^4$
PCG-Sor					
Gmres(100)	265	$2,8448 \times 10^{-6}$	$3,2742 \times 10^{-6}$	1.52	$3,9005 \times 10^4$
Gmres(100)-diag	7	$9,3854 \times 10^{-9}$	$1,9473 \times 10^{-5}$	0.86	$4,06 \times 10^4$
Gmres(100)-sor					
AUG3D					
Método	Iteraciones	$\ Ax - b\ $	$\ Gx + A^T y + g\ $	cputime	$f(x)$
PCG-Sin	13	$3,037 \times 10^{-10}$	$7,3543 \times 10^{-6}$	38	$1,319 \times 10^4$
PCG-Diag	6	$1,3573 \times 10^{-12}$	$6,7670 \times 10^{-5}$	38	$1,3190 \times 10^4$
PCG-Sor					
Gmres(100)	240	$1,1052 \times 10^{-8}$	$1,7206 \times 10^{-8}$	18	$2,46 \times 10^4$
Gmres(100)-diag	10	$1,0207 \times 10^{-11}$	$2,7617 \times 10^{-8}$	36	$2,46 \times 10^4$
Gmres(100)-sor					
GENHS28					
Método	Iteraciones	$\ Ax - b\ $	$\ Gx + A^T y + g\ $	cputime	$f(x)$
PCG-Sin	100	$1,0699 \times 10^{-11}$	$3,6081 \times 10^{-11}$	120	$5,44 \times 10^2$
PCG-Diag					
PCG-Sor	7	$1,5749 \times 10^{-14}$	$4,6027 \times 10^{-11}$	117	$5,44 \times 10^2$
Gmres(100)	5000	$8,3227 \times 10^{-9}$	$2,3110 \times 10^{-7}$	58	$5,44 \times 10^2$
Gmres(100)-diag	301	$3,1070 \times 10^{-14}$	$1,6134 \times 10^{-12}$	17	$5,44 \times 10^2$
Gmres(100)-sor	10	$9,9207 \times 10^{-7}$	1.0035	1.34	$5,44 \times 10^2$

- [1] M. Argáez, “Solving overdetermined systems in lp quasi-norms”. Special SCAN '08 Issue of Reliable Computing, 2010.
- [2] M. Argáez, “A projected conjugate gradient algorithm for KKT systems”, tech. Rep. TR01-2010, Mathematical Sciences Dept., University of Texas at El Paso.
- [3] M. Benzi and G. H. Golub, “A preconditioner for generalized saddle point problems”, SIAM J. Matrix Anal. Appl., 26(1):20–41, 2004.
- [4] 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 79:1862–1877, 2009.