UNIVERSIDAD CENTRAL DE VENEZUELA
FACULTAD DE CIENCIAS
ESCUELA DE MATEMÁTICA

Aceleración del Método de Cimmino por Gradientes Conjugados

Trabajo Especial de Grado presentado ante la ilustre Universidad Central de Venezuela por el **Br. Roberto José Morillo Castro** para optar al título de Licenciado en Matemática.

Tutor: Prof. Luis Manuel Hernández

Caracas, Venezuela
12 de Febrero del 2009

Nosotros, los abajo firmantes, designados por la Universidad Central de Venezuela como integrantes del Jurado Examinador del Trabajo Especial de Grado titulado "**Aceleración del Método de Cimmino por Gradientes Conjugados**", presentado por el **Br. Roberto José Morillo Castro**, titular de la Cédula de Identidad **17.475.920**, certificamos que este trabajo cumple con los requisitos exigidos por nuestra Magna Casa de Estudios para optar al título de **Licenciado en Matemática**.

Prof. Luis Manuel Hernández R.
Tutor

Dr. José Luis Sánchez
Jurado

Dr. Marcos Raydan
Jurado

ÍNDICE GENERAL

Introducción **6**

1 Preliminares **9**

1.1 Operadores en Espacio de Hilbert 9

1.2 Ángulo entre subespacios 16

2 Método de las Proyecciones Alternantes **20**

2.1 Introducción 20

2.2 Proyecciones Alternantes 20

2.2.1 Velocidad de Convergencia 22

2.3 Método de Kaczmarz 24

2.4 Método de Cimmino 25

3 Método de los Gradientes Conjugados **30**

3.1 Introducción 30

3.2 Descripción del método 30

3.3 Caso Cuadrático 31

3.3.1 Gradientes Conjugados como Método Iterativo 36

4 Aceleración del método de Cimmino **39**

4.1 El Método de Cimmino como método de Optimización 39

4.2 Aceleración del Método de Cimmino 42

4.3 Método de Cimmino Acelerado para problemas de Saddle Point (Punto de Ensilladura) 44

5 Experimentación Numérica **48**

5.1 Resultados 52

5.1.1 Experimento 1 52

5.1.2 Experimento 2 61

5.2 Comparación de los métodos 69

Conclusiones y Recomendaciones **70**

Bibliografía **72**

Introducción

Los métodos basados en proyecciones, han sido utilizados ampliamente en la literatura para la resolución de problemas de factibilidad y mejor aproximación. Entre las aplicaciones podemos encontrar: resolución de sistemas de ecuaciones lineales, tratamiento de imágenes, algunas nociones de teoría de predicción en el campo de la probabilidad y estadística , etc. Entre estos métodos basados en proyecciones, destaca el método de las Proyecciones Alternantes de Von Neumann y la generalización del mismo realizada por Halperin [8], y el método de Cimmino [2]. Sin embargo, ambos métodos pudieran presentar lenta velocidad de convergencia, en particular, cuando el ángulo entre los subespacios involucrados es pequeño.

Para estos métodos se han desarrollado diversos esquemas de aceleración, el más conocido es el esquema de aceleración de Koshy-Gearhart [13] para el método de Von Neumann. Actualmente, se han desarrollado esquemas de aceleración viendo a estos métodos como métodos para minimizar una cierta funcional cuadrática.

En trabajos recientes, se han desarrollado versiones aceleradas del método de Von Neumann [15],[23] y del método de Cimmino basada en una escogencia espectral [11] del tamaño del paso.

En este trabajo, proponemos una versión acelerada del método de Cimmino basada en el método de los Gradientes Conjugados. Esta versión que desarrollaremos conserva la característica de paralelización, que es la ventaja principal del método de Cimmino esperando que su velocidad de convergencia sea notablemente superior.

Por último, se desarrollaran experimentos numéricos destinados a comparar el desempeño del esquema propuesto respecto al método de Cimmino clásico y su aceleración mediante Barzilai-Borwein [1], [11]. Igualmente se harán experimentos para examinar la velocidad de convergencia en función del número de bloques al resolver problemas tipo punto de ensilladura (Saddle Point) que son importantes en la literatura.

CAPÍTULO 1

Preliminares

En este capítulo se presentan algunas definiciones, teoremas y proposiciones importantes que serán de gran utilidad en los capítulos posteriores, las mismas pueden ser encontradas con más detalles en [4], [7], [8].

1.1 Operadores en Espacio de Hilbert

Definición 1.1. *Sea X un espacio Vectorial sobre $\mathbb{K}$ donde ($\mathbb{K}$ es $\mathbb{R}$ o $\mathbb{C}$). Un producto interno en X es una función $\langle .,. \rangle : X \times Y \to \mathbb{K}$ tal que:*

(i) $\langle x, y \rangle = \overline{\langle y, x \rangle}$ *para todo* $x, y \in X$

(ii) $\langle x + y, z \rangle = \langle x, z \rangle + \langle y, z \rangle$ *para todo* $x, y, z \in X$

(iii) $\langle \lambda x, y \rangle = \langle x, \lambda y \rangle = \lambda \langle x, y \rangle$ *para todo* $x, y \in X$, *para todo* $y \in \mathbb{K}$

(iv) $\langle x, x \rangle \geq 0$ *para todo* $x \in X$

(v) $\langle x, x \rangle = 0$ *si y solo si* $x = 0$

Definición 1.2. *Un Espacio de Hilbert H, es un espacio con producto interno que es completo con respecto a la norma dada por el producto interno.*

Definición 1.3. *Dado un espacio de Hilbert H, y $M \subset H$, definiremos al complemento ortogonal de M, como*

$$M^{\perp} := \{y \in H | \langle x, y \rangle = 0 \ \forall \ x \in M\}$$

Proposición 1.4. *Sean H un espacio de Hilbert y $M \subset H$ entonces $M^{\perp}$ es un subespacio cerrado de H.*

Teorema 1.5. *Sea H un espacio de Hilbert. Todo subconjunto cerrado, convexo y no vacío de H contiene un único elemento de norma mínima.*

Teorema 1.6. *Sea H un espacio de Hilbert y M un subespacio cerrado de H entonces para cada $x \in H$ existe un único $y \in M$ y un único $z \in M^{\perp}$ tal que $x = y + z$.*

Corolario 1.7. *Sean H un espacio de Hilbert y M un subespacio cerrado de H entonces*

$$H = M \oplus M^{\perp}$$

donde $M^{\perp}$ es un subespacio cerrado de H, que es ortogonal a M.

Observación 1.8. *Las demostraciones de los resultados anteriormente enunciados, pueden ser encontradas con detalles en [4].*

Sea H un espacio de Hilbert

Definición 1.9. *Un operador en H es una función T la cual es definida sobre algún subconjunto S de H que tiene uno o mas valores T_x en H correspondientes a cada elemento x de S.*

Llamaremos S al **Dominio** $D(T)$ y llamaremos al conjunto de todos los valores de T_x, $x \in D(T)$ el **Rango** de T. Además al conjunto de todos los elementos (x, T_x) con $x \in D(T)$ y $T_x \in R(T)$ lo llamaremos **Gráfico** $G(T)$ de T.

Definición 1.10. *Sea $T : X \longrightarrow Y$ un operador, donde X, Y son Espacios Normados, T es Acotado si existe un número M tal que $\| T_x \| \leq M \| x \| \; \forall \; x \in X$. Si T es acotado, se define la norma inducida como $\| T \| = \sup\{\frac{\|T_x\|}{\|x\|} : x \in X, x \neq 0\}$.*

Ahora, nos interesara considerar el operador Proyección. Recordemos que si M es un subespacio cerrado en H, el vector $m_0 \in H$ tal que $x - m_0 \in M^{\perp}$ es llamado Proyección Ortogonal de x sobre M, y se denota $P_M(x) = m_0$.

Teorema 1.11. *Un operador T es una Proyección P_S si y solo si:*

(i) *T es S.V, Lineal, Acotado y $D(T) = H$*

(ii) *$\langle T_x, x \rangle = \langle x, T_y \rangle \; \forall \; x, \; y \in H$ (T es **Autoadjunto**)*

(iii) *$T^2 = T$, donde $T^2 \equiv TT$; es decir, T es **idempotente**. Además, S es determinado únicamente por T ($R(T) = S$)*

Demostración. Esta Demostración puede ser revisada de forma más detallada en [10]

$\Longrightarrow$)

(i) **Linealidad**

Sea x,$y \in H$, así $x = x_1 + x_2$ y $y = y_1 + y_2$, donde $x_1, y_1 \in S$ y $x_2, y_2 \in S^{\perp}$, Luego $x + y = (x_1 + y_1) + (x_2 + y_2)$ es la descomposición ortogonal de $x + y$.

Por otro lado, $P_S(x + y) = x_1 + y_1 = P_S(x) + P_S(y)$, pero además para cualquier escalar α,

$$\alpha x = \alpha x_1 + \alpha x_2$$

y

$$P_S = \alpha P_S(x)$$

con lo que queda demostrado que P_s es Lineal.

Acotamiento

Como H es un espacio de Hilbert, tenemos que:

$$\| x \|^2 = \| x_1 \|^2 + \| x_2 \|^2$$

luego,

$$\| P_S(x) \|^2 = \| x \|^2 - \| x_2 \|^2 \leq \| x \|^2$$

con lo que obtenemos que P_S es Acotado.

S.V

Esto es inmediato debido a la única descomposición ortogonal de $x \in H$, H espacio de Hilbert y la definición de proyección ortogonal de x sobre S (Subespacio Cerrado).

El hecho que $D(T) = H$, es inmediato por la definición de proyección ortogonal.

(ii) **Auto Adjunto**

Consideremos

$$\begin{aligned} < x, P_S(y) > &= < x_1 + x_2, y_1 > \\ &= < x_1, y_1 > \\ &= < x_1, y_1 + y_2 > \\ &= < P_S(x), y > \end{aligned}$$

De donde obtenemos que P_S es Auto Adjunto para algún $x \in H$.

(iii) **Idempotencia**

Para algún $x \in H$, $P_S(x) \in H$, tenemos que $P_S(P_S(x)) = P_S(x)$ de donde obtenemos $P_S^2 = P_S$, Ahora solo nos falta ver la unicidad de S, para ello supongamos existe un subespacio cerrado S tal que para algún $x \in H$, $T_x = P_S(x)$, pero para esto, existen dos posibilidades:

(a) $x \in S$

(b) $x \notin S$

Si $x \in S$, $T_x = P_S(x) = x \in R(T)$.

Si $x \notin S$, $T_x = P_S(x) \neq x$. Así $S = x \in H : T_x = x$ y $S \subset R(T)$. Ahora sea x algún elemento de H, y sea $T_x = y \in R(T)$, luego $T_x^2 = T_y$, y como $T^2 = T$, tenemos $T_x = T_y$ de donde obtenemos que $T_y = y$ de donde se obtiene que $R(T) \subset S$.
De donde $S = R(T)$ lo cual prueba la unicidad.

$\Longleftarrow$)

Sea $S = x \in H : T_x = x$. Por la parte anterior (Unicidad) sabemos que $S = R(T)$ y además sabemos T es lineal, luego S es un subespacio lineal de H.

Ahora veamos que S es cerrado, para esto debemos ver que T es continuo. Si $\| T_x \| = 0$, obviamente $\| T_x \| \leq \| x \|$. Supongamos ahora que $\| T_x \| \neq 0$, Por la desigualdad de Cauchy Schwartz

$$\begin{aligned} \|T_x^2\| &= |\langle T_x, T_x \rangle| \\ &= |\langle T_x^2, x \rangle| \\ &= \langle T_x, x \rangle \\ &\leq \| T_x \| \| x \| \; \forall x \in H \end{aligned}$$

Luego $\| T_x \| \leq \| x \|$. Pero esta relación es valida para todo $x \in H$, tomando algún $x, y \in H$

$$\|T_{(x-y)}\| = \|T_x - T_y\| \leq \|x - y\|$$

De donde obtenemos que T es continuo en H y S es Cerrado, en consecuencia S es un subespacio cerrado, con lo que hemos mostrado que $T = P_s$.

Consideremos $S = R(T) = T_x = T_x : x \in H$, ahora veremos que T aplicado sobre S es la identidad y T aplicado sobre $S^{\perp}$ es el operador 0, y así $T = P_S$.
Si $y \in S$, tenemos que $y = T_x$, para algún $x \in H$. Por otro lado

$$T_y = T_x^2 = T_x = y$$

Por lo tanto $T_y = y$ si $y \in S$. Ahora si $z \in S^{\perp}$, tenemos que para algún $u \in H$

$$< u, T_z >=< T_u, z >= 0$$

de donde obtenemos que $T_u \in S$. Por lo que $T_z = 0$ si $z \in S^{\perp}$.
En consecuencia a esto, parta algún $x \in H$, podemos escribir

$$x = y + z$$

con $y \in S$ y $z \in S^{\perp}$, así

$$T_x = T_y + T_z = y$$

De donde obtenemos que T es la proyección sobre S. □

El siguiente teorema muestra condiciones necesarias para que el producto de dos operadores sea una proyección.

Teorema 1.12.

(i) Si U y V son proyecciones en H. Entonces una condición suficiente y necesaria para que UV sea una proyección es que $UV = VU$

(ii) Si $U = P_M$ y $V = P_N$ (M y N son subespacios cerrados), Entonces $UV = P_{M \cap N}$

Demostración. Ver [10]. □

Otras consecuencias, no tan directas de los dos teoremas anteriores, se encuentran en los siguientes lemas, cuyas demostraciones no se incluyen.
Para visualizar detalles de interés respecto a las mismas ver [9].

Lema 1.13. *Sea P un operador Lineal y Acotado en un espacio de Hilbert H, P es una proyección ortogonal, si y sólo si $P^2 = P$ y $P^* = P$, donde $P = P_M$ y $M = P(H)$.*

Lema 1.14. *Las siguientes afirmaciones, son equivalentes:*

(a) P_M *y* P_N *conmutan.* ($P_M P_N = P_N P_M$)

(b) $P_M P_N = P_{M \cap N}$

(c) $P_M P_N$ *es una proyección ortogonal.*

Lema 1.15. *Las siguientes afirmaciones, son equivalentes:*

(a) $P_M P_N = 0$

(b) $P_N P_M = 0$

(c) $M \perp N$, (i.e $\langle x, y \rangle = 0, \ \forall \ x \in M, \ y \in N$).

Lema 1.16. *Las siguientes afirmaciones, son equivalentes:*

(a) $P_M P_N = P_M$

(b) $P_N P_M = P_M$

(c) $M \subset N$

En particular, P_M *y* P_N *conmutan con* $P_{M \cap N}$ *si* $P_M P_{M \cap N} = P_{M \cap N} = P_N P_{M \cap N}$.

Lema 1.17. *Las siguientes condiciones son equivalentes:*

1. P_M *conmuta con* P_N
2. $P_{M^\perp}$ *conmuta con* P_N
3. P_M *conmuta con* P_N
4. P_M *conmuta con* $P_{N^\perp}$
5. $M = M \cap N + M \cap N^\perp$

1.2 Ángulo entre subespacios

Existen varias maneras de definir la noción de ángulo entre dos subespacios, recordemos que para vectores $u, v \in H$, H espacio de Hilbert, el ángulo definido entre ambos viene dado por la expresión

$$cos\theta = \frac{< u, v >}{||u||_2||v||_2}$$

En el caso de subespacios, las fórmulas más utilizadas para definir la noción de ángulo, son la definición propuesta por Friedricks (1.18) y la definición propuesta por Dixmier (1.19).

Consideraremos H un espacio de Hilbert y M, N dos subespacios cerrados del mismo, la norma definida por el producto interno como $||x|| = \sqrt{< x, x >}$.

Definición 1.18. *(Friedricks, 1939) [9] El ángulo* $\theta(M, N)$ *entre dos subespacios cerrados* M *y* N *de* H*, es el ángulo en* $[0, \pi/2]$ *cuyo coseno* $C(M, N) = \cos\theta(M, N)$ *esta dado por:*

$$sup\{|\langle x, y\rangle| : x \in M \cap (M \cap N)^\perp, \parallel x \parallel \leq 1,\ y \in N \cap (N \cap M)^\perp, \parallel y \parallel \leq 1\}$$

Definición 1.19. *(Dixmier, 1947)[9] El mínimo ángulo $\theta_0(M,N)$ entre dos subespacios cerrados M y N de H, es el ángulo en $[0,\pi/2]$ cuyo coseno $C_0(M,N) = \cos\theta_0(M,N)$ esta dado por:*

$$sup\{|\langle x,y\rangle| : x \in M, \| x \| \leq 1,\ y \in N \| y \| \leq 1\}$$

Es importante recalcar, que si $M \cap N = \emptyset$ ambas definiciones coinciden, ambas son generalizaciones de la formula general del coseno de un ángulo dado por 2 puntos sobre un espacio de Hilbert.

Las nociones de ángulos entre subespacios descritas anteriormente nos permiten realizar un estudio de la velocidad de convergencia del método de las proyecciones alternantes, cuya velocidad de convergencia depende del ángulo que formen entre si los subespacios involucrados, en particular si el mismo es muy pequeño la velocidad de convergencia es muy lenta [8]. En el próximo capítulo, se describe de forma un poco más detallada la noción de ángulos entre subespacios.

A continuación mostraremos un lema establecido por Kayalar, Weinert y Deutsch [18], que relaciona el ángulo formado por los subespacios en función de las proyecciones.

Lema 1.20. *(Kayalar, Weinert y Deutsch) Sean M, N subespacios de un espacio de Hilbert H entonces, se puede verificar que:*

$$\begin{aligned} 1.\ C(M,N) &= c_0(M, N \cap (M \cap N)^{\perp}) \\ 2.\ C_0(M,N) &= ||P_M P_N|| \\ 3.\ C(M,N) &= ||P_M P_N - P_{M\cap N}|| = ||P_M P_N P_{(M\cap N)^{\perp}}|| \end{aligned}$$

Demostración. Antes de comenzar, consideraremos

$$B(H) = \{x \in H/\ ||x|| \leq 1\}$$

1.

$$\begin{aligned}
C(M,N) &= C_0(M \cap (M \cap N)^\perp, N \cap (M \cap N)^\perp) \\
&= \sup\{|\langle x, y\rangle| / x \in M \cap (M \cap N)^\perp \cap B(H), y \in N \cap (M \cap N)^\perp \cap B(H)\} \\
&= \sup\{|P_{M\cap(M\cap N)^\perp} x, P_{N\cap(M\cap N)^\perp} y| / x, y \in B(H)\} \\
&\quad \text{por lemas 1.16 y 1.17} \\
&= \sup\{|P^\perp_{(M\cap N)} P_M x, P^\perp_{(M\cap N)} P_N y| / x, y \in B(H)\} \\
&\quad \text{usando que P es idempotente y autoadjunta} \\
&= \sup\{|P_M x, P_{(M\cap N)^\perp} P_N y| / x, y \in B(H)\} \\
&\quad \text{nuevamente por los lemas 1.16 y 1.17} \\
&= \sup\{|P_M x, P_{N\cap(M\cap N)^\perp} y| / x, y \in B(H)\} \\
&= \sup\{|\langle x, y\rangle| \ / x \in M \cap B(H), y \in N \cap (M \cap N)^\perp \cap B(H)\} \\
&= C_0(M, N \cap (M \cap N)^\perp)
\end{aligned}$$

Usando simetría, tenemos que

$$C(M,N) = C_0(M \cap (M \cap N)^\perp, N)$$

2. Por definición tenemos que

$$\begin{aligned}
C_0(M,N) &= \sup\{|\langle x, y\rangle| \ / x \in M \cap B(H), y \in N \cap B(H)\} \\
&= \sup\{|\langle P_M x, P_N y\rangle| \ / x, y \in B(H)\} \\
&\quad \text{como P es idempotente} \\
&= \sup\{|\langle x, P_M P_N y\rangle| \ / x, y \in B(H)\} \\
&= ||P_M P_N||
\end{aligned}$$

3. Por la parte anterior

$$\begin{aligned} C(M,N) &= ||P_{M\cap(M\cap N)^\perp} P_{N\cap(M\cap N)^\perp}|| \\ &= ||P_M P_{(M\cap N)^\perp} P_N P_{(M\cap N)^\perp}|| \\ &= ||P_M P_N P_{(M\cap N)^\perp}|| \\ &= ||P_M P_N (I - P_{M\cap N})|| \\ &= ||P_M P_N - P_M P_N P_{M\cap N}|| \\ &= ||P_M P_N - P_{M\cap N}|| \end{aligned}$$

□

Para más propiedades entre ángulos y proyecciones ver el articulo de F. Deutsch [9].

CAPÍTULO 2

Método de las Proyecciones Alternantes

2.1 Introducción

El método de Proyecciones Alternantes (MAP), fue propuesto originalmente por John Von Neumann, el cual trató el problema de encontrar la proyección de un punto dado sobre la intersección de 2 subespacios de Hilbert cerrados. Tiempo después, Cheney y Goldstein [6] generalizaron el método a casos de 2 conjuntos convexos cerrados.

El estudio de este método servirá como motivación para el estudio de otros métodos como el de Kaczmarz y el de Cimmino utilizando como referencia [7] y [10].

2.2 Proyecciones Alternantes

Geométricamente, el método se encarga de encontrar un punto x_0 en la intersección de dos subespacios $A \cap B$, proyectando de manera alternada sobre cada uno de ellos hasta encontrar la mejor aproximación del punto en la intersección.

Uno de los primeros resultados que dieron origen a este método y que sirvió como

motivación para la implementación del mismo es el siguiente:
Dado H un espacio de Hilbert, M y N dos subespacios de H, entonces $P_M P_N = P_N P_M$, si y sólo si $P_M P_N = P_{M \cap N}$, es decir, P_M y P_N conmutan si y sólo si la composición también es una proyección ortogonal.

Von Neumann a partir de este resultado, se interesó en el caso en que P_M y P_N no conmuten, obteniendo el siguiente resultado:

Teorema 2.1. *(Von Neumman, 1933)[7] Sean M y N subespacios cerrados en un espacio de Hilbert H. Entonces, para cada $x \in H$,*

$$\lim_{n \longrightarrow \infty} (P_M P_N)^n x = P_{M \cap N} x$$

Observación 2.2. *La demostración del teorema anterior puede ser encontrada con detenimiento en [7], [23].*

Este teorema, da lugar al siguiente esquema iterativo:

$$\begin{aligned} x_0 &= x \\ x_n &= P_M P_N x_{n-1} = (P_M P_N)^n x \end{aligned}$$

Este esquema iterativo genera una sucesión $x_n$que converge a $P_{M \cap N} x_0$.

El resultado obtenido en el teorema anterior, puede ser generalizado al caso de más subespacios, tal como se enuncia en el siguiente teorema:

Teorema 2.3. *(Halperin, 1962)[7] Para cada $x \in H$,*

$$\lim_{n \to \infty} (P_{M_k} P_{M_{k-1}} ... P_{M_1})^n x = P_{\cap_1^k M_i} x$$

La demostración de este teorema puede ser encontrada en [7], [10].

En particular, si $M_1, M_2, ..., M_k$ son subespacios cerrados de un espacio de Hilbert H, $M = \cap_{i=1}^k M_i$ y $P_i = P_{M_i}$. Este teorema es útil para encontrar la mejor aproximación desde la intersección de M a cualquier $x \in H$, realizando iteraciones repetitivas sobre

cada subespacio de M_i.

La velocidad de convergencia con la que $(P_k...P_2P_1)^n x$ tiende a $P_{M_k}x$ para cada $x \in H$ depende de la norma del operador $(P_k...P_2P_1)^n - P_M$ que implícitamente depende de los ángulos formados entre los subespacios [7]; por lo que la convergencia puede ser arbitrariamente lenta en algunas aplicaciones. Sin embargo, el MAP ha encontrado una amplia gama de aplicaciones en varias áreas distintas a la matemática, entre ellas:

1. Resolución de Ecuaciones Lineales
2. Teoría lineal de aproximación en el campo de la Probabilidad y Estadística
3. Problema de Dirichlet
4. Cálculo del Núcleo de Bregman
5. Aproximación de Funciones Multivaluadas
6. Restauración de Imágenes
7. Tomografía Computarizada

Observación 2.4. *Muchas de estas aplicaciones se encuentran descritas en el paper de F. Deutsch [7].*

2.2.1 Velocidad de Convergencia

La velocidad de convergencia del método de las proyecciones alternantes (MAP) ha sido estudiado por varios autores, entre ellos F. Deutsch [7], Kayalar y Weinert [18], Smith, Solmon y Wagner [22]. La velocidad del MAP es r-lineal, y se establece en función de las nociones de ángulos definidas en el capítulo anterior.

El siguiente teorema muestra la forma en que influye el ángulo que formen entre si los subespacios involucrados en el MAP.

Teorema 2.5. *(Aronszajn, 1950)[10] Sean M, N subespacios cerrados en un espacio de Hilbert H y $C = C(M, N)$. Entonces, para cada $x \in H$*

$$||(P_N P_M)^n(x) - P_{M\cap N}(x)|| \leq C^{2n-1}||x - P_{M\cap N}(x)|| \leq C^{2n-1}||x||$$

para $n = 1, 2, ...$, la constante C^{2n-1} es lo más pequeño posible.

Las siguientes figuras muestran el comportamiento del MAP en el caso de 2 subespacios

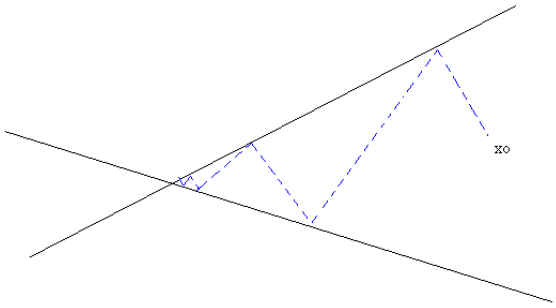

Figura 2.1: MAP: ángulo entre subespacios grande

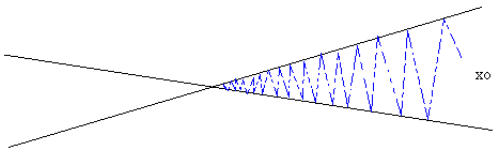

Figura 2.2: MAP: ángulo entre subespacios pequeño

Estas figuras nos permiten visualizar la idea descrita en el capítulo anterior sobre la velocidad de convergencia respecto al ángulo que formen entre si los subespacios involucrados. En la figura (2.1) notamos que el método converge en pocas iteraciones debido a que el ángulo que forman entre si los subespacios es grande, mientras que en la

figura (2.2) notamos lo contrario ya que el ángulo es más pequeño.

Para el caso de más de dos subespacios, existe un resultado análogo al anterior, el cual se enuncia a continuación:

Teorema 2.6. *(Smith, Solmon, Wagner; 1977) [7] Sean $M_1, M_2, \ldots, M_k$, subespacios cerrados de H y sea $M = \cap_{i=1}^{k} M_i$. Entonces, para cada $x \in H$ y un entero $n \geq 1$*

$$||(P_{M_k} P_{M_{k-1}} ... P_{M_1})^n(x) - P_{\cap_{i=1}{}^k}(x)|| \leq C^n ||x||$$

Donde

$$C = [1 - \prod_{i=1}^{k-1} \sin^2 \theta_i]^{1/2}$$

y θ_i es el ángulo entre el subespacio M_i y $\cap_{j=i+1}^{k} M_j$.

La demostración de los teoremas 2.5 y 2.6 puede ser encontrada con más detenimiento en [8] o en [10].

2.3 Método de Kaczmarz

Este método nace como una extensión inmediata del MAP debido a la generalización que realiza el Teorema de Halperin. El método de Kaczmarz para resolver sistemas de ecuaciones se deduce de este teorema justo cuando las M_i son variedades lineales cerradas (subespacios trasladados)con $\cap_{i=1}^{k} M_i \neq \emptyset$.

Consideremos el siguiente sistema de ecuaciones

$$Ax = b$$

donde A es una matriz con coeficientes reales de tamaño $m \times n$, $x \in \mathbb{R}^n$ y $b \in \mathbb{R}^m$, si generalizamos este problema para un espacio de Hilbert H, es equivalente a encontrar un punto (si la solución existe) en la intersección de los k hiperplanos o variedades lineales cerradas, dado por

$$H_i = \{x \in H : \langle a_i, x \rangle = b_i\}$$

donde $a_i = (a_{i1}, a_{i2}, ..., a_{im}) \in \mathbb{R}^m$, $x \in \mathbb{R}^m$ y $b_i \in \mathbb{R}$ con $i = 1, ..., k$.

Así pues, encontrar la solución de $Ax = b$, es equivalente a encontrar un punto en $\cap_{i=1}^{k} M_i$, y para encontrar dicho punto basta tomar un iterado inicial $x_0 \in R^m$, y de manera inductiva definir

$$x_n = (P_{H_k} P_{H_{k-1}} ... P_{H_2} P_{H_1})^n (x) x_{n-1} \; n = 1, 2, ...$$

Pero por el MAP sabemos que

$$x_n \to P_{\cap_{i=1}^{k} H_i}(x_0)$$

y este valor es el que satisface el sistema $Ax = b$.

Siguiendo las ideas del MAP, a partir de un $x_0 \in H$ arbitrario, un paso típico del método de Kaczmarz, puede describirse como

$$x_{k+1} = x_k + w_k(P_{H_{i_k}} - x_k)$$

donde $0 < \epsilon \leq w_k \leq 2 - \epsilon < 2$, para todo k y para un ϵ pequeño dado. [23]

2.4 Método de Cimmino

Este resumen se basa en el survey de Benzi [2] en las notas de Luis M. Hernández [16].

Este método fue propuesto por el matemático Gianfranco Cimmino, es muy parecido al MAP, pues encuentra el punto de intersección de n subespacios proyectando simultáneamente sobre cada uno de ellos.

Cimmino consideró en su método un sistema de ecuaciones lineales de la forma $Ax = b$, donde A es una matriz real de tamaño $n \times n$ inicialmente asumida como no singular, y $b \in \mathbb{R}^n$.

Si $a_i^T = [a_{i1}, a_{i2}, a_{i3}, \ldots, a_{in}]$ denota la *i-ésima* fila de A, la solución $X_* = A^{-1}b$, es la única intersección de los hiperplanos descritos por

$$\langle a_i, x\rangle = b_i, \text{ con } i = 1, 2, \ldots, n \tag{2.1}$$

De este modo, dada una aproximación inicial $x^{(0)} \in \mathbb{R}^n$, Cimmino, toma para cada $i = 1, 2, \ldots, n$, la proyección $x_i^{(0)}$ de $x^{(0)}$, con respecto a cada hiperplano $< a_i, x >= b_i$ para $i = 1, \ldots, n$ (2.1), así

$$x_i^{(0)} = x^{(0)} + 2\frac{b_i - \langle a_i, x^{(0)}\rangle}{\|a_i\|^2}a_i \tag{2.2}$$

Cimmino observó que el punto inicial $x_i^{(0)}$ y su proyección sobre los n hiperplanos están en un entorno cuyo punto central era precisamente común a todos los hiperplanos, dicho punto era la solución al sistema de ecuaciones descrito inicialmente ($Ax = b$). Para esto, en el vector de posición de cada proyección de $x_{(0)}$ denotados por x_i^0 para $i = 1, \ldots, n$, se sustituyen n cantidades arbitrarias $m_1, m_2, \ldots, m_n$, posteriormente se calcula el centro de gravedad del sistema de masas, el cual se encuentra dentro del entorno común a todos los hiperplanos, generando el proximo iterado de la secuencia.[2]

Así, el nuevo punto que se genera en la iteración anterior $x^{(1)}$ es mejor aproximación a la solución del sistema que $x^{(0)}$, es decir

$$\|x^{(1)} - x_*\| \leq \|x^{(0)} - x_*\|.$$

Este procedimiento se continua realizando, hasta obtener una mejor aproximación de $x^{(1)}$, la sucesión $\{x^{(k)}\}$, converge a $x_* = A^{-1}b$ cuando $k \to \infty$.

En forma matricial, el método de Cimmino puede ser escrito como

$$x^{(x+1)} = x^{(k)} + \frac{2}{\mu}A^T D^T D(b - Ax^{(k)}) \tag{2.3}$$

con $k = 0, 1, \ldots$, donde

$$D = \begin{pmatrix} \frac{\sqrt{m_1}}{\|a_1\|} & & & \\ & \frac{\sqrt{m_2}}{\|a_2\|} & & \\ & & \ddots & \\ & & & \frac{\sqrt{m_n}}{\|a_n\|} \end{pmatrix}$$

y $\mu = \sum_{i=1}^{n} m_i$. En particular si tomamos, $m_i = \|a_i\|^2$. obtenemos

$$x^{(k+1)} = x^{(k)} + \frac{2}{\mu} A^t (b - Ax^{(k)})$$

El cual no es más que un caso particular del método iterativo de Richardson aplicado al sistema de ecuaciones $A^t Ax = A^t b$.

Cuando el sistema anterior es inconsistente, los iterados generados por el método de Cimmino converge a la solución de mínimos cuadrados, es decir, $||b - Ax||_2 = min$.

La siguiente figura nos permite tener una idea del funcionamiento del método en el caso de dos subespacios

Acá se puede observar la manera como el método encuentra cada uno de sus iterados. Es decir, dado un punto inicial, se calcula la distancia del segmento que une la proyección de dicho punto a cada subespacio, convirtiendo el promedio de estos en el proximo iterado. Este proceso se suele realizar hasta obtener la convergencia. Aunque en este método la velocidad de convergencia puede ser muy lenta, existe una gran ventaja que favorece el uso del mismo, este método es muy paralelizable, lo que permite de alguna manera u otra garantizar la eficacia al resolver alguna aplicación desde diferentes unidades de procesamiento en arquitecturas computacionales paralelas.

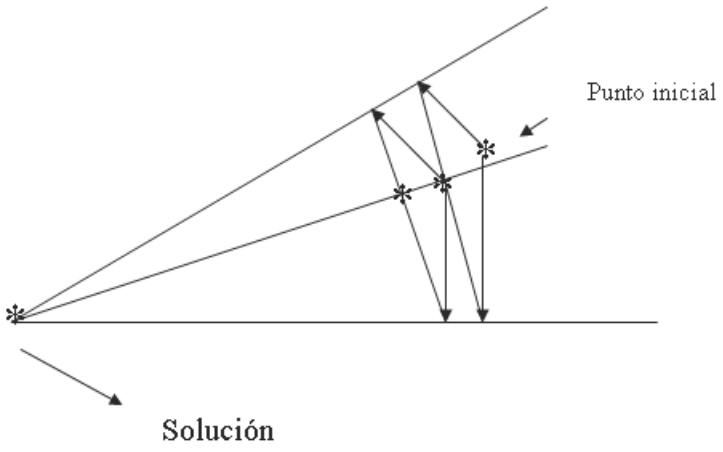

Figura 2.3: Método de Cimmino

El método de Cimmino y el de Kaczmarz [11], se encuentran relacionados entre si, el de Cimmino resulta más eficaz si se adapta a computadoras paralelas, mientras que el método de Kaczmarz tiende a converger de forma más rápida.

El método de Cimmino puede extenderse al caso de encontrar un punto en la intersección de subespacios. En este caso, toma la siguiente forma :

Sean M_i, con $i = 1, ..., n$ subespacios de un espacio de Hilbert H. Sea P_{M_i} el operador proyección ortogonal sobre M_i, luego para un punto dado, $x = x_0 \in H$

$$x_{k+1} = \frac{1}{n} \sum_{i=1}^{n} P_{M_i} x_k$$

Es decir, el iterado x_{k+1} es un promedio de los subespacios M_i del iterado anterior.
El método de Cimmino tiene muchas aplicaciones en varios campos de la ciencia, entre ellos:

- Programación en matemática convexa
- Adaptación a la terapia de radiación

- Solución de problemas con inversas en el area de Astronomía, Física médica y Geofísica
- Reconstrucción de imágenes mediante proyecciones
- Formación de redes neurales
- Solución de grandes sistemas lineales, mediante la discretización de ecuaciones.

Para más detalles de estas y otras aplicaciones, ver [2].

CAPÍTULO 3

Método de los Gradientes Conjugados

3.1 Introducción

El método de los gradientes conjugados (GC), es un procedimiento iterativo originalmente propuesto por (Hestenes y Stiefel, 1950)[20] como un método directo para resolver sistemas lineales de la forma

$$Ax = b \tag{3.1}$$

donde $A \in \mathbb{R}^{n \times n}$ es una matriz simétrica y positivo definida y b un vector en $\mathbb{R}^{n \times 1}$.
Este resumen se basa en lo descrito en [19], [20], [21].

3.2 Descripción del método

En teoría, es un método directo que produce una secuencia de iterados x^i, $i = 1, ..., n$ tal que $Ax^i = b$, para cualquier vector inicial x^0. Es decir, teóricamente el método GC tiene terminación finita. Sin embargo, en la aritmética computacional esto no es del todo cierto, en principio, en la década de los 50 se abandonó como método directo. Tiempo después, el método GC fue recuperado, esta vez como método iterativo, ya que produce

una secuencia que numéricamente converge a la solución del sistema $Ax = b$. El método de los gradientes conjugados es famoso por su rápida velocidad de convergencia y el uso de poca memoria en el computador a la hora de ser ejecutado, tiene extensiones al caso indefinido y más aun para el caso no simétrico, en este capítulo, solo nos interesaremos por la extension al caso simétrico.

Antes de comenzar el estudio del método de los gradientes conjugados, se describe brevemente en que consisten los métodos tipo gradientes, los cuales son aquellos que toman el gradiente negativo como dirección de búsqueda, en particular, nos interesaremos en la forma de estos métodos de minimización, en el caso donde f sea una función cuadrática.

Figura 3.1: Método de mínimo descenso

3.3 Caso Cuadrático

La idea fundamental del método de los gradientes conjugados en el caso cuadrático, se basa en minimizar una función $q(x)$, definida por

$$q(x) = \frac{1}{2}x^T Ax - x^T b + c$$

donde x y b son vectores en $\mathbb{R}^n$ y $c \in \mathbb{R}$.
De este modo, si la matriz A es simétrica positivo definida, minimizar la función $q(x)$ es equivalente a resolver el sistema planteado en (3.1).

Consideremos el siguiente conjunto de direcciones $\{s_1, s_2, \ldots, s_n\}$, que satisfacen la siguiente propiedad:

$$s_i^T A s_j = 0 \text{ si } i \neq j \text{ y } 1 \leq i, j \leq n$$

La propiedad anterior se conoce como **A - conjugancia** y garantiza que los vectores $\{s_1, s_2, \ldots, s_n\}$, formen un conjunto linealmente independiente en $\mathbb{R}^n$.

Si se usan estos vectores, como direcciones de búsqueda, se obtiene el siguiente método iterativo, el cual convergerá a lo sumo en n pasos independientemente del iterado inicial x_0 escogido

$$x_{k+1} = x_k + \alpha s_k \tag{3.2}$$

donde α es un parámetro que cambia en cada iteración, con el objetivo de minimizar la función $q(x_k + \alpha s_k)$ y s_k la dirección de mínimo descenso.

Lema 3.1. *Si x y u son vectores cualesquiera en $\mathbb{R}^n$ con $u \neq 0$, entonces*

$$argmin_{\alpha>0}\ q(x + \alpha u) = \frac{r(x)^t u}{u^t A u}$$

donde $r(x) = -\nabla q(x) = b - Ax$.

Demostración. Este resultado, se obtiene de aplicar condiciones necesarias y suficientes de optimalidad en una variable a la función

$$\phi(\alpha) = q(x + \alpha u) = q(x) + \frac{1}{2}\alpha^2 u^t A u - \alpha u^t r(x)$$

. □

Teorema 3.2. *Sea $u_1, u_2, \ldots u_n$ un conjunto A - conjugado de vectores no ceros. Si x_0 (arbitrario) es dado y*

$$x_k = x_{k-1} + \alpha u_k \text{ para } 1 \leq k \leq n \tag{3.3}$$

donde α_k se escoge como el $argmin_{\alpha>0}\ q(x + \alpha u)$, entonces $Ax_n = b$.

Demostración. Multiplicando por A la ecuación descrita en (3.3) y aplicando de manera recursiva la misma ecuación obtenemos

$$Ax_n = Ax_0 + \alpha_1 Au_1 + \ldots + \alpha Au_n$$

Restando el vector b en ambos lados, haciendo el producto interno con u_k y usando la A-conjugancia, se tiene que

$$(Ax_n - b)^t u_k = (Ax_0 - b)^t u_k + \alpha_k u_k^t A u_k \tag{3.4}$$

Ahora, queremos probar que el lado derecho de (3.4) es cero para todo k, por el Lema (3.3), $\alpha_k = \frac{(b - Ax_{k-1}^t u_k)}{u_k^t A u_k}$ y aplicando lo anterior nuevamente en forma recursiva, tenemos

$$Ax_{k-1} = Ax_0 + \alpha_1 Au_1 + \ldots + \alpha_{k-1} Au_{k-1}$$

de donde,

$$\alpha_k = \frac{(b - Ax_0)^t u_k}{u_k^t A u_k},$$

y por lo tanto el lado derecho de la ecuación (3.4) es cero.
Como consecuencia,$(Ax_n - b)^t u_k = 0$, para todo k, y como cada u_k es linealmente independiente, tenemos que $Ax_n = b$ □

Es importante destacar de la demostración del Teorema (3.2) que en cada iteración se cumple que $r_k^t u_j = 0$ para todo $1 \leq j \leq k$, donde r_k denota r_k denota $r(x_k)$. Además, en cada iteración

$$q(x) = q(x_{k-1}) - \frac{(u_k^t r_{k-1})^2}{2u_k^t A u_k},$$

y como A es positiva definida (PD), las direcciones u_k son de descenso siempre y cuando $u_k^t r_{k-1} \neq 0$. Más aún, la propiedad optimal de α_k y la A-conjugancia garantizan que cada x_k minimiza a $q(x)$ sobre la variedad lineal (subespacio trasladado) $x_0 + exp\{u_1, \ldots, u_k\}$, donde $exp\{u_1, \ldots, u_k\}$ es el subespacio generado por los vectores $u_1, \ldots, u_k$.

Teorema 3.3. *Bajo las hipótesis del Teorema (3.2), Para cada k se cumple que*

$$q(x_k) = \min_{x \in x_0 + exp\{u_1,...,u_k\}} q(x) \tag{3.5}$$

Demostración. Como q es una función estrictamente convexa, basta demostrar que el residual r_k es ortogonal al subespacio $exp\{u_1, ..., u_k\}$, pero esto es cierto, ya que $r_k^t u_j = 0$ para todo $1 \leq j \leq k$. □

La propiedad (3.5), garantiza la terminación finita de todo método que use direcciones A-conjugadas. En Particular en el método de Gradientes Conjugados comenzamos con un iterado inicial x_0 y construimos secuencialmente un conjunto de direcciones A-conjugadas $\{u_1, u_2, \ldots\}$ tal que los iterados $x_k = x_{k-1} + \alpha_k u_k$ satisfacen (3.5). Además, como deseamos obtener una buena reducción del valor de $q(x_k)$ en cada k, escogemos u_k como el vector más proximo a r_{k-1} (aquel que minimiza a $||u - r_{k-1}||_2$) que preserva la A-conjugancia de los vectores $\{u_1, \ldots, u_{k-1}\}$. En su versión mas cruda, el algoritmo iterativo tiene la siguiente estructura:

Algoritmo 1 Gradientes Conjugados (1º Versión)

- **Entrada**: x^0, $g_0 = \nabla f(x_0)$, $s_0 = -g_0$,
- **Salida** : x_{k+1} (Solución Numérica)
- Para $k = 1, 2, ...$ hacer:
- $x_{k+1} = x_k + \alpha s_k$
- $g_{k+1} = \nabla f(x_{k+1})$
- $\beta_{k+1} = \frac{(g_{k+1}^T g_{k+1})}{g_k^T g_k}$
- $s_{k+1} = -g_{k+1} + \beta_{k+1} s_k$
- end, para

Observación 3.4. *En el algoritmo anterior, el valor de α se obtiene mediante la dirección de búsqueda del método, es decir, dada una dirección de descenso, tal como la del gradiente negativo, la determinación de un valor apropiado para α en cada iteración es equivalente a resolver un problema de minimización en una dimensión [14]:*

$$\min_{\alpha} f(x_k - \alpha \nabla f(x_k)).$$

Notese las siguientes características derivadas del algoritmo anterior:

- El Gradiente negativo ó el menos gradiente es simplemente el vector residual:

$$-\nabla q(x) = b - Ax = r$$

- Para buscar la dirección s_k, no hace falta una búsqueda lineal porque la elección más optima para la elección de α es determinada analíticamente, específicamente el mínimo sobre α ocurre cuando el nuevo residual es ortogonal a la dirección encontrada, es decir:

$$0 = \frac{d}{d\alpha} q(x_{k+1}) = \nabla(x_{k+1})^T \frac{d}{d\alpha} x_{k+1} = (Ax_{k+1} - b)^T (\frac{d}{d\alpha}(x_k + \alpha s_k)) = -r_{k+1}^T s_k \tag{3.6}$$

Donde el nuevo residual puede ser expresado en términos del residual anterior y la dirección de búsqueda,

$$r_{k+1} = b - Ax_{k+1} = b - A(x_k + \alpha s_k) = (b - Ax_k) - \alpha A s_k = r_k - \alpha A s_k \tag{3.7}$$

de donde obtenemos:

$$\alpha = \frac{r_k^T s_k}{s_k^T A s_k} \tag{3.8}$$

Ahora, si sustituimos estos cambios en (3.3), obtenemos el siguiente esquema iterativo:

Algoritmo 2 Gradientes Conjugados (2º Versión)

- **Entrada**: x^0, $g_0 = \nabla f(x_0)$, $s_0 = r_0 = -g_0$
- **Salida** : x_{k+1} (Solución Numérica)
- $\alpha_k = \frac{r_k^T s_k}{s_k^T A s_k}$
- $x_{k+1} = x_k + \alpha_k s_k$
- $r_{k+1} = r_k - \alpha_k A s_k$
- $\beta_{k+1} = \frac{(r_{k+1}^T r_{k+1})}{r_k^T r_k}$
- $s_{k+1} = r_{k+1} + \beta_{k+1} s_k$

3.3.1 Gradientes Conjugados como Método Iterativo

La terminación finita del método GC, es un resultado de mucha elegancia matemática que depende fuertemente de la A-conjugancia. Lamentablemente, los errores de redondeo, con los que debemos aprender a vivir en la aritmética computacional, destruyen la ortogonalidad y por ende la A-conjugancia del método. Por lo tanto, la terminación finita de GC no es un resultado realista. Mas aún, en los casos en que fuese posible observar la terminación del método, realizar n iteraciones cuando los problemas son grandes es un resultado indeseable. Sin embargo, el método GC puede ser visto como un esquema iterativo en el sentido clásico, y afortunadamente en ese caso posee propiedades que permiten proponer variantes que lo transforman en un método rápido y eficiente para problemas grandes y sparse. [20]

A tal sentido se estudiarán algunas propiedades de convergencia del método, en relación entre el número de iteraciones requeridas y la ubicación de los espectros de A. Estos teoremas y sus demostraciones se encuentran con mayores detalles en notas del prof. Marcos Raydan [20].

Teorema 3.5. *Si $A = I + B$ es Positivo definida y el rango de B es $p < n$, entonces el Gradiente Conjugado converge a lo sumo en p, iteraciones.*

Demostración. Ver [20] □

Este teorema nos indica que que si A es una perturbación del rango bajo la matriz identidad, o si A posee pocos autovalores distintos, entonces el Gradiente Conjugado convergerá en pocas iteraciones. Otros resultados importantes sobre la velocidad de la convergencia del método se pueden obtener usando la norma A definida de la siguiente manera

$$||z||_A = \sqrt{z^t A z}$$

En efecto para todo $x \in \mathbb{R}^n$, y denotando $x_* = A^{-1}b$, tenemos

$$||x - x_*||_A^2 = 2q(x) + x_*^t A x_*,$$

y por lo tanto minimizar q o minimizar el error en la norma A son problemas equivalentes.

Teorema 3.6. *El iterado x_k generado por el GC cumple lo siguiente*

$$||x_k - x_*||_A^2 = \min_{P_{k-1}} (x_0 - x_*)^t A (I - A P_{k-1}(A))^2 (x_0 - x_*), \tag{3.9}$$

sobre todos los posibles polinomios de grado $k-1$.

La demostración de este resultado, puede ser encontrada en [20], para nuestros efectos, solo nos interesa la igualdad del mismo.

Teorema 3.7. *En el método de GC se cumple que*

$$||x_k - x_*||_A^2 \leq \max_{\lambda_i} (I - \lambda_i P_{k-1}(\lambda_i))^2 ||x_0 - x_*||_A^2,$$

para todo polinomio P_{k-1} de grado $k-1$, y donde el máximo se toma en los autovalores λ_i de A.

Demostración. Por el teorema (3.2), podemos escribir

$$x_0 - x_* = \lambda_1 v_1 + ... + \lambda_n v_n \tag{3.10}$$

aplicando la norma a ambos lados, tenemos

$$||x_0 - x_*|| = ||\lambda_1 v_1 + ... + \lambda_n v_n||$$

pero como cada v_i es ortonormal, tenemos que

$$||x_0 - x_*||_A^2 = \sum_{i=1}^{n} \lambda_i \delta_i^2 \tag{3.11}$$

ahora si combinamos (3.10) y (3.11) obtenemos, para cualquier polinomio de grado $k-1$, lo siguiente

$$\begin{aligned} ||x_k - x_*||_A^2 &\leq \sum_{i=1}^{n} (I - \lambda_i P_{k-1})^2 \delta_i^2 \\ &\leq \max_{\lambda_i} (I - \lambda_i P_{k-1}(\lambda_i))^2 \sum_{i}^{n} \lambda_i \delta_i^2 \end{aligned}$$

que combinando con (3.11) nos permite concluir (3.10). □

Este teorema permite obtener una cota de la velocidad de convergencia del método GC que solo depende de la condición espectral $\kappa_2(A) = \lambda_{max}/\lambda_{min}$ de la matriz A, donde λ_{max} *y* λ_{min} son respectivamente el máximo y el mínimo autovalor de A.

CAPÍTULO 4

Aceleración del método de Cimmino

En este capítulo se describen algunos aspectos fundamentales del método propuesto, entre ellos su estructura, velocidad de convergencia, condiciones necesarias y/o suficientes para garantizar convergencia así como una descripción breve del tipo de problemas que estudiaremos después de la implementación del esquema iterativo.

4.1 El Método de Cimmino como método de Optimización

En esta sección hablaremos del método de Cimmino visto desde otra perspectiva, dichas acotaciones permitirán el desarrollo de nuestros objetivos posteriormente, estas y otras notas al respecto se encuentran en notas de un articulo de Luis M. Hernández [15].

Sea H un espacio de Hilbert con dimensión finita, y sea P un operador lineal en H. Definimos la función

$$q(x) = \frac{1}{2}\|Px\|^2.$$

Calculamos el gradiente $\nabla q(x)$, definiendo para $d \in H$ dada la función auxiliar,

$$g(t) = q(x+td) = \frac{1}{2}\langle P(x+td)P(x+td)\rangle \tag{4.1}$$

$$= \frac{1}{2}\left(\|Px\|^2 + 2t\langle Px, Pd\rangle + t^2\|Pd\|^2\right) \tag{4.2}$$

Derivamos la función auxiliar,

$$g'(t) = \langle Px, Pd\rangle + t\|Pd\|^2$$

$$g'(0) = \langle Px, Pd\rangle$$

Luego $< \nabla q(x), d >= g'(0)$, por lo tanto,

$$\nabla q(x) = P^*P(x), \tag{4.3}$$

donde P^* es el operador adjunto de P.

Para calcular la Hessiana, derivamos de nuevo,

$$g''(t) = \langle Pd\rangle^2 = \langle Pd, Pd\rangle. \tag{4.4}$$

Como $g''(0) = \langle d, \nabla^2 q(x)d\rangle$, en consecuencia tenemos que la Hessiana en este caso es una constante dada por el operador

$$\nabla^2 q(x) = P^*P \tag{4.5}$$

Si el operador lineal P es idempotente ($P^2 = P$)y autoadjunto ($P^* = P$), entonces el gradiente y la Hessiana de $q(x)$ son respectivamente:

$$\nabla q(x) = Px, \tag{4.6}$$

y

$$\nabla^2 q(x) = P. \tag{4.7}$$

Este es el caso cuando P es un operador de proyección ortogonal en un subespacio.

Sean $M_i,\ i = 1, \ldots, m$, subespacios de H. Sea P_{M_i} el operador de proyección ortogonal sobre M_i. Definamos la función

$$f(x) = \frac{1}{2}\sum_{i=1}^{m} \|x - P_{M_i}x\|^2 = \frac{1}{2}\sum_{i=1}^{m} \|(I - P_{M_i})x\|^2$$

El operador $I - P_{M_i}$ es también un operador de proyección para cada i. En efecto, es fácil verificar que $I - P_{M_i}$ es el operador de proyección ortogonal sobre $M_i^{\perp}$, complemento ortogonal de M_i.

Por lo tanto, aplicando (4.6) y (4.7) tenemos,

$$\nabla f(x) = \sum_{i=1}^{m} x - P_{M_i}x.$$

y además,

$$\nabla^2 f(x) = \sum_{i=1}^{m} I - P_{M_i}.$$

El operador $\nabla^2 f(x)$ es semi-positivo definido. En efecto:

Sea $x \neq 0$ cualquiera,

$$\langle x, \nabla^2 f(x)x\rangle = \sum_{i=1}^{m} \|x - P_{M_i}x\|^2 \geq 0.$$

Además, $\langle x, \nabla^2 f(x)x\rangle = 0$ si y solo si $x \in \cap_{i=1}^{m} M_i$.

Tomando esto en consideración, se plantearan métodos iterativos para resolver el problema de minimización

$$\text{mín}\, f(x) = \frac{1}{2}\sum_{i=1}^{m} \|x - P_{M_i}x\|^2. \tag{4.8}$$

Las soluciones de este problema de minimización, son los $x \in \cap_{i=1}^{m} M_i$.

Así para $k = 1, \ldots,$ y para $x_0 \in H$ dado, definimos la iteración,

$$x_{k+1} = x_k - \alpha_k \nabla f(x_k), \tag{4.9}$$

que para la función dada corresponde con:

$$x_{k+1} = x_k - \alpha_k \sum_{i=1}^{m} x - P_{M_i} x_k. \tag{4.10}$$

En el caso donde $\alpha_k \equiv 1/m$, esta iteración es equivalente a,

$$x_{k+1} = \frac{1}{m} \sum_{i=1}^{m} P_{M_i} x_k, \tag{4.11}$$

que no es más que el método de Cimmino en su versión clásica, y este a su vez un método de Cauchy o de descenso con la dirección del gradiente a paso fijo, lo cual explicaría la lenta convergencia del método. A partir de acá se propone el uso de técnicas de optimización más eficientes, como por ejemplo la aceleración del método de Cimmino con el método de los Gradientes Conjugados que es un método bastante eficiente.

4.2 Aceleración del Método de Cimmino

El método de Cimmino, es un método ampliamente conocido por su paralelismo, hecho que lo hace eficaz a la hora de resolver problemas en arquitecturas computacionales paralelas.

En la sección anterior, nos dimos cuenta que el método de Cimmino es un método tipo gradiente a paso fijo, pues usa una sola dirección de búsqueda para sus iterados (gradiente negativo), lo que explicaría la lentitud del método.

Si se realizan otras elecciones de α_k , se pueden encontrar opciones más rápidas que el método clásico de Cimmino. Por ejemplo, podemos hacer una versión Cimmino del

método de Cauchy tomando

$$\alpha_k = \frac{\langle g_k, g_k \rangle}{\langle g_k, H g_k \rangle}$$

donde

$$g_k = \nabla f(x) = \sum_{i=1}^{m} x - P_{M_i} x,$$

y

$$H = \nabla^2 f(x) = \sum_{i=1}^{m} I - P_{M_i}$$

Sin embargo, el método de Cauchy es conocido también por presentar convergencia lenta, por lo tanto al utilizar la dirección del gradiente podemos crear una versión diferente del método de Cimmino tomando como tamaño de paso α_k,

$$\alpha_k = \frac{\langle s_{k-1}, s_{k-1} \rangle}{\langle s_{k-1}, y_{k-1} \rangle},$$

donde

$s_{k-1} = x_k - x_{k-1}$ y $y_{k-1} = g_k - g_{k-1}$, esta versión es una versión Cimmino del método de Barzilai Borwein [1] .

En el trabajo realizado por A. Fariñas y L.M. Hernández [11], se propone realizar una versión basada en el método de Barzilai-Borwein del método de Cimmino, con muy buenos resultados. Esta versión, utiliza igualmente la dirección del gradiente negativo cambiando la escogencia de la longitud del paso.

En este trabajo se propone una versión del método de Cimmino basada en el método de los gradientes conjugados para minimizar la funcional descrita en (4.8).

El esquema iterativo de Cimmino Gradientes Conjugados, tiene la siguiente estructura:

Algoritmo 3 Método de Cimmino Gradientes Conjugados

- **Entrada**: x_0, $g_0 = \nabla f(x_0)$, $s_0 = r_0 = -g_0$
- **Salida** : x_{k+1} (Solución Numérica)
- $\alpha_k = \frac{r_k^T s_k}{s_k^T H s_k}$
- $x_{k+1} = x_k + \alpha_k s_k$
- $r_{k+1} = r_k - \alpha_k H s_k$
- $\beta_{k+1} = \frac{(r_{k+1}^T r_{k+1})}{r_k^T r_k}$
- $s_{k+1} = r_{k+1} + \beta_{k+1} s_k$

Este algoritmo aplica el método de los gradientes conjugados a la funcional descrita anteriormente y se considera $H = \sum_{i=1}^{m} I - P_{M_i}$. Para calcular el gradiente, se realizan las proyecciones sobre cada uno de los subespacios (puede hacerse en paralelo).

4.3 Método de Cimmino Acelerado para problemas de Saddle Point (Punto de Ensilladura)

En la parte anterior, se describió un poco la forma en que esta estructurado el nuevo esquema propuesto (acelerado), en esta sección, se describirá como se puede utilizar el método de Cimmino para resolver problemas de punto de ensilladura, este resumen se basa en lo descrito por Luis M. Hernández en su articulo [16].

Un problema de punto de ensilladura es un sistema lineal que presenta la forma:

$$\begin{pmatrix} A_{m\times m} & B^T \\ B_{n\times m} & 0 \end{pmatrix} \begin{pmatrix} x \\ \lambda \end{pmatrix} = \begin{pmatrix} f \\ 0 \end{pmatrix} \tag{4.12}$$

donde los elementos que conforman (4.12) cumplen las siguientes condiciones:

- A es una matriz simétrica, $(A = A^T)$
- A debe ser Positiva definida
- $n \geq m$

El sistema anterior (4.12) proviene de condiciones de optimización de 1º orden y de problemas de programación cuadrática:

$$\text{mín}\, M(x) = \frac{1}{2}x^T Ax - f^T x + (Bx)^T \lambda$$

sujeto a

$$Bx = 0$$

En este caso, la variable λ de (4.12) representa el vector de multiplicadores de Lagrange y toda la solución (x_*, λ_*) de este problema, es un punto de ensilladura para el lagrangeano

$$L(x, y) = \frac{1}{2}x^T Ax - f^T x + (Bx)^T \lambda$$

de donde proviene el nombre de punto de ensilladura para los sistemas modelados de esta forma.

Si $(x_*, \lambda_*) \in \mathbb{R}^{n+m}$ es un punto de ensilladura, debe satisfacer

$$L(x_*, \lambda) \leq L(x_*, \lambda_*) \leq L(x, \lambda_*)\ \forall x \in \mathbb{R}^n, \forall \lambda \in \mathbb{R}^m$$

o de una manera equivalente

$$\min_x \max_\lambda L(x, \lambda) = L(x_*, \lambda_*) = \max_\lambda \min_x L(x, \lambda)$$

Este tipo de problemas a gran escala, han venido abarcando todo el área de la Ingeniería, entre los campos donde se encuentra el mayor número de aplicaciones podemos destacar:

- Dinámica Computacional.
- Estimación de mínimos cuadrados.
- Economía
- Circuitos y redes Eléctricos
- Reconstrucción de Imágenes

Referencias para estas y otras aplicaciones pueden ser encontradas con mayores detalles en [3].

Ahora estableceremos condiciones de convergencia para el sistema definido en (4.12).

Teorema 4.1. *Dadas $A \in \mathbb{R}^{n\times n}$ una matriz simétrica positivo definida y $B \in \mathbb{R}^{m\times n}$, con rango m, $m \geq n$, entonces el sistema (4.12) tiene solución única $(x, \lambda)^T$ que satisface que para todo $x \in \ker B$, $f - Ax \perp \ker B$.*

Teorema 4.2. *Bajo las mismas hipótesis anteriores, si en la solución $(x, \lambda)^T$ de (4.12), el vector x corresponde a la proyección ortogonal $x_u = A^{-1}f$ sobre el Kernel de B, en el sentido del producto escalar $< ., . >_A$.*

Observación 4.3. *Las demostraciones de estos teoremas pueden ser encontradas con más detenimiento en [16], para nuestro efecto, solo nos interesaremos en lo que exponen dichos resultados.*

Ahora, si particionamos $B \in \mathbb{R}^{m\times n}$ en r bloques de filas:

$$B^T = [B_1^T, B_2^T, ..., B_r^T]$$

obtenemos que,

$$\ker B = \bigcap_{i=1}^{r} \ker B_i$$

Por el teorema (4.2), se tiene que la solución x para el problema de punto de ensilladura descrito en (4.12) es la proyección A-ortogonal de $x_u = A^{-1}f$, sobre el subespacio $\ker B = \cap_{i=1}^{r} \ker B_i$, así este problema se transforma en otro de mejor aproximación con el producto interno definido por $< x, x >_A = < x, Ax >$ cuando A es simétrica positivo definida, el cual puede ser resuelto por el método de Cimmino [11]:

$$x_{k+1} = \frac{1}{r} \sum_{i=1}^{m} P_{M_i} x_k$$

Pero ahora, en esta elección, el operador P_{M_i}, representa la proyección A-ortogonal sobre $M_i = \ker B_i$. Además como el $\ker B_i$ es un subespacio, podemos calcular la proyección A-ortogonal de un vector $y \in \mathbb{R}^n$ sobre el $\ker B_i$ mediante $P_{M_i} y$, obteniendo x de un problema de punto de ensilladura más pequeño:

$$\begin{pmatrix} A & B_i^T \\ B_i & 0 \end{pmatrix} \begin{pmatrix} x \\ \lambda \end{pmatrix} = \begin{pmatrix} Ay \\ 0 \end{pmatrix} \tag{4.13}$$

lo que hace mucho más sencilla la resolución de este tipo de problemas ya que hay que resolver un sistema de ecuaciones lineales de menor dimensión, donde cada proyección A-ortogonal sobre $\ker B_i$ puede ser representada independientemente, para $i = 1, \ldots, m$, sin embargo, es sabido que el método de Cimmino puede presentar lenta velocidad de convergencia [5], razon por la cual se propone la aceleración del esquema.

CAPÍTULO 5

Experimentación Numérica

En este capitulo, se realizarán algunos experimentos que nos permitirán observar el comportamiento de las iteraciones en el esquema de aceleración propuesto, además se realizaran comparaciones con otros métodos iterativos.

Los experimentos a ejecutar son los siguientes:

1. Probar que el GC Cimmino es la mejor aceleración comparando con otros métodos.
   Acá se realizará dicha prueba a través de la comparación del esquema iterativo propuesto con otros en la resolución de aplicaciones de Saddle Point tomadas de la colección del CUTEr [12] y otros de la discretización realizada por la Descomposición de Dominios.

2. Verificación del mejor tamaño de bloque.
   Acá solo usaremos el esquema propuesto (GC- Cimmino) bajo distintas hipótesis, se medirá el tiempo de ejecución del algoritmo para dar condiciones relacionadas con su convergencia.

Los problemas de Saddle Point con los que trabajaremos son los mismos expuestos en el

capitulo anterior:

$$\begin{pmatrix} A & B^T \\ B & 0 \end{pmatrix} \begin{pmatrix} x \\ \lambda \end{pmatrix} = \begin{pmatrix} Ay \\ 0 \end{pmatrix} \tag{5.1}$$

con las condiciones;

- B es una matriz $m \times n$, cuyo rango es m y $m \leq n$.
- A es una matriz $n \times n$, simétrica semi positivo definida.

Los problemas de descomposición de dominio, son extraídos del articulo del Luis M. Hernández [17], dichos problemas son referidos a problemas de punto de ensilladura después de discretizar mediante la descomposición de dominios la ecuación de Poisson de la forma:

$$\begin{cases} -\Delta u = f & \text{en } \Omega \\ u = 0 & \text{en } \partial\Omega \end{cases}$$

Donde Ω es un dominio rectangular acotado en $\mathbb{R}^2$, $\partial\Omega$ la frontera de Ω y f una función conocida.

El dominio Ω a su vez, esta subdividido en cuatro subdominios Ω_i, $i = 1, \ldots, 4$, separados por una interfaz, además de esto se definen matrices A_i, $i = 1, \ldots, 4$, correspondientes a la discretización de la ecuación de Poisson en cada subdominio Ω_i. De este modo la matriz A correspondiente al problema de punto de ensilladura (Saddle Point), tendría la siguiente forma:

$$A = \begin{pmatrix} A_1 & 0 & 0 & 0 \\ 0 & A_2 & 0 & 0 \\ 0 & 0 & A_3 & 0 \\ 0 & 0 & 0 & A_4 \end{pmatrix}$$

A es una matriz diagonal por bloques simétrica y positivo definida. La matriz B necesaria para la generación del problema de Saddle Point, se obtiene por las condiciones de acoplamiento sobre la interfaz, denotaremos a Γ como la interfaz que separa a cada subdominio, esta a su vez se subdivide en subinterfaces Γ_{ij}, con $i, j = 1, ..., 4$, separando así a los subdominios Ω_i y Ω_j, por lo que la matriz B se forma por la matriz B_{ij} que son las condiciones de la subinterfaz Γ_{ij} y cada B_{ij} proviene de la discretización de la condición de acoplamiento:

$$\forall v_i \in V_i, v_j \in V_j, w_{ij} \in W_{ij} : \int_{\Gamma_{ij}} (v_i - v_j) w_{ij} dx = 0$$

donde,

- v_i es la solución local del subdominio Ω_i sobre Γ_{ij}.
- $v_i = v_i \in H'(\Omega_i); v_i|_{\partial\omega} = 0$es el espacio donde la solución local esta definida.
- w_{ij} es una función de prueba de un subespacio propio $W_{ij} = H^{-1/2}(\Gamma_{ij})$

A continuación mostraremos la descomposición de dominio de Ω:

Con dicha descomposición, la discretización de la matriz B_{ij}, expresa una condición del tipo:

$$\forall v_i \in V_i^h, v_j \in V_j^h, w_{ij} \in W_{ij}^h : \int_{\Gamma_{ij}} (v_i - v_j) w_{ij} dx = 0$$

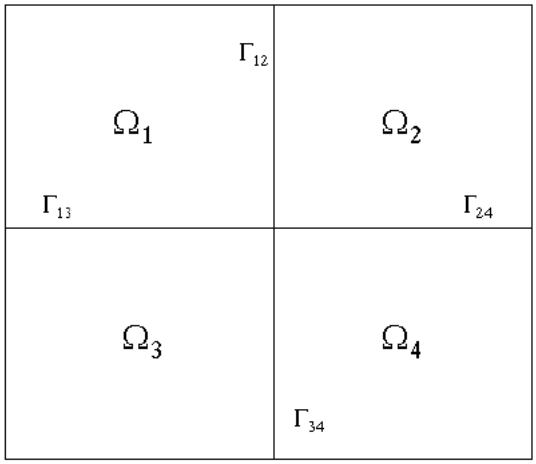

Figura 5.1: Descomposición de Dominio de Ω

Los experimentos numéricos se tomaron del siguiente subespacio:

$$V_i^h = [\{\Phi_k^i\}_{k=1}^{n_i}]$$

El cual, es el subespacio generado por $\{\Phi_k^i\}_{k=1}^{n_i}$, donde;

- n_i es el número de nodos en la interfaz Γ proveniente del subdominio Ω_i.
- $\{x_k^j\}_{k=1}^{n_j}$ los nodos de la interfaz Ω que corresponden al subdominio Ω_j.

Las funciones $\Phi_k^{(j)}$, j=1,...,4; k=1,...,n, están definidas por:

$$\Phi_k^{(j)}(x) = \begin{cases} \frac{x - x_{k-1}^{(j)}}{x_k^{(j)} - x_{k-1}^{(j)}}, & \text{si } x \in [x_{k-1}^{(j)}, x_k^{(j)}] \\ \frac{x_{k+1}^{j} - x}{x_{k+1}^{(j)} - x_k^{(j)}}, & \text{si } x \in [x_x^{(j), x_{k+1}^{(j)}}] \\ 0, & \text{en otro caso} \end{cases}$$

Para $j = 2, ..., n_{j-1}$,

$$\Phi_1^{(j)}(x) = \begin{cases} \frac{x_2^{(j)} - x}{x_2^{(j)} - x_1^{(j)}}, & \text{si } x \in [x_1^{(j)}, x_2^{(j)}] \\ 0, & \text{en otro caso} \end{cases}$$

Para encontrar más detalles de este tipo de problemas, ver [11], [17].

5.1 Resultados

En esta sección presentaremos los resultados respecto al rendimiento de los siguientes esquemas iterativos:

- Cimmino
- Cimmino Gradientes Conjugados
- Cimmino Barzilai Borwein [11]

A continuación se realizará un estudio detallado de la velocidad en cada esquema iterativo de los mencionados anteriormente, observando el cumplimiento de los objetivos implícitos en los experimentos.

Estamos considerando para cada experimento una Tolerancia de (1×10^{-8}) además de un número fijo de iteraciones, que al momento de ser superado, detiene el proceso.

Todos los problemas realizados, son de la forma descrita en el capitulo anterior, y se compilaron en un procesador Intel Dual-Core de 2 GHz, usando MATLAB 7.

5.1.1 Experimento 1

Acá, buscamos comparar los diversos esquemas numéricos, para ello resolvimos 4 problemas tomados de la colección del CUTEr y 4 problemas de descomposición de dominios, descomponiendo la matriz B de cada problema fila a fila, (bloques tamaño 1).

A continuación mostraremos los gráficos tanto de velocidad de convergencia como el de tiempo de ejecución de cada aplicación, los cuales serán de gran utilidad para luego establecer conclusiones.

Aplicaciones del CUTEr

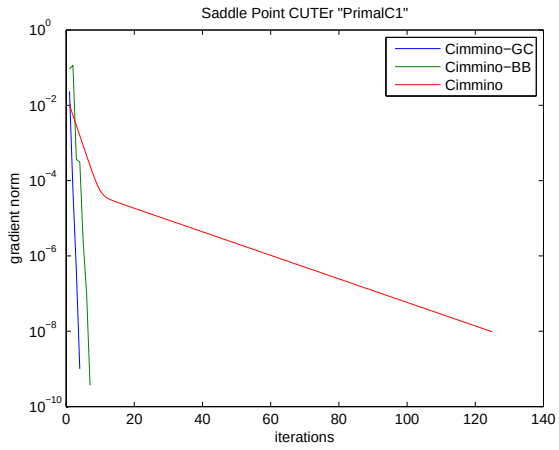

Figura 5.2: Matriz $A = 239 \times 239$ y $B = 9 \times 239$

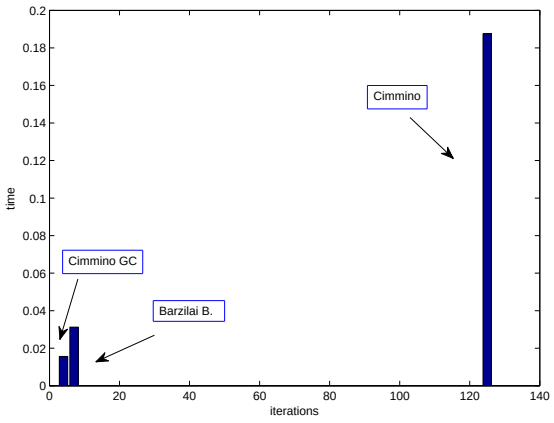

Figura 5.3: Tiempo de métodos iterativos aplicados a PrimalC1

Método	Iteraciones	Tiempo
Cimmino	125	0.187
Barzilai Borwein	7	0.031
GC Cimmino	4	0.015

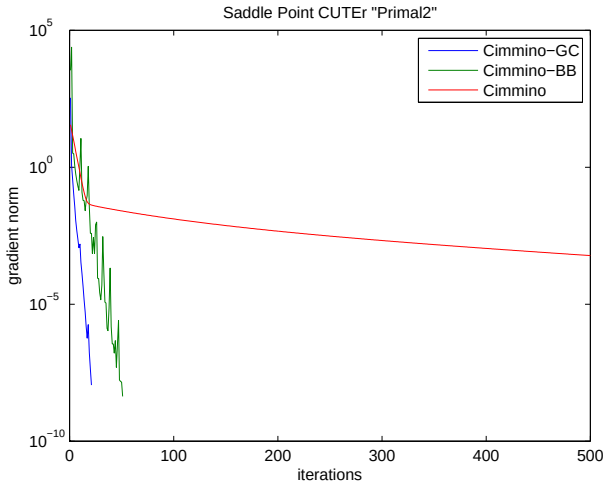

Figura 5.4: Matriz $A = 745 \times 745$ y $B = 96 \times 745$

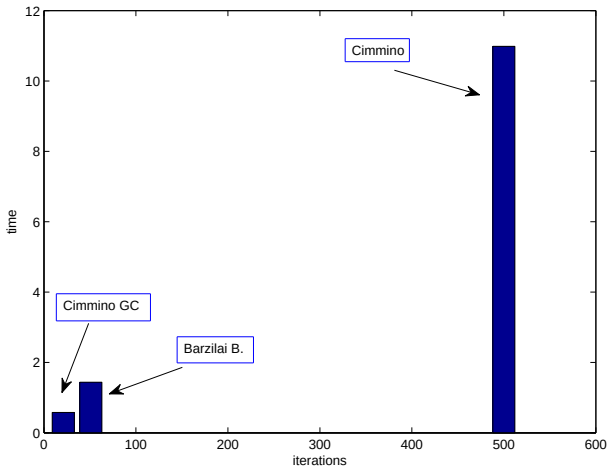

Figura 5.5: Tiempo de métodos iterativos aplicados a Primal2

Método	Iteraciones	Tiempo
Cimmino	> 500	$> 10{,}421$
Barzilai Borwein	51	1.328
GC Cimmino	21	0.578

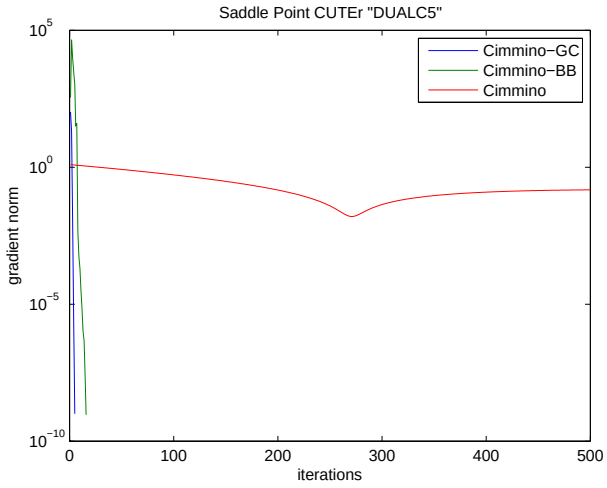

Figura 5.6: Matriz $A = 285 \times 285$ y $B = 278 \times 285$

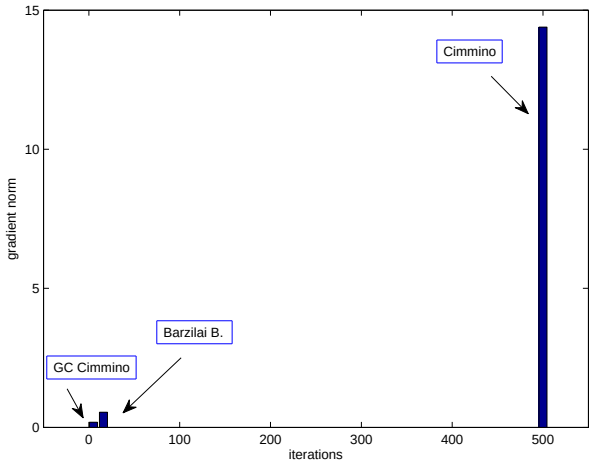

Figura 5.7: Tiempo de métodos iterativos aplicados a DUALC5

Método	Iteraciones	Tiempo
Cimmino	> 500	$> 14{,}350$
Barzilai Borwein	16	0,546
GC Cimmino	5	0.187

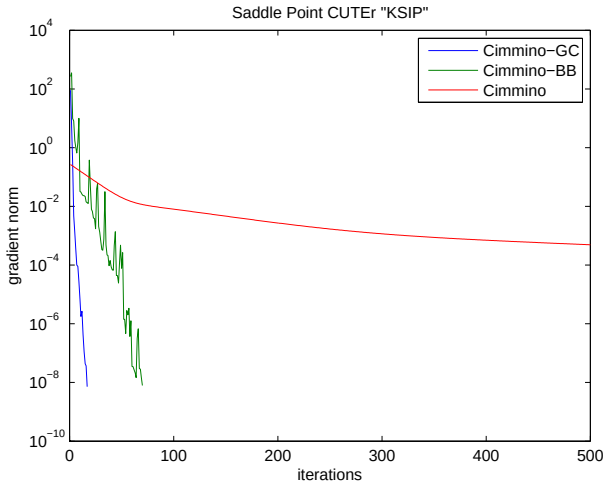

Figura 5.8: Matriz $A = 1021 \times 1021$ y $B = 1001 \times 1021$

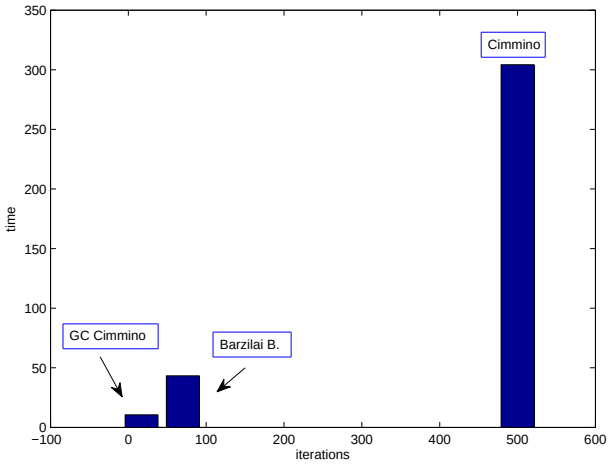

Figura 5.9: Tiempo de métodos iterativos aplicados a KSIP

Método	Iteraciones	Tiempo
Cimmino	> 500	$> 3{,}042 \times 10^2$
Barzilai Borwein	70	43,347
GC Cimmino	17	10.562

Descomposición de Dominios

Para estos problemas se dieron condiciones sobre los nodos que conforman el subdominio del dominio general después de haber realizado la descomposición respectiva.

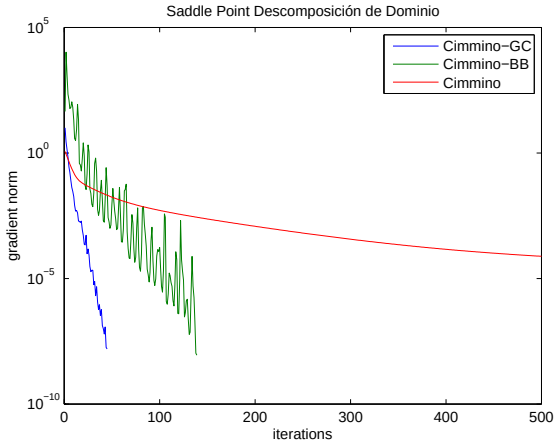

Figura 5.10: Matriz $A = 400 \times 400$ y $B = 39 \times 400$

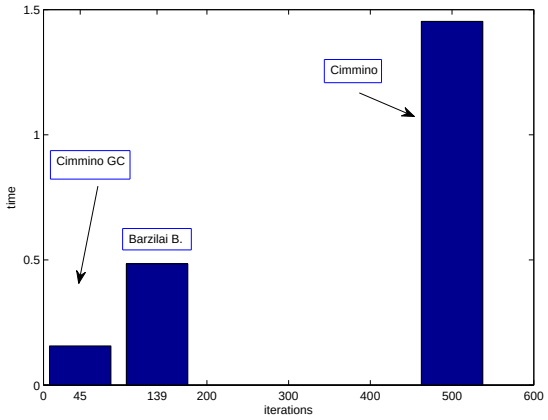

Figura 5.11: Tiempo de métodos iterativos aplicados a DD1

Método	Iteraciones	Tiempo
Cimmino	> 500	$> 1{,}406$
Barzilai Borwein	139	0.453
GC Cimmino	45	0.171

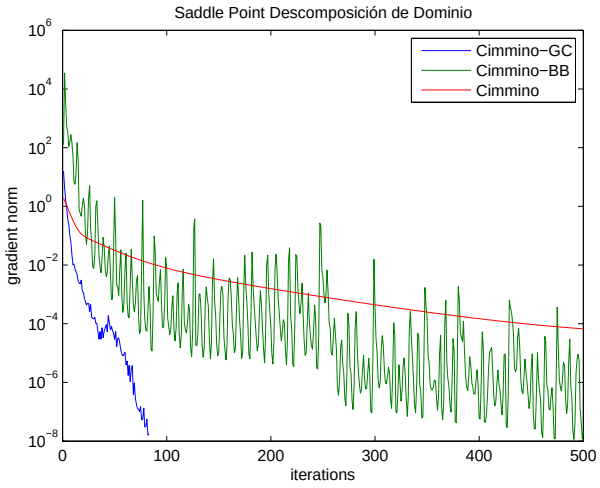

Figura 5.12: Matriz $A = 650 \times 650$ y $B = 65 \times 650$

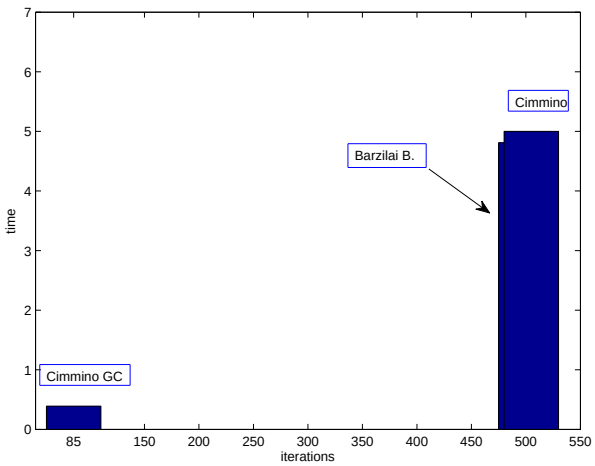

Figura 5.13: Tiempo de métodos iterativos aplicados a DD2

Método	Iteraciones	Tiempo
Cimmino	> 500	$> 4{,}812$
Barzilai Borwein	> 500	$> 6{,}171$
GC Cimmino	83	1.078

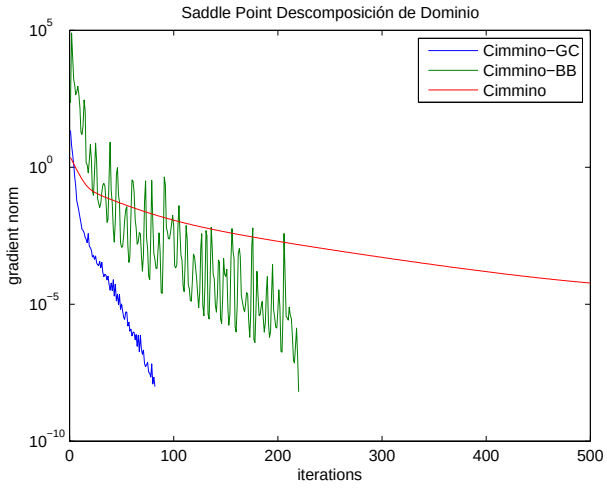

Figura 5.14: Matriz $A = 2500 \times 2500$ y $B = 99 \times 2500$

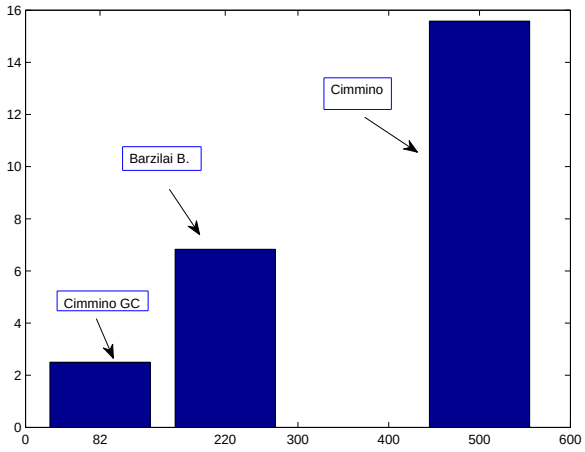

Figura 5.15: Tiempo de métodos iterativos aplicados a DD3

Método	Iteraciones	Tiempo
Cimmino	> 500	$> 13{,}093$
Barzilai Borwein	220	6,296
GC Cimmino	82	2.468

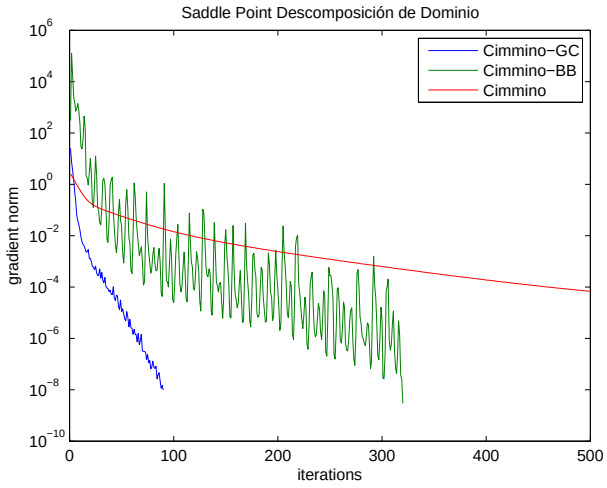

Figura 5.16: Matriz $A = 3600 \times 3600$ y $B = 119 \times 3600$

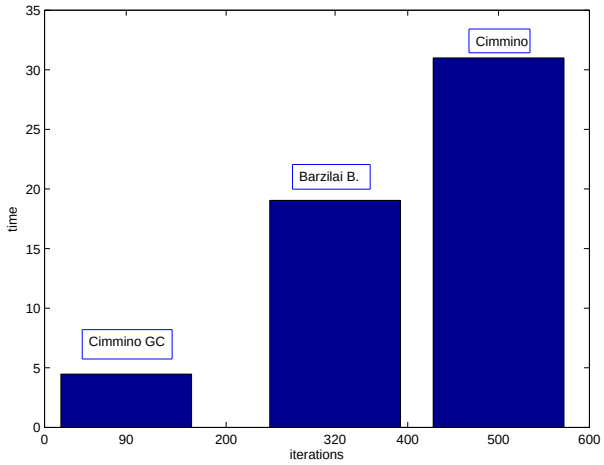

Figura 5.17: Tiempo de métodos iterativos aplicados a DD4

Método	Iteraciones	Tiempo
Cimmino	> 500	$> 32{,}406$
Barzilai Borwein	320	18,875
GC Cimmino	90	6.937

5.1.2 Experimento 2

Acá solo usamos el método con mejor aceleración (Cimmino GC), ahora veremos cuan rápido es su convergencia por bloques midiendo el tiempo de ejecución. Por cada problema se particionará la matriz B por bloques a fin de observar si la velocidad de convergencia depende o de la partición escogida, además de esto se incluirá el cociente entre el Tiempo de ejecución y el número de bloques para crear condiciones para su implementación en procesadores paralelos.

Aplicaciones del CUTEr

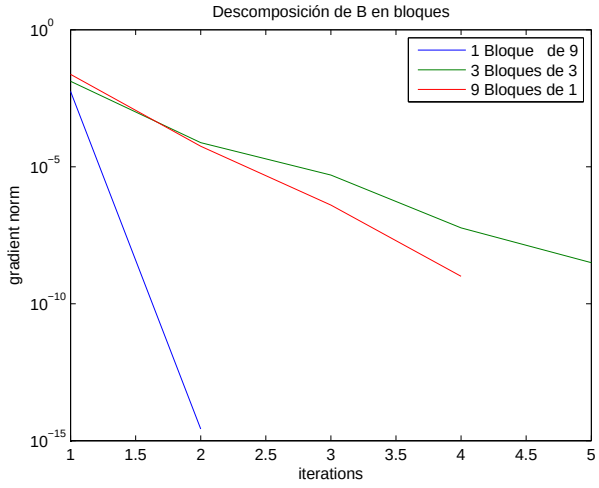

Figura 5.18: Descomposición por bloques PrimalC1 (CUTEr), $B_{9\times 239}$

Nº. de Bloques	Tamaño	Tiempo	Iteraciones	Tiempo/Nº. Bloques
1	9	0,203	2	0,203
3	3	0,265	5	0,088
9	1	0,031	4	$3{,}444 \times 10^{-3}$

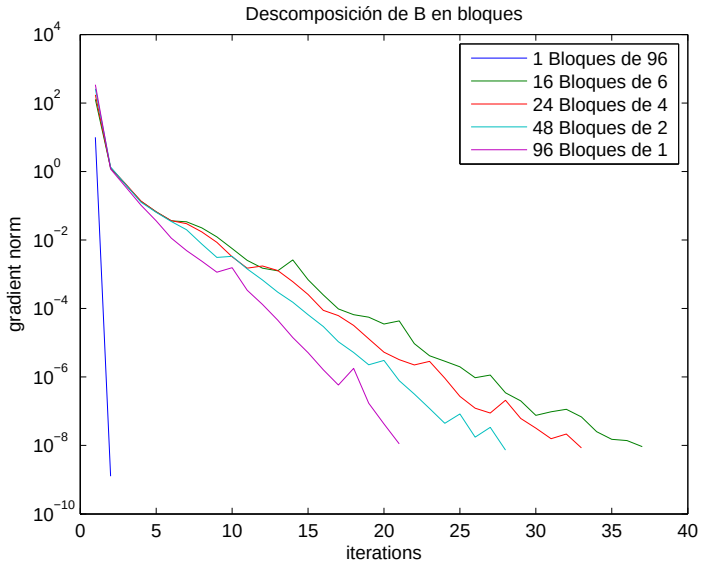

Figura 5.19: Descomposición por bloques Primal2 (CUTEr), $B_{96\times745}$

Nº. de Bloques	Tamaño	Tiempo	Iteraciones	Tiempo/Nº. Bloques
1	96	10,468	2	10,468
16	6	47,140	37	2,946
24	4	43,406	33	1,887
48	2	45,182	28	0,941
96	1	0,593	21	$6{,}17 \times 10^{-3}$

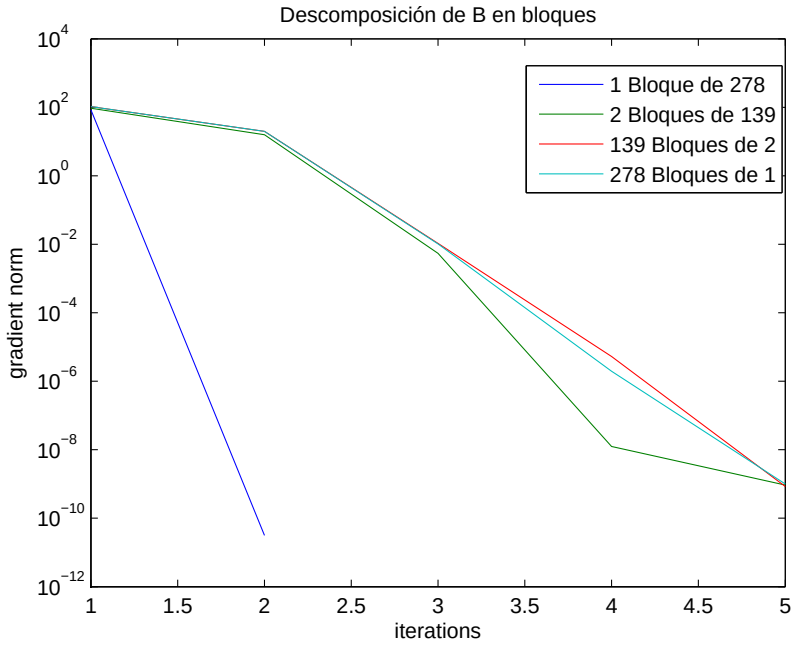

Figura 5.20: Descomposición por bloques DUALC5 (CUTEr), $B_{278\times285}$

Nº. de Bloques	Tamaño	Tiempo	Iteraciones	Tiempo/Nº. Bloques
1	278	1,046	2	1,046
2	139	2,156	5	1,078
139	2	18,859	5	0,135
278	1	0,187	5	$6{,}726 \times 10^{-4}$

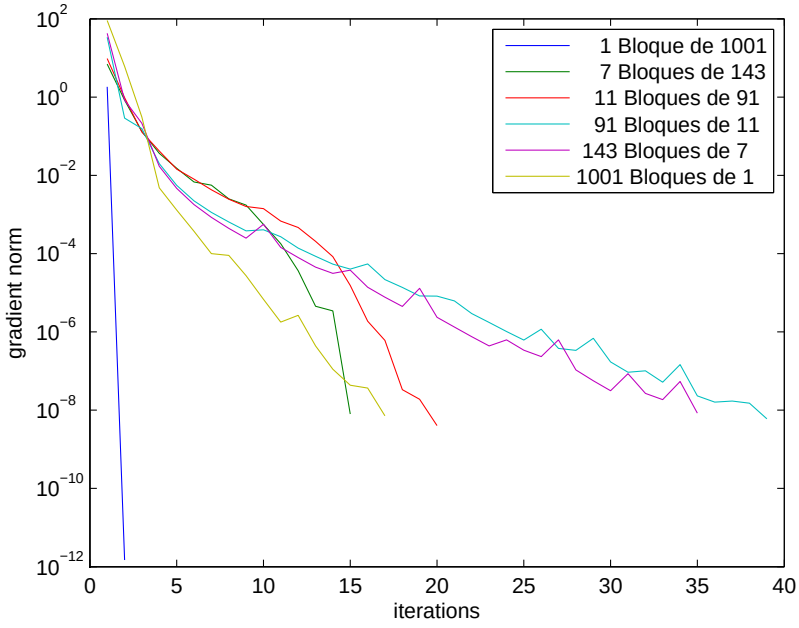

Figura 5.21: Descomposición por bloques KSIP (CUTEr), $B_{1001\times1021}$

Nº. de Bloques	Tamaño	Tiempo	Iteraciones	Tiempo/Nº. Bloques
1	1001	16,812	2	16,812
7	143	24,578	15	3,511
11	91	32,015	20	2,910
91	11	$2{,}190 \times 10^2$	39	2,395
143	7	$3{,}147 \times 10^2$	35	2,200
1001	1	10,562	17	0,010

Descomposición de Dominio

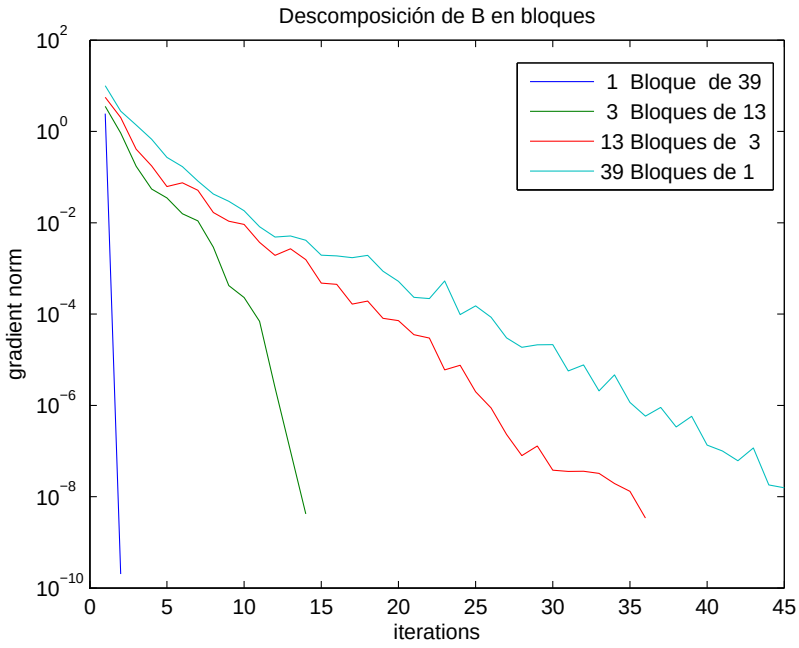

Figura 5.22: Descomposición en bloques D.D, $B_{39\times 400}$

Nº. de Bloques	Tamaño	Tiempo	Iteraciones	Tiempo/Nº. Bloques
1	39	0,178	2	0,178
3	13	0,984	14	0,328
13	3	4,250	36	0,326
39	1	0,125	45	$2{,}77 \times 10^{-2}$

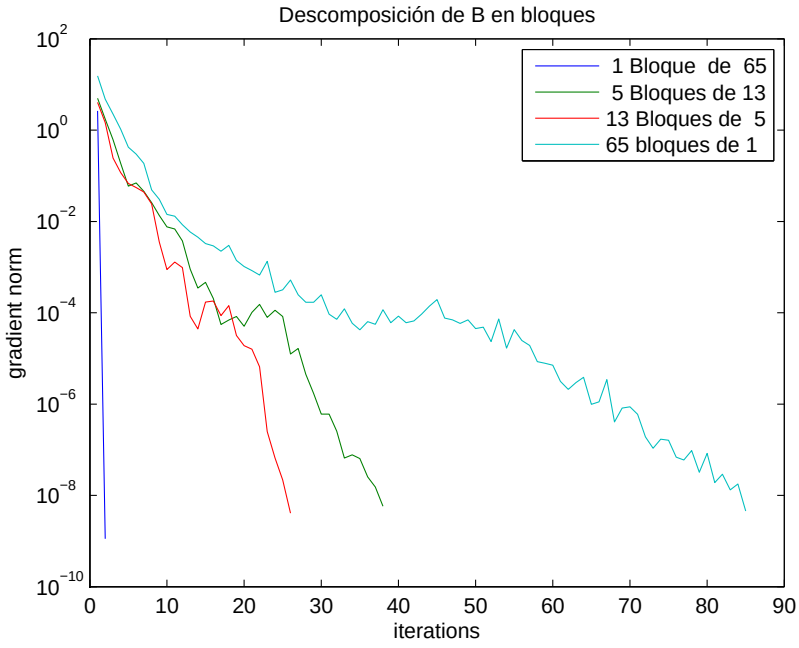

Figura 5.23: Descomposición en bloques D.D, $B_{65\times1250}$

Nº. de Bloques	Tamaño	Tiempo	Iteraciones	Tiempo/Nº. Bloques
1	65	0,771	2	0,771
5	13	18,218	38	3,643
13	5	7,434	26	0,571
65	1	1,078	83	0,016

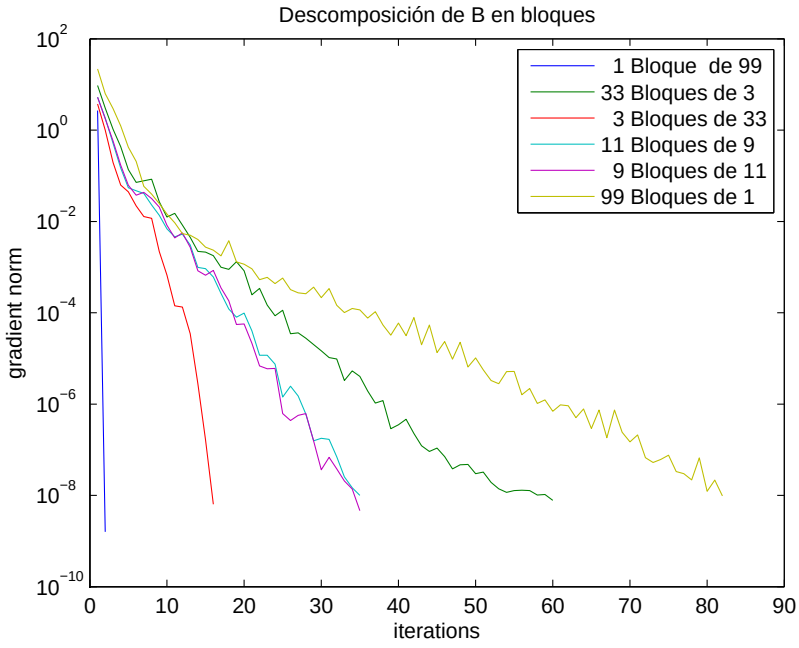

Figura 5.24: Descomposición en bloques D.D, $B_{99\times2500}$

Nº. de Bloques	Tamaño	Tiempo	Iteraciones	Tiempo/Nº. Bloques
1	99	2,821	2	2,821
33	3	95,855	60	2,904
3	33	8,390	16	2,796
11	9	38,703	35	3,518
9	11	30,875	35	3,430
99	1	2,391	82	0,024

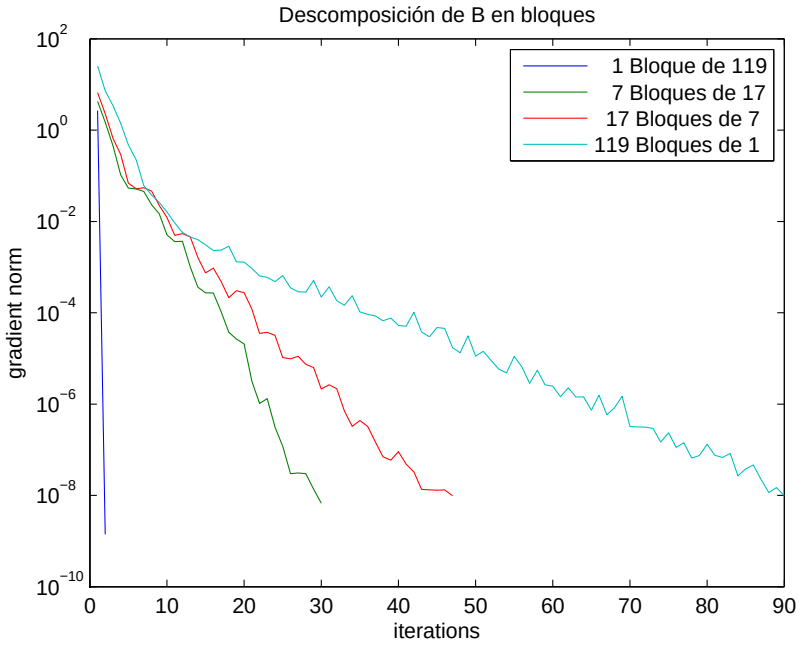

Figura 5.25: Descomposición en bloques D.D, $B_{119\times 3200}$

Nº. de Bloques	Tamaño	Tiempo	Iteraciones	Tiempo/Nº. Bloques
1	119	6,593	2	6,593
7	17	36,187	30	5,169
17	7	95,319	47	5,607
119	1	4,453	90	0,037

5.2 Comparación de los métodos

En el primer experimento se observa como el Método de Cimmino Gradientes Conjugados presenta una mejor velocidad de convergencia que la versión clásica o la versión Barzilai-Borwein. Como el costo por iteración no es mucho mayor al de los dos anteriores, el tiempo final de ejecución fue mucho menor para el método Cimmino-GC.

Respecto al segundo experimento, se constato que al particionar la matriz B fila a fila, el método de Cimmino-GC, realiza mucho más iteraciones pero su tiempo de ejecución es mucho menor al obtenido al particionar la matriz B en bloques de tamaño superior a 1, dado que al resolver problemas de saddle point tamaño n=1, es mucho más sencillo que resolver uno de tamaño mayor.

A pesar que al aumentar el número de bloques, tiende a aumentar el número de iteraciones del método, al final, el cociente entre el tiempo y el número de bloques tiende a ser mucho menor, por lo que se espera un buen desempeño en arquitecturas paralelas.

Conclusiones y Recomendaciones

Después de haber realizado y analizado los experimentos que sustentan esta investigación, presentamos las siguientes conclusiones:

- Es necesaria la aceleración del método de Cimmino en su versión clásica, en este caso, se procedió al uso de herramientas de optimización y se obtuvieron muy buenos resultados experimentales.

- La versión Barzilai - Borwein del método de Cimmino, también acelera la versión clásica de Cimmino, sin embargo la aceleración obtenida por el híbrido entre Gradientes Conjugados y Cimmino genera una mejor velocidad de convergencia, hecho que se puede constatar tanto con su numero de iteraciones como con su tiempo de ejecución.

- El paralelismo presente en el método de Cimmino versión clásica que también se preserva en el esquema de aceleración propuesto (GC Cimmino), puede ser aprovechado para realizar cálculos en bloques mediante procesadores separados.

Para dar continuidad a este estudio, se recomienda:

- Realizar experimentos con el método de Cimmino Gradiente Conjugado empleando procesadores paralelos.
- Tratar de realizar híbridos con otros esquemas a fin de obtener una velocidad de convergencia mucho más rápida que esta.

BIBLIOGRAFÍA

[1] Barzilai, J. and Borwein J.M., *Two-point step size Gradient Methods*, IMA Journal of numerical Analysis, volume 8, pp. 1-141, 1988.

[2] Benzi, M., *Gianfranco Cimmino's contributions to Numerical Mathematics*, Departament of Mathematics and Computer Science, Emory University, Atlanta, Georgia, pp. 7-9, 2005.

[3] Benzi, M. and Golub, G.H. and Liesen, J., *Numerical solution of saddle point problems*, Acta Numérica, Cambridge Univ. Press, volume 14, pp. 1-137, 2005.

[4] Bruzual R. y Domínguez M., *Espacios de Hilbert*, Guía del Laboratorio de Formas en Grupos, Facultad de Ciencias, Escuela de Matemática, Universidad Central de Venezuela, 2003

[5] Censor, Y., *Computational acceleration of projection algorithms for the linear best approximation problem*. Linear Algebra and its Applications, 2006.

[6] Cheney, W. and Goldstein, A., *Proximity maps for convex sets*, Proc. Amer. Math Soc., volume 10, pp. 448-450, 1959.

[7] Deutsch, F., *The Method of Alternating Orthogonal Projections* Approximation Theory, Splines Functions and Applications, Proc. NATO Adv. Study Inst., Maratea/ Italy, pp. 105-121, 1991.

[8] Deutsch, F., *Best Approximation in Inner Product Spaces*, Canadian Mathematical Society, Springer Verlarg, 2001.

[9] Deutsch, F., *The Angle Between Subspaces of a Hilbert Space*, Approximation theory, wavelets and applications, pp. 107-130, 1994.

[10] Escalante R. and Raydan M., *Alternating Projections Methods: Theory and Applications*, en preparación, 2005.

[11] Fariñas A., *Aceleración espectral del Método de Cimmino*, Universidad Central de Venezuela, Trabajo Especial de Grado, Facultad de Ciencias, Escuela de Matemática, Universidad Central de Venezuela, 2008.

[12] Gould, N.I.M. and Orban, D. and Toint, P.L., *CUTEr and SifDec: A constrained and unconstrained testing enviroment, revisited*, ACM Transactions on Mathemathical software (TOMS), volume 29, ACM Press New York, NY, USA, pp. 373-394, 2003.

[13] Gearhart, W. and Koshy, M., *Acceleration schemes for the method of alternating projections*, Journal of computational and applied mathematics, volume 26, Elsevier, pp. 235-249, 1989.

[14] Heath, M.T., *Scientific Computing: An Introductory Survey*, McGraw-Hill Science/Engineering/Math, 2002.

[15] Hernández-Ramos, L. M., *MAP acelerado con optimización*, documento en preparación, Facultad de Ciencias, Escuela de Computación, Universidad Central de Venezuela 2008.

[16] Hernández-Ramos, L. M., *Acelerated Cimmino's method for saddle point problems*, Reporte técnico RT-2008-06, Facultad de Ciencias, Escuela de Computación, Universidad Central de Venezuela, 2008.

[17] Hernández-Ramos, L. M., *Alternating oblique projections for coupled linear systems*, Numerical Algorithms 38, Facultad de Ciencias, Escuela de Computación, Universidad Central de Vewnezuela, p. 285-303, 2005.

[18] Kayalar, S. and Weinert, H.L., *Error bounds for the method of alternating projections*, Mathematics of Control, Signals, and Systems (MCSS), volume 1, Springer, p. 43-59, 1988.

[19] Kelley, C.T., *Iterative Methods for Linear and Nonlinear Equations*, Society for Industrial and Applied Mathematics Philadelphia, 1995.

[20] Raydan M., *Minimización Irrestricta a Gran Escala*, en preparacion, 2007.

[21] Saad, Y., *Iterative methods for sparse linear systems*, Society for Industrial Mathematics, 2003.

[22] Smith, K.T., and Solmon, D.C. and Wagner, S.L. *Practical and mathematical aspects of the problem of reconstructing objects from radiographs*, American Mathematical Society, volume 83, number 6, p. 1227, 1997.

[23] Sulbarán, G., *Algunas ideas sobre el Método de Proyecciones Alternantes (MAP)*, Trabajo especial de Grado, Facultad de Ciencias, Escuela de Matemática, Universidad Central de Venezuela, 2008.