



**UNIVERSIDAD NACIONAL AUTÓNOMA
DE MÉXICO**

INSTITUTO DE GEOFÍSICA

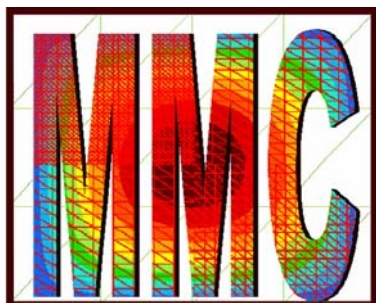
Y

**GRUPO DE MODELACIÓN
MATEMÁTICA Y COMPUTACIONAL**

Resolución de Grandes Sistemas de Ecuaciones Lineales

Antonio Carrillo Ledesma

<http://www.mmc.igeofcu.unam.mx/>



Modelación Matemática y Computacional

**INSTITUTO DE GEOFÍSICA
UNAM**

2006

Índice

1. Solución de Grandes Sistemas de Ecuaciones	2
1.1. Métodos Directos	3
1.2. Métodos Iterativos	4
2. Precondicionadores	9
2.1. ¿Qué es un Precondicionador?	9
2.2. Gradiente Conjugado Precondicionado	11
2.2.1. Precondicionador a Posteriori	13
2.2.2. Precondicionador a Priori	16
3. Bibliografía	19

1. Solución de Grandes Sistemas de Ecuaciones

En el capítulo anterior se discutió como proceder para transformar un problema de ecuaciones diferenciales parciales con valores en la frontera en un sistema algebraico de ecuaciones y así poder hallar la solución resolviendo el sistema de ecuaciones lineales que se pueden expresar en la forma matricial siguiente

$$\underline{\underline{A}}\underline{u} = \underline{b} \quad (1)$$

donde la matriz $\underline{\underline{A}}$ es bandada (muchos elementos son nulos) y en problemas reales tiene grandes dimensiones.

La elección del método específico para resolver el sistema de ecuaciones depende de las propiedades particulares de la matriz $\underline{\underline{A}}$, en las siguientes secciones examinaremos varios métodos y sus implicaciones en cuanto al costo computacional de la resolución del sistema de ecuaciones. En términos generales, si el problema de ecuaciones diferenciales parciales con valores en la frontera en dos dimensiones se discretiza usando una malla de $n \times m$ nodos, el sistema algebraico de ecuaciones asociado es del orden de $(n \times m)^2$, pero en general la matriz $\underline{\underline{A}}$ resultante para el tipo de problemas de interés en el presente trabajo es bandada y definida positiva, por ello es posible hacer uso de estas características para solucionar el sistema algebraico de ecuaciones de forma óptima.

Los métodos de resolución del sistema algebraico de ecuaciones $\underline{\underline{A}}\underline{u} = \underline{b}$ se clasifican en dos grandes grupos: los métodos directos y los métodos iterativos.

En los métodos directos la solución \underline{u} se obtiene en un número fijo de pasos y sólo están sujetos a los errores de redondeo. En los métodos iterativos, se realizan iteraciones para aproximarse a la solución \underline{u} aprovechando las características propias de la matriz $\underline{\underline{A}}$, tratando de usar un menor número de pasos que en un método directo.

Los métodos iterativos rara vez se usan para resolver sistemas lineales de dimensión pequeña (el concepto de dimensión pequeña es muy relativo), ya que el tiempo necesario para conseguir una exactitud satisfactoria rebasa el que requieren los métodos directos. Sin embargo, en el caso de sistemas grandes con un alto porcentaje de elementos cero, son eficientes tanto en el almacenamiento en la computadora como en el tiempo que se invierte en su solución. Por ésta razón al resolver éstos sistemas algebraicos de ecuaciones es preferible aplicar los métodos iterativos tales como son: Jacobi, Gauss-Seidel, sobrerrelajación sucesiva (SOR), etc. Para más información de éstos y otros métodos, así como pruebas en la velocidad de convergencia y precisión, pueden consultarse en [15], [5], [7], [9], [17], [24], [25] y [19].

Cabe hacer mención de que la mayoría del tiempo de cómputo necesario para resolver el problema de ecuaciones diferenciales parciales (EDP), es consumido en la solución del sistema algebraico de ecuaciones asociado a la discretización, por ello es determinante elegir aquel método numérico que minimice el tiempo invertido en este proceso.

1.1. Métodos Directos

En estos métodos, la solución \underline{u} se obtiene en un número fijo de pasos y sólo están sujetos a los errores de redondeo. Entre los métodos más importantes podemos encontrar: Eliminación Gausiana, descomposición LU, eliminación bandada y descomposición de Cholesky.

Los métodos antes mencionados, se colocaron en orden descendente en cuanto al consumo de recursos computacionales y ascendente en cuanto al aumento en su eficiencia; describiéndose a continuación:

Eliminación Gausiana Tal vez es el método más utilizado para encontrar la solución usando métodos directos. Este algoritmo sin embargo no es eficiente, ya que en general, un sistema de N ecuaciones requiere para su almacenaje en memoria de N^2 entradas para la matriz \underline{A} , pero cerca de $N^3/3 + O(N^2)$ multiplicaciones y $N^3/3 + O(N^2)$ adiciones para encontrar la solución siendo muy costoso computacionalmente.

La eliminación Gausiana se basa en la aplicación de operaciones elementales a renglones o columnas de tal forma que es posible obtener matrices equivalentes.

Escribiendo el sistema de N ecuaciones lineales con N incógnitas como

$$\sum_{j=1}^N a_{ij}^{(0)} x_j = a_{i,n+1}^{(0)}, \quad i = 1, 2, \dots, N \quad (2)$$

y si $a_{11}^{(0)} \neq 0$ y los pivotes $a_{ii}^{(i-1)}$, $i = 2, 3, \dots, N$ de las demás filas, que se obtienen en el curso de los cálculos, son distintos de cero, entonces, el sistema lineal anterior se reduce a la forma triangular superior (eliminación hacia adelante)

$$x_i + \sum_{j=i+1}^N a_{ij}^{(i)} x_j = a_{i,n+1}^{(i)}, \quad i = 1, 2, \dots, N$$

donde

$$\begin{aligned} k &= 1, 2, \dots, N; \{j = k + 1, \dots, N\} \\ a_{kj}^{(k)} &= \frac{a_{kj}^{(k-1)}}{a_{kk}^{(k-1)}}; \\ i &= k + 1, \dots, N + 1 \{ \\ a_{ij}^{(k)} &= a_{ij}^{(k-1)} - a_{kj}^{(k)} a_{ik}^{(k-1)} \} \} \end{aligned}$$

y las incógnitas se calculan por sustitución hacia atrás, usando las fórmulas

$$\begin{aligned} x_N &= a_{N,n+1}^{(N)}; \\ i &= N - 1, N - 2, \dots, 1 \\ x_i &= a_{i,n+1}^{(i)} - \sum_{j=i+1}^N a_{ij}^{(i)} x_j. \end{aligned}$$

En algunos casos nos interesa conocer $\underline{\underline{A}}^{-1}$, por ello si la eliminación se aplica a la matriz aumentada $\underline{\underline{A}} \mid \underline{\underline{I}}$ entonces la matriz $\underline{\underline{A}}$ de la matriz aumentada se convertirá en la matriz $\underline{\underline{I}}$ y la matriz $\underline{\underline{I}}$ de la matriz aumentada será $\underline{\underline{A}}^{-1}$. Así, el sistema $\underline{\underline{A}}\underline{\underline{u}} = \underline{\underline{b}}$ se transformará en $\underline{\underline{u}} = \underline{\underline{A}}^{-1}\underline{\underline{b}}$ obteniendo la solución de $\underline{\underline{u}}$.

Descomposición LU Sea U una matriz triangular superior obtenida de $\underline{\underline{A}}$ por eliminación bandada. Entonces $\underline{\underline{U}} = \underline{\underline{L}}^{-1}\underline{\underline{A}}$, donde $\underline{\underline{L}}$ es una matriz triangular inferior con unos en la diagonal. Las entradas de $\underline{\underline{L}}^{-1}$ pueden obtenerse de los coeficientes m_{ij} definidos en el método anterior y pueden ser almacenados estrictamente en las entradas de la diagonal inferior de $\underline{\underline{A}}$ ya que estas ya fueron eliminadas. Esto proporciona una factorización $\underline{\underline{LU}}$ de $\underline{\underline{A}}$ en la misma matriz $\underline{\underline{A}}$ ahorrando espacio de memoria.

El problema original $\underline{\underline{A}}\underline{\underline{u}} = \underline{\underline{b}}$ se escribe como $\underline{\underline{LU}}\underline{\underline{u}} = \underline{\underline{b}}$ y se reduce a la solución sucesiva de los sistemas lineales triangulares

$$\underline{\underline{L}}\underline{\underline{y}} = \underline{\underline{b}} \quad \text{y} \quad \underline{\underline{U}}\underline{\underline{u}} = \underline{\underline{y}}.$$

La descomposición $\underline{\underline{LU}}$ requiere también $N^3/3$ operaciones aritméticas para la matriz llena, pero sólo Nb^2 operaciones aritméticas para la matriz con un ancho de banda de b siendo esto muy económico computacionalmente.

Nótese que para una matriz no singular $\underline{\underline{A}}$, la eliminación de Gaussiana (sin redondear filas y columnas) es equivalente a la factorización $\underline{\underline{LU}}$.

Eliminación Bandada Cuando se usa la ordenación natural de los nodos, la matriz $\underline{\underline{A}}$ que se genera es bandada, por ello se puede ahorrar considerable espacio de almacenamiento en ella. Este algoritmo consiste en triangular a la matriz $\underline{\underline{A}}$ por eliminación hacia adelante operando sólo sobre las entradas dentro de la banda central no cero. Así el renglón j es multiplicado por $m_{ij} = a_{ij}/a_{jj}$ y el resultado es restado al renglón i para $i = j + 1, j + 2, \dots$

El resultado es una matriz triangular superior $\underline{\underline{U}}$ que tiene ceros abajo de la diagonal en cada columna. Así, es posible resolver el sistema resultante al sustituir en forma inversa las incógnitas.

Descomposición de Cholesky Cuando la matriz es simétrica y definida positiva, se obtiene la descomposición $\underline{\underline{LU}}$ de la matriz $\underline{\underline{A}}$, así $\underline{\underline{A}} = \underline{\underline{LDU}} = \underline{\underline{LDL}}^T$ donde $\underline{\underline{D}} = \text{diag}(\underline{\underline{U}})$ es la diagonal con entradas positivas. La mayor ventaja de esta descomposición es que, en el caso en que es aplicable, el costo de cómputo es sustancialmente reducido, requiere de $N^3/6$ multiplicaciones y $N^3/6$ adiciones.

1.2. Métodos Iterativos

En estos métodos se realizan iteraciones para aproximarse a la solución $\underline{\underline{u}}$ aprovechando las características propias de la matriz $\underline{\underline{A}}$, tratando de usar un menor número de pasos que en un método directo, para más información de estos y otros métodos ver [15] y [24].

Un método iterativo en el cual se resuelve el sistema lineal

$$\underline{\underline{A}}\underline{u} = \underline{b} \quad (3)$$

comienza con una aproximación inicial \underline{u}^0 a la solución \underline{u} y genera una sucesión de vectores $\{u^k\}_{k=1}^{\infty}$ que converge a \underline{u} . Los métodos iterativos traen consigo un proceso que convierte el sistema $\underline{\underline{A}}\underline{u} = \underline{b}$ en otro equivalente de la forma $\underline{u} = \underline{T}\underline{u} + \underline{c}$ para alguna matriz fija \underline{T} y un vector \underline{c} . Luego de seleccionar el vector inicial \underline{u}^0 la sucesión de los vectores de la solución aproximada se genera calculando

$$\underline{u}^k = \underline{T}\underline{u}^{k-1} + \underline{c} \quad \forall k = 1, 2, 3, \dots \quad (4)$$

La convergencia a la solución la garantiza el siguiente teorema cuya solución puede verse en [25].

Teorema 1 *Si $\|\underline{T}\| < 1$, entonces el sistema lineal $\underline{u} = \underline{T}\underline{u} + \underline{c}$ tiene una solución única \underline{u}^* y las iteraciones \underline{u}^k definidas por la fórmula $\underline{u}^k = \underline{T}\underline{u}^{k-1} + \underline{c} \quad \forall k = 1, 2, 3, \dots$ convergen hacia la solución exacta \underline{u}^* para cualquier aproximación lineal \underline{u}^0 .*

Notemos que mientras menor sea la norma de la matriz \underline{T} , más rápida es la convergencia, en el caso cuando $\|\underline{T}\|$ es menor que uno, pero cercano a uno, la convergencia es muy lenta y el número de iteraciones necesario para disminuir el error depende significativamente del error inicial. En este caso, es deseable proponer al vector inicial \underline{u}^0 de forma tal que se mínimo el error inicial. Sin embargo, la elección de dicho vector no tiene importancia si la $\|\underline{T}\|$ es pequeña ya que la convergencia es rápida.

Como es conocido, la velocidad de convergencia de los métodos iterativos dependen de las propiedades espectrales de la matriz de coeficientes del sistema de ecuaciones, cuando el operador diferencial \mathcal{L} de la ecuación del problema a resolver es auto-adjunto se obtiene una matriz simétrica y positivo definida y el número de condicionamiento de la matriz $\underline{\underline{A}}$, es por definición

$$cond(\underline{\underline{A}}) = \frac{\lambda_{\max}}{\lambda_{\min}} \geq 1 \quad (5)$$

donde λ_{\max} y λ_{\min} es el máximo y mínimo de los eigenvalores de la matriz $\underline{\underline{A}}$. Si el número de condicionamiento es cercano a 1 los métodos numéricos al solucionar el problema convergerá en pocas iteraciones, en caso contrario se requerirán muchas iteraciones. Frecuentemente al usar el método de elemento finito se tiene una velocidad de convergencia de $O\left(\frac{1}{h^2}\right)$ y en el caso de métodos de descomposición de dominio se tiene una velocidad de convergencia de $O\left(\frac{1}{h}\right)$ en el mejor de los casos, donde h es la máxima distancia de separación entre nodos continuos de la partición, es decir, que poseen una pobre velocidad de convergencia cuando $h \rightarrow 0$, para más detaller ver [2].

Entre los métodos más usados para el tipo de problemas tratados en el presente trabajo podemos encontrar: Jacobi, Gauss-Seidel, Richardson, relajación sucesiva, gradiente conjugado, gradiente conjugado preconditionado.

Los métodos antes mencionados se colocaron en orden descendente en cuanto al consumo de recursos computacionales y ascendente en cuanto al aumento en la eficiencia en su desempeño, describiéndose a continuación:

Jacobi Si todos los elementos de la diagonal principal de la matriz $\underline{\underline{A}}$ son diferentes de cero $a_{ii} \neq 0$ para $i = 1, 2, \dots, n$. Podemos dividir la i -ésima ecuación del sistema lineal (3) por a_{ii} para $i = 1, 2, \dots, n$, y después trasladamos todas las incógnitas, excepto x_i , a la derecha, se obtiene el sistema equivalente

$$\underline{u} = \underline{\underline{B}}\underline{u} + \underline{d}$$

donde

$$d_i = \frac{b_i}{a_{ii}} \quad \text{y} \quad B = \{b_{ij}\} = \begin{cases} -\frac{a_{ij}}{a_{ii}} & \text{si } j \neq i \\ 0 & \text{si } j = i \end{cases}.$$

Las iteraciones del método de Jacobi están definidas por la fórmula

$$x_i = \sum_{j=1}^n b_{ij}x_j^{(k-1)} + d_i$$

donde $x_i^{(0)}$ son arbitrarias ($i = 1, 2, \dots, n; k = 1, 2, \dots$).

También el método de Jacobi se puede expresar en términos de matrices. Supongamos por un momento que la matriz $\underline{\underline{A}}$ tiene la diagonal unitaria, esto es $\text{diag}(\underline{\underline{A}}) = \underline{\underline{I}}$. Si descomponemos $\underline{\underline{A}} = \underline{\underline{I}} - \underline{\underline{B}}$, entonces el sistema dado por la Ecs. (3) se puede reescribir como

$$(\underline{\underline{I}} - \underline{\underline{B}})\underline{u} = \underline{b}.$$

Para la primera iteración asumimos que $\underline{k} = \underline{b}$; entonces la última ecuación se escribe como $\underline{u} = \underline{\underline{B}}\underline{u} + \underline{k}$. Tomando una aproximación inicial \underline{u}^0 , podemos obtener una mejor aproximación reemplazando \underline{u} por la más reciente aproximación de \underline{u}^m . Esta es la idea que subyace en el método Jacobi. El proceso iterativo queda como

$$\underline{u}^{m+1} = \underline{\underline{B}}\underline{u}^m + \underline{k}. \quad (6)$$

La aplicación del método a la ecuación de la forma $\underline{\underline{A}}\underline{u} = \underline{b}$, con la matriz $\underline{\underline{A}}$ no cero en los elementos diagonales, se obtiene multiplicando la Ec. (3) por $D^{-1} = [\text{diag}(\underline{\underline{A}})]^{-1}$ obteniendo

$$\underline{\underline{B}} = \underline{\underline{I}} - \underline{\underline{D}}^{-1}\underline{\underline{A}}, \quad \underline{k} = \underline{\underline{D}}^{-1}\underline{b}.$$

Gauss-Seidel Este método es una modificación del método Jacobi, en el cual una vez obtenido algún valor de \underline{u}^{m+1} , este es usado para obtener el resto de los valores utilizando los valores más actualizados de \underline{u}^{m+1} . Así, la Ec. (6) puede ser escrita como

$$u_i^{m+1} = \sum_{j < i} b_{ij}u_j^{m+1} + \sum_{j > i} b_{ij}u_j^m + k_i. \quad (7)$$

Notemos que el método Gauss-Siedel requiere el mismo número de operaciones aritméticas por iteración que el método de Jacobi. Este método se escribe en forma matricial como

$$\underline{u}^{m+1} = \underline{E}\underline{u}^{m+1} + \underline{F}\underline{u}^m + \underline{k} \quad (8)$$

donde \underline{E} y \underline{F} son las matrices triangular superior e inferior respectivamente. Este método mejora la convergencia con respecto al método de Jacobi en un factor aproximado de 2.

Richardson Escribiendo el método de Jacobi como

$$\underline{u}^{m+1} - \underline{u}^m = \underline{b} - \underline{A}\underline{u}^m \quad (9)$$

entonces el método Richardson se genera al incorporar la estrategia de sobrerelajación de la forma siguiente

$$\underline{u}^{m+1} = \underline{u}^m + \omega (\underline{b} - \underline{A}\underline{u}^m). \quad (10)$$

El método de Richardson se define como

$$\underline{u}^{m+1} = (\underline{I} - \omega \underline{A}) \underline{u}^m + \omega \underline{b} \quad (11)$$

en la práctica encontrar el valor de ω puede resultar muy costoso computacionalmente y las diversas estrategias para encontrar ω dependen de las características propias del problema, pero este método con un valor ω óptimo resulta mejor que el método de Gauss-Seidel.

Relajación Sucesiva Partiendo del método de Gauss-Siedel y sobrerelajando este esquema, obtenemos

$$u_i^{m+1} = (1 - \omega) u_i^m + \omega \left[\sum_{j=1}^{i-1} b_{ij} u_j^{m+1} + \sum_{j=i+1}^N b_{ij} u_j^m + k_i \right] \quad (12)$$

y cuando la matriz \underline{A} es simétrica con entradas en la diagonal positivas, éste método converge si y sólo si \underline{A} es definida positiva y $\omega \in (0, 2)$. En la práctica encontrar el valor de ω puede resultar muy costoso computacionalmente y las diversas estrategias para encontrar ω dependen de las características propias del problema.

Gradiente Conjugado El método del gradiente conjugado ha recibido mucha atención en su uso al resolver ecuaciones diferenciales parciales y ha sido ampliamente utilizado en años recientes por la notoria eficiencia al reducir considerablemente en número de iteraciones necesarias para resolver el sistema algebraico de ecuaciones. Aunque los pioneros de este método fueron Hestenes y Stiefel (1952), el interés actual arranca a partir de que Reid (1971) lo planteara como

un método iterativo, que es la forma en que se le usa con mayor frecuencia en la actualidad, esta versión está basada en el desarrollo hecho en [9].

La idea básica en que descansa el método del gradiente conjugado consiste en construir una base de vectores ortogonales y utilizarla para realizar la búsqueda de la solución en forma más eficiente. Tal forma de proceder generalmente no sería aconsejable porque la construcción de una base ortogonal utilizando el procedimiento de Gramm-Schmidt requiere, al seleccionar cada nuevo elemento de la base, asegurar su ortogonalidad con respecto a cada uno de los vectores construidos previamente. La gran ventaja del método de gradiente conjugado radica en que cuando se utiliza este procedimiento, basta con asegurar la ortogonalidad de un nuevo miembro con respecto al último que se ha construido, para que automáticamente esta condición se cumpla con respecto a todos los anteriores.

En el algoritmo de gradiente conjugado (CGM), se toman como datos de entrada al sistema

$$\underline{\underline{A}}u = \underline{b} \quad (13)$$

el vector de búsqueda inicial \underline{u}^0 y se calcula $\underline{r}^0 = \underline{b} - \underline{\underline{A}}u^0$, $\underline{p}^0 = \underline{r}^0$, quedando el método esquemáticamente como:

$$\begin{aligned} \beta^{k+1} &= \frac{\underline{\underline{A}}p^k \cdot \underline{r}^k}{\underline{\underline{A}}p^k \cdot \underline{p}^k} \\ \underline{p}^{k+1} &= \underline{r}^k - \beta^{k+1} \underline{p}^k \\ \alpha^{k+1} &= \frac{\underline{r}^k \cdot \underline{r}^k}{\underline{\underline{A}}p^{k+1} \cdot \underline{p}^{k+1}} \end{aligned} \quad (14)$$

$$\begin{aligned} \underline{u}^{k+1} &= \underline{u}^k + \alpha^{k+1} \underline{p}^{k+1} \\ \underline{r}^{k+1} &= \underline{r}^k - \alpha^{k+1} \underline{\underline{A}}p^{k+1}. \end{aligned}$$

Si denotamos $\{\lambda_i, V_i\}_{i=1}^N$ como las eigensoluciones de $\underline{\underline{A}}$, i.e. $\underline{\underline{A}}V_i = \lambda_i V_i$, $i = 1, 2, \dots, N$. Ya que la matriz $\underline{\underline{A}}$ es simétrica, los eigenvalores son reales y podemos ordenarlos por $\lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_N$. Definimos el número de condición por $Cond(\underline{\underline{A}}) = \lambda_N/\lambda_1$ y la norma de la energía asociada a $\underline{\underline{A}}$ por $\|\underline{u}\|_{\underline{\underline{A}}}^2 = \underline{u} \cdot \underline{\underline{A}}u$ entonces

$$\|\underline{u} - \underline{u}^k\|_{\underline{\underline{A}}} \leq \|\underline{u} - \underline{u}^0\|_{\underline{\underline{A}}} \left[\frac{1 - \sqrt{Cond(\underline{\underline{A}})}}{1 + \sqrt{Cond(\underline{\underline{A}})}} \right]^{2k}. \quad (15)$$

El siguiente teorema nos da idea del espectro de convergencia del sistema $\underline{\underline{A}}u = \underline{b}$ para el método de gradiente conjugado.

Teorema 2 Sea $\kappa = cond(\underline{\underline{A}}) = \frac{\lambda_{\max}}{\lambda_{\min}} \geq 1$, entonces el método de gradiente

conjugado satisface la $\underline{\underline{A}}$ -norma del error dado por

$$\frac{\|e^n\|}{\|e^0\|} \leq \frac{2}{\left[\left(\frac{\sqrt{\kappa}+1}{\sqrt{\kappa}-1}\right)^n + \left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^{-n}\right]} \leq 2 \left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^n \quad (16)$$

donde $\underline{e}^m = \underline{u} - \underline{u}^m$ del sistema $\underline{\underline{A}}\underline{u} = \underline{b}$.

Notemos que para κ grande se tiene que

$$\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1} \simeq 1 - \frac{2}{\sqrt{\kappa}} \quad (17)$$

tal que

$$\|\underline{e}^n\|_{\underline{\underline{A}}} \simeq \|\underline{e}^0\|_{\underline{\underline{A}}} \exp\left(-2\frac{n}{\sqrt{\kappa}}\right)$$

de lo anterior podemos esperar un espectro de convergencia del orden de $O(\sqrt{\kappa})$ iteraciones, para mayor referencia ver [25].

2. Precondicionadores

Una vía que permite mejorar la eficiencia de los métodos iterativos consiste en transformar al sistema de ecuaciones en otro equivalente, en el sentido de que posea la misma solución del sistema original pero que a su vez tenga mejores condiciones espectrales. Esta transformación se conoce como precondicionamiento y consiste en aplicar al sistema de ecuaciones una matriz conocida como precondicionador encargada de realizar el mejoramiento del número de condicionamiento.

Una amplia clase de precondicionadores han sido propuestos basados en las características algebraicas de la matriz del sistema de ecuaciones, mientras que por otro lado también existen precondicionadores desarrollados a partir de las características propias del problema que lo origina, un estudio más completo puede encontrarse en [2] y [17].

2.1. ¿Qué es un Precondicionador?

De una manera formal podemos decir que un precondicionador consiste en construir una matriz $\underline{\underline{C}}$, la cuál es una aproximación en algún sentido de la matriz $\underline{\underline{A}}$ del sistema $\underline{\underline{A}}\underline{u} = \underline{b}$, de manera tal que si multiplicamos ambos miembros del sistema de ecuaciones original por $\underline{\underline{C}}^{-1}$ obtenemos el siguiente sistema

$$\underline{\underline{C}}^{-1}\underline{\underline{A}}\underline{u} = \underline{\underline{C}}^{-1}\underline{b} \quad (18)$$

donde el número de condicionamiento de la matriz del sistema transformado $\underline{\underline{C}}^{-1}\underline{\underline{A}}$ debe ser menor que el del sistema original, es decir

$$\text{Cond}(\underline{\underline{C}}^{-1}\underline{\underline{A}}) < \text{Cond}(\underline{\underline{A}}), \quad (19)$$

dicho de otra forma un preconditionador es una inversa aproximada de la matriz original

$$\underline{\underline{C}}^{-1} \simeq \underline{\underline{A}}^{-1} \quad (20)$$

que en el caso ideal $\underline{\underline{C}}^{-1} = \underline{\underline{A}}^{-1}$ el sistema convergería en una sola iteración, pero el coste computacional del cálculo de $\underline{\underline{A}}^{-1}$ equivaldría a resolver el sistema por un método directo. Se sugiere que $\underline{\underline{C}}$ sea una matriz lo más próxima a $\underline{\underline{A}}$ sin que su determinación suponga un coste computacional elevado.

Dependiendo de la forma de plantear el producto de $\underline{\underline{C}}^{-1}$ por la matriz del sistema obtendremos distintas formas de preconditionamiento, estas son:

$\underline{\underline{C}}^{-1} \underline{\underline{A}} x = \underline{\underline{C}}^{-1} b$	Precondicionamiento por la izquierda
$\underline{\underline{A}} \underline{\underline{C}}^{-1} \underline{\underline{C}} x = b$	Precondicionamiento por la derecha
$\underline{\underline{C}}_1^{-1} \underline{\underline{A}} \underline{\underline{C}}_2^{-1} \underline{\underline{C}}_2 x = \underline{\underline{C}}_1^{-1} b$	Precondicionamiento por ambos lados si $\underline{\underline{C}}$ puede factorizarse como $\underline{\underline{C}} = \underline{\underline{C}}_1 \underline{\underline{C}}_2$.

El uso de un preconditionador en un método iterativo provoca que se incurra en un costo de cómputo extra debido a que inicialmente se construye y luego se debe aplicar en cada iteración. Teniéndose que encontrar un balance entre el costo de construcción y aplicación del preconditionador versus la ganancia en velocidad en convergencia del método.

Ciertos preconditionadores necesitan poca o ninguna fase de construcción, mientras que otros pueden requerir de un trabajo substancial en esta etapa. Por otra parte la mayoría de los preconditionadores requieren en su aplicación un monto de trabajo proporcional al número de variables; esto implica que se multiplica el trabajo por iteración en un factor constante.

De manera resumida un buen preconditionador debe reunir las siguientes características:

- i) Al aplicar un preconditionador $\underline{\underline{C}}$ al sistema original de ecuaciones $\underline{\underline{A}} u = b$, se debe reducir el número de iteraciones necesarias para que la solución aproximada tenga la convergencia a la solución exacta con una exactitud ε prefijada.
- ii) La matriz $\underline{\underline{C}}$ debe ser fácil de calcular, es decir, el costo computacional de la construcción del preconditionador debe ser pequeño comparado con el costo total de resolver el sistema de ecuaciones $\underline{\underline{A}} u = b$.
- iii) El sistema $\underline{\underline{C}} z = r$ debe ser fácil de resolver. Esto debe interpretarse de dos maneras:
 - a) El monto de operaciones por iteración debido a la aplicación del preconditionador $\underline{\underline{C}}$ debe ser pequeño o del mismo orden que las que se requerirían sin preconditionamiento. Esto es importante si se trabaja en máquinas secuenciales.
 - b) El tiempo requerido por iteración debido a la aplicación del preconditionador debe ser pequeño.

En computadoras paralelas es importante que la aplicación del preconditionador sea paralelizable, lo cual eleva su eficiencia, pero debe de existir un balance entre la eficacia de un preconditionador en el sentido clásico y su eficiencia en paralelo ya que la mayoría de los preconditionadores tradicionales tienen un componente secuencial grande.

El método de gradiente conjugado por si mismo no permite el uso de preconditionadores, pero con una pequeña modificación en el producto interior usado en el método, da origen al método de gradiente conjugado preconditionado que a continuación detallaremos.

2.2. Gradiente Conjugado Precondicionado

Cuando la matriz $\underline{\underline{A}}$ es simétrica y definida positiva se puede escribir como

$$\lambda_1 \leq \frac{\underline{u} \underline{\underline{A}} \cdot \underline{u}}{\underline{u} \cdot \underline{u}} \leq \lambda_n \quad (21)$$

y tomando la matriz $\underline{\underline{C}}^{-1}$ como un preconditionador de $\underline{\underline{A}}$ con la condición de que

$$\lambda_1 \leq \frac{\underline{u} \underline{\underline{C}}^{-1} \underline{\underline{A}} \cdot \underline{u}}{\underline{u} \cdot \underline{u}} \leq \lambda_n \quad (22)$$

entonces la Ec. (13) se puede escribir como

$$\underline{\underline{C}}^{-1} \underline{\underline{A}} \underline{u} = \underline{\underline{C}}^{-1} \underline{b} \quad (23)$$

donde $\underline{\underline{C}}^{-1} \underline{\underline{A}}$ es también simétrica y definida positiva en el producto interior $\langle \underline{u}, \underline{v} \rangle = \underline{u} \cdot \underline{\underline{C}} \underline{v}$, porque

$$\begin{aligned} \langle \underline{u}, \underline{\underline{C}}^{-1} \underline{\underline{A}} \underline{v} \rangle &= \underline{u} \cdot \underline{\underline{C}} (\underline{\underline{C}}^{-1} \underline{\underline{A}} \underline{v}) \\ &= \underline{u} \cdot \underline{\underline{A}} \underline{v} \end{aligned} \quad (24)$$

que por hipótesis es simétrica y definida positiva en ese producto interior.

La elección del producto interior $\langle \cdot, \cdot \rangle$ quedará definido como

$$\langle \underline{u}, \underline{v} \rangle = \underline{u} \cdot \underline{\underline{C}}^{-1} \underline{\underline{A}} \underline{v} \quad (25)$$

por ello las Ecs. (14[1]) y (14[3]), se convierten en

$$\alpha^{k+1} = \frac{\underline{r}^k \cdot \underline{r}^k}{\underline{p}^{k+1} \cdot \underline{\underline{C}}^{-1} \underline{p}^{k+1}} \quad (26)$$

y

$$\beta^{k+1} = \frac{\underline{p}^k \cdot \underline{\underline{C}}^{-1} \underline{r}^k}{\underline{p}^k \cdot \underline{\underline{A}} \underline{p}^k} \quad (27)$$

generando el método de gradiente conjugado preconditionado con preconditionador $\underline{\underline{C}}^{-1}$. Es necesario hacer notar que los métodos gradiente conjugado y

gradiente conjugado preconditionado sólo difieren en la elección del producto interior.

Para el método de gradiente conjugado preconditionado, los datos de entrada son un vector de búsqueda inicial \underline{u}^0 y el preconditionador $\underline{\underline{C}}^{-1}$. Calculándose $\underline{r}^0 = \underline{b} - \underline{\underline{A}}\underline{u}^0$, $\underline{p} = \underline{\underline{C}}^{-1}\underline{r}^0$, quedando el método esquemáticamente como:

$$\begin{aligned}
\beta^{k+1} &= \frac{\underline{p}^k \cdot \underline{\underline{C}}^{-1}\underline{r}^k}{\underline{p}^k \cdot \underline{\underline{A}}\underline{p}^k} \\
\underline{p}^{k+1} &= \underline{r}^k - \beta^{k+1}\underline{p}^k \\
\alpha^{k+1} &= \frac{\underline{r}^k \cdot \underline{r}^k}{\underline{p}^{k+1} \cdot \underline{\underline{C}}^{-1}\underline{p}^{k+1}} \\
\underline{u}^{k+1} &= \underline{u}^k + \alpha^{k+1}\underline{p}^{k+1} \\
\underline{r}^{k+1} &= \underline{\underline{C}}^{-1}\underline{r}^k - \alpha^{k+1}\underline{\underline{A}}\underline{p}^{k+1}.
\end{aligned} \tag{28}$$

Algoritmo Computacional del Método Dado el sistema $\underline{\underline{A}}\underline{u} = \underline{b}$, con la matriz $\underline{\underline{A}}$ simétrica y definida positiva de dimensión $n \times n$. La entrada al método será una elección de \underline{u}^0 como condición inicial, $\varepsilon > 0$ como la tolerancia del método, N como el número máximo de iteraciones y la matriz de preconditionamiento $\underline{\underline{C}}^{-1}$ de dimensión $n \times n$, el algoritmo del método de gradiente conjugado preconditionado queda como:

$$\begin{aligned}
\underline{r} &= \underline{b} - \underline{\underline{A}}\underline{u} \\
\underline{w} &= \underline{\underline{C}}^{-1}\underline{r} \\
\underline{v} &= (\underline{\underline{C}}^{-1})^T \underline{w} \\
\alpha &= \sum_{j=1}^n w_j^2 \\
k &= 1
\end{aligned}$$

Mientras que $k \leq N$

Si $\|\underline{v}\|_\infty < \varepsilon$ Salir

$$\underline{x} = \underline{\underline{A}}\underline{v}$$

$$t = \frac{\alpha}{\sum_{j=1}^n v_j x_j}$$

$$\underline{u} = \underline{u} + t\underline{v}$$

$$\underline{r} = \underline{r} - t\underline{x}$$

$$\underline{w} = \underline{\underline{C}}^{-1}\underline{r}$$

$$\beta = \sum_{j=1}^n w_j^2$$

Si $\|\underline{r}\|_\infty < \varepsilon$ Salir

$$s = \frac{\beta}{\alpha}$$

$$\underline{v} = (\underline{\underline{C}}^{-1})^T \underline{w} + s\underline{v}$$

$$\alpha = \beta$$

$$k = k + 1$$

La salida del método será la solución aproximada $\underline{u} = (u_1, \dots, u_n)$ y el residual $\underline{r} = (r_1, \dots, r_n)$.

En el caso del método sin preconditionamiento, $\underline{\underline{C}}^{-1}$ es la matriz identidad, que para propósitos de optimización sólo es necesario hacer la asignación de vectores correspondiente en lugar del producto de la matriz por el vector. En el caso de que la matriz \underline{A} no sea simétrica, el método de gradiente conjugado puede extenderse para soportarlas, para más información sobre pruebas de convergencia, resultados numéricos entre los distintos métodos de solución del sistema algebraico $\underline{Au} = \underline{b}$ generada por la discretización de un problema elíptico y como extender estos para matrices no simétricas ver [9] y [7].

Clasificación de los Precondicionadores En general se pueden clasificar en dos grandes grupos según su manera de construcción: los algebraicos o a posteriori y los a priori o directamente relacionados con el problema continuo que lo origina.

2.2.1. Precondicionador a Posteriori

Los preconditionadores algebraicos o a posteriori son los más generales, ya que sólo dependen de la estructura algebraica de la matriz \underline{A} , esto quiere decir que no tienen en cuenta los detalles del proceso usado para construir el sistema de ecuaciones lineales $\underline{Au} = \underline{b}$. Entre estos podemos citar los métodos de preconditionamiento del tipo Jacobi, SSOR, factorización incompleta, inversa aproximada, diagonal óptimo y polinomial.

Precondicionador Jacobi El método preconditionador Jacobi es el preconditionador más simple que existe y consiste en tomar en calidad de preconditionador a los elementos de la diagonal de \underline{A}

$$C_{ij} = \begin{cases} A_{ij} & \text{si } i = j \\ 0 & \text{si } i \neq j. \end{cases} \quad (29)$$

Debido a que las operaciones de división son usualmente más costosas en tiempo de cómputo, en la práctica se almacenan los recíprocos de la diagonal de \underline{A} .

Ventajas: No necesita trabajo para su construcción y puede mejorar la convergencia.

Desventajas: En problemas con número de condicionamiento muy grande, no es notoria la mejora en el número de iteraciones.

Precondicionador SSOR Si la matriz original es simétrica, se puede descomponer como en el método de sobrerrelajamiento sucesivo simétrico (SSOR) de la siguiente manera

$$\underline{\underline{A}} = \underline{\underline{D}} + \underline{\underline{L}} + \underline{\underline{L}}^T \quad (30)$$

donde $\underline{\underline{D}}$ es la matriz de la diagonal principal y $\underline{\underline{L}}$ es la matriz triangular inferior.

La matriz en el método SSOR se define como

$$\underline{\underline{C}}(\omega) = \frac{1}{2-w} \left(\frac{1}{\omega} \underline{\underline{D}} + \underline{\underline{L}} \right) \left(\frac{1}{\omega} \underline{\underline{D}} \right)^{-1} \left(\frac{1}{\omega} \underline{\underline{D}} + \underline{\underline{L}} \right)^T \quad (31)$$

en la práctica la información espectral necesaria para hallar el valor óptimo de ω es demasiado costoso para ser calculado.

Ventajas: No necesita trabajo para su construcción, puede mejorar la convergencia significativamente.

Desventajas: Su paralelización depende fuertemente del ordenamiento de las variables.

Precondicionador de Factorización Incompleta Existen una amplia clase de preconditionadores basados en factorizaciones incompletas. La idea consiste en que durante el proceso de factorización se ignoran ciertos elementos diferentes de cero correspondientes a posiciones de la matriz original que son nulos. La matriz preconditionadora se expresa como $\underline{\underline{C}} = \underline{\underline{L}}\underline{\underline{U}}$, donde $\underline{\underline{L}}$ es la matriz triangular inferior y $\underline{\underline{U}}$ la superior. La eficacia del método depende de cuán buena sea la aproximación de $\underline{\underline{C}}^{-1}$ con respecto a $\underline{\underline{A}}^{-1}$.

El tipo más común de factorización incompleta se basa en seleccionar un subconjunto S de las posiciones de los elementos de la matriz y durante el proceso de factorización considerar a cualquier posición fuera de éste igual a cero. Usualmente se toma como S al conjunto de todas las posiciones (i, j) para las que $A_{ij} \neq 0$. Este tipo de factorización es conocido como factorización incompleta LU de nivel cero, ILU(0).

El proceso de factorización incompleta puede ser descrito formalmente como sigue:

Para cada k , si $i, j > k$:

$$S_{ij} = \begin{cases} A_{ij} - A_{ij}A_{ij}^{-1}A_{kj} & \text{Si } (i, j) \in S \\ A_{ij} & \text{Si } (i, j) \notin S. \end{cases} \quad (32)$$

Una variante de la idea básica de las factorizaciones incompletas lo constituye la factorización incompleta modificada que consiste en que si el producto

$$A_{ij} - A_{ij}A_{ij}^{-1}A_{kj} \neq 0 \quad (33)$$

y el llenado no está permitido en la posición (i, j) , en lugar de simplemente descartarlo, esta cantidad se le subtrae al elemento de la diagonal A_{ij} . Matemáticamente esto corresponde a forzar a la matriz preconditionadora a tener la misma suma por filas que la matriz original. Esta variante resulta de interés puesto

que se ha probado que para ciertos casos la aplicación de la factorización incompleta modificada combinada con pequeñas perturbaciones hace que el número de condicionamiento espectral del sistema preconditionado sea de un orden inferior.

Ventaja: Puede mejorar el condicionamiento y la convergencia significativamente.

Desventaja: El proceso de factorización es costoso y difícil de paralelizar en general.

Precondicionador de Inversa Aproximada El uso del preconditionador de inversas aproximada se han convertido en una buena alternativa para los preconditionadores implícitos debido a su naturaleza paralelizable. Aquí se construye una matriz inversa aproximada usando el producto escalar de Frobenius.

Sea $\mathcal{S} \subset C_n$, el subespacio de las matrices $\underline{\underline{C}}$ donde se busca una inversa aproximada explícita con un patrón de dispersión desconocido. La formulación del problema esta dada como: Encontrar $\underline{\underline{C}}_0 \in \mathcal{S}$ tal que

$$\underline{\underline{C}}_0 = \arg \min_{\underline{\underline{C}} \in \mathcal{S}} \|\underline{\underline{AC}} - \underline{\underline{I}}\|. \quad (34)$$

Además, esta matriz inicial $\underline{\underline{C}}_0$ puede ser una inversa aproximada de $\underline{\underline{A}}$ en un sentido estricto, es decir,

$$\|\underline{\underline{AC}}_0 - \underline{\underline{I}}\| = \varepsilon < 1. \quad (35)$$

Existen dos razones para esto, primero, la ecuación (35) permite asegurar que $\underline{\underline{C}}_0$ no es singular (lema de Banach), y segundo, esta será la base para construir un algoritmo explícito para mejorar $\underline{\underline{C}}_0$ y resolver la ecuación $\underline{\underline{Au}} = \underline{\underline{b}}$.

La construcción de $\underline{\underline{C}}_0$ se realiza en paralelo, independizando el cálculo de cada columna. El algoritmo permite comenzar desde cualquier entrada de la columna k , se acepta comúnmente el uso de la diagonal como primera aproximación. Sea r_k el residuo correspondiente a la columna k -ésima, es decir

$$r_k = \underline{\underline{AC}}_k - \underline{\underline{e}}_k \quad (36)$$

y sea \mathcal{I}_k el conjunto de índices de las entradas no nulas en r_k , es decir, $\mathcal{I}_k = \{i = \{1, 2, \dots, n\} \mid r_{ik} \neq 0\}$. Si $\mathcal{L}_k = \{l = \{1, 2, \dots, n\} \mid C_{lk} \neq 0\}$, entonces la nueva entrada se busca en el conjunto $\mathcal{J}_k = \{j \in \mathcal{L}_k^c \mid A_{ij} \neq 0, \forall i \in \mathcal{I}_k\}$. En realidad las únicas entradas consideradas en $\underline{\underline{C}}_k$ son aquellas que afectan las entradas no nulas de r_k . En lo que sigue, asumimos que $\mathcal{L}_k \cup \{j\} = \{i_1^k, i_2^k, \dots, i_{p_k}^k\}$ es no vacío, siendo p_k el número actual de entradas no nulas de $\underline{\underline{C}}_k$ y que $i_{p_k}^k = j$, para todo $j \in \mathcal{J}_k$. Para cada j , calculamos

$$\|\underline{\underline{AC}}_k - \underline{\underline{e}}_k\|_2^2 = 1 - \sum_{l=1}^{p_k} \frac{[\det(\underline{\underline{D}}_l^k)]^2}{\det(\underline{\underline{G}}_{l-2}^k) \det(\underline{\underline{G}}_l^k)} \quad (37)$$

donde, para todo k , $\det(\underline{\underline{G}}_0^k) = 1$ y $\underline{\underline{G}}_l^k$ es la matriz de Gram de las columnas $i_1^k, i_2^k, \dots, i_{p_k}^k$ de la matriz $\underline{\underline{A}}$ con respecto al producto escalar Euclideo; $\underline{\underline{D}}_l^k$ es la matriz que resulta de remplazar la última fila de la matriz $\underline{\underline{G}}_l^k$ por $a_{ki_1^k}, a_{ki_2^k}, \dots, a_{ki_l^k}$, con $1 \leq l \leq p_k$. Se selecciona el índice j_k que minimiza el valor de $\|\underline{\underline{A}}\underline{\underline{C}}_k - \underline{\underline{e}}_k\|_2$.

Esta estrategia define el nuevo índice seleccionado j_k atendiendo solamente al conjunto \mathcal{L}_k , lo que nos lleva a un nuevo óptimo donde se actualizan todas las entradas correspondientes a los índices de \mathcal{L}_k . Esto mejora el criterio de (34) donde el nuevo índice se selecciona manteniendo las entradas correspondientes a los índices de \mathcal{L}_k . Así $\underline{\underline{C}}_k$ se busca en el conjunto

$$\mathcal{S}_k = \{\underline{\underline{C}}_k \in \mathbb{R}^n \mid C_{ik} = 0, \forall i \in \mathcal{L}_k \cup \{j_k\}\},$$

$$\underline{\underline{m}}_k = \sum_{l=1}^{p_k} \frac{\det(\underline{\underline{D}}_l^k)}{\det(\underline{\underline{G}}_{l-2}^k) \det(\underline{\underline{G}}_l^k)} \tilde{m}_l \quad (38)$$

donde $\tilde{\underline{\underline{C}}}_l$ es el vector con entradas no nulas i_h^k ($1 \leq h \leq l$). Cada una de ellas se obtiene evaluado el determinante correspondiente que resulta de remplazar la última fila del $\det(\underline{\underline{G}}_l^k)$ por e_h^t , con $1 \leq l \leq p_k$.

Evidentemente, los cálculos de $\|\underline{\underline{A}}\underline{\underline{C}}_k - \underline{\underline{e}}_k\|_2^2$ y de $\underline{\underline{C}}_k$ pueden actualizarse añadiendo la contribución de la última entrada $j \in \mathcal{J}_k$ a la suma previa de 1 a $p_k - 1$. En la práctica, $\det(\underline{\underline{G}}_l^k)$ se calcula usando la descomposición de Cholesky puesto que $\underline{\underline{G}}_l^k$ es una matriz simétrica y definida positiva. Esto sólo involucra la factorización de la última fila y columna si aprovechamos la descomposición de $\underline{\underline{G}}_{l-1}^k$. Por otra parte, $\det(\underline{\underline{D}}_l^k) / \det(\underline{\underline{G}}_l^k)$ es el valor de la última incógnita del sistema $\underline{\underline{G}}_l^k \underline{\underline{d}}_l = \begin{pmatrix} a_{ki_1^k}, a_{ki_2^k}, \dots, a_{ki_l^k} \end{pmatrix}^T$ necesitándose solamente una sustitución por descenso. Finalmente, para obtener $\tilde{\underline{\underline{C}}}_l$ debe resolverse el sistema $\underline{\underline{G}}_l^k \underline{\underline{v}}_l = \underline{\underline{e}}_l$, con $\tilde{\underline{\underline{C}}}_{i_1^k l} = v_{hl}$, ($1 \leq h \leq l$).

Ventaja: Puede mejorar el condicionamiento y la convergencia significativamente y es fácilmente paralelizable.

Desventaja: El proceso construcción es algo laborioso.

2.2.2. Precondicionador a Priori

Los preconditionadores a priori son más particulares y dependen para su construcción del conocimiento del proceso de discretización de la ecuación diferencial parcial, dicho de otro modo dependen más del proceso de construcción de la matriz $\underline{\underline{A}}$ que de la estructura de la misma.

Estos preconditionadores usualmente requieren de más trabajo que los del tipo algebraico discutidos anteriormente, sin embargo permiten el desarrollo de métodos de solución especializados más rápidos que los primeros.

Veremos algunos de los métodos más usados relacionados con la solución de ecuaciones diferenciales parciales en general y luego nos concentraremos en el caso de los métodos relacionados directamente con descomposición de dominio.

En estos casos el preconditionador $\underline{\underline{C}}$ no necesariamente toma la forma simple de una matriz, sino que debe ser visto como un operador en general. De aquí que $\underline{\underline{C}}$ podría representar al operador correspondiente a una versión simplificada del problema con valores en la frontera que deseamos resolver.

Por ejemplo se podría emplear en calidad de preconditionador al operador original del problema con coeficientes variables tomado con coeficientes constantes. En el caso del operador de Laplace se podría tomar como preconditionador a su discretización en diferencias finitas centrales.

Por lo general estos métodos alcanzan una mayor eficiencia y una convergencia óptima, es decir, para ese problema en particular el preconditionador encontrado será el mejor preconditionador existente, llegando a disminuir el número de iteraciones hasta en un orden de magnitud. Donde muchos de ellos pueden ser paralelizados de forma efectiva.

El Uso de la Parte Simétrica como Preconditionador La aplicación del método del gradiente conjugado en sistemas no auto-adjuntos requiere del almacenamiento de los vectores previamente calculados. Si se usa como preconditionador la parte simétrica

$$(\underline{\underline{A}} + \underline{\underline{A}}^T)/2 \quad (39)$$

de la matriz de coeficientes $\underline{\underline{A}}$, entonces no se requiere de éste almacenamiento extra en algunos casos, resolver el sistema de la parte simétrica de la matriz $\underline{\underline{A}}$ puede resultar más complicado que resolver el sistema completo.

El Uso de Métodos Directos Rápidos como Precondicionadores En muchas aplicaciones la matriz de coeficientes $\underline{\underline{A}}$ es simétrica y positivo definida, debido a que proviene de un operador diferencial auto-adjunto y acotado. Esto implica que se cumple la siguiente relación para cualquier matriz $\underline{\underline{B}}$ obtenida de una ecuación diferencial similar

$$c_1 \leq \frac{x^T \underline{\underline{A}} x}{x^T \underline{\underline{B}} x} \leq c_2 \quad \forall x \quad (40)$$

donde c_1 y c_2 no dependen del tamaño de la matriz. La importancia de esta propiedad es que del uso de $\underline{\underline{B}}$ como preconditionador resulta un método iterativo cuyo número de iteraciones no depende del tamaño de la matriz.

La elección más común para construir el preconditionador $\underline{\underline{B}}$ es a partir de la ecuaciones diferenciales parciales separable. El sistema resultante con la matriz $\underline{\underline{B}}$ puede ser resuelto usando uno de los métodos directos de solución rápida, como pueden ser por ejemplo los basados en la transformada rápida de Fourier.

Como una ilustración simple del presente caso obtenemos que cualquier operador elíptico puede ser preconditionado con el operador de Poisson.

Construcción de Precondicionadores para Problemas Elípticos Empleando DDM Existen una amplia gama de este tipo de precondicionadores, pero son específicos al método de descomposición de dominio usado, para el método de subestructuración, los más importantes se derivan de la matriz de rigidez y por el método de proyecciones, el primero se detalla en la sección (??) y el segundo, conjuntamente con otros precondicionadores pueden ser consultados en [11], [5], [4] y [2].

La gran ventaja de este tipo de precondicionadores es que pueden ser óptimos, es decir, para ese problema en particular el precondicionador encontrado será el mejor precondicionador existente, llegando a disminuir el número de iteraciones hasta en un orden de magnitud.

3. Bibliografía

Referencias

- [1] A. Quarteroni, A. Valli; *Domain Decomposition Methods for Partial Differential Equations*. Clarendon Press Oxford 1999.
- [2] A. Toselli, O. Widlund; *Domain Decomposition Methods - Algorithms and Theory*. Springer, 2005.
- [3] B. D. Reddy; *Introductory Functional Analysis - With Applications to Boundary Value Problems and Finite Elements*. Springer 1991.
- [4] B. F. Smith, P. E. Bjørstad, W. D. Gropp; *Domain Decomposition, Parallel Multilevel Methods for Elliptic Partial Differential Equations*. Cambridge University Press, 1996.
- [5] B. I. Wohlmuth; *Discretization Methods and Iterative Solvers Based on Domain Decomposition*. Springer, 2003.
- [6] I. Foster; *Designing and Building Parallel Programs*. Addison-Wesley Inc., Argonne National Laboratory, and the NSF, 2004.
- [7] G. Herrera; Análisis de Alternativas al Método de Gradiente Conjugado para Matrices no Simétricas. Tesis de Licenciatura, Facultad de Ciencias, UNAM, 1989.
- [8] I. Herrera, M. Díaz; *Modelación Matemática de Sistemas Terrestres* (Notas de Curso en Preparación). Instituto de Geofísica, (UNAM).
- [9] I. Herrera; *Un Análisis del Método de Gradiente Conjugado*. Comunicaciones Técnicas del Instituto de Geofísica, UNAM; Serie Investigación, No. 7, 1988.
- [10] I. Herrera; *Método de Subestructuración* (Notas de Curso en Preparación). Instituto de Geofísica, (UNAM).
- [11] J. H. Bramble, J. E. Pasciak and A. H. Schatz. *The Construction of Preconditioners for Elliptic Problems by Substructuring*. I. Math. Comput., 47, 103-134, 1986.
- [12] J. L. Lions & E. Magenes; *Non-Homogeneous Boundary Value Problems and Applications Vol. I*, Springer-Verlag Berlin Heidelberg New York 1972.
- [13] K. Hutter & K. Jöhnk; *Continuum Methods of Physical Modeling*. Springer-Verlag Berlin Heidelberg New York 2004.
- [14] L. F. Pavarino, A. Toselli; *Recent Developments in Domain Decomposition Methods*. Springer, 2003.

- [15] M.B. Allen III, I. Herrera & G. F. Pinder; *Numerical Modeling in Science And Engineering*. John Wiley & Sons, Inc . 1988.
- [16] M. Diaz; *Desarrollo del Método de Colocación Trefftz-Herrera Aplicación a Problemas de Transporte en las Geociencias*. Tesis Doctoral, Instituto de Geofísica, UNAM, 2001.
- [17] M. Diaz, I. Herrera; *Desarrollo de Precondicionadores para los Procedimientos de Descomposición de Dominio*. Unidad Teórica C, Posgrado de Ciencias de la Tierra, 22 pags, 1997.
- [18] P.G. Ciarlet, J. L. Lions; *Handbook of Numerical Analysis, Vol. II*. North-Holland, 1991.
- [19] R. L. Burden y J. D. Faires; *Análisis Numérico*. Math Learning, 7 ed. 2004.
- [20] S. Friedberg, A. Insel, and L. Spence; *Linear Algebra*, 4th Edition, Prentice Hall, Inc. 2003.
- [21] W. Gropp, E. Lusk, A. Skjellein, *Using MPI, Portable Parallel Programming With the Message Passing Interface*. Scientific and Engineering Computation Series, 2ed, 1999.
- [22] W. Rudin; *Principles of Mathematical Analysis*. McGraw-Hill International Editions, 1976.
- [23] X. O. Olivella, C. A. de Sacibar; *Mecánica de Medios Continuos para Ingenieros*. Ediciones UPC, 2000.
- [24] Y. Saad; *Iterative Methods for Sparse Linear Systems*. SIAM, 2 ed. 2000.
- [25] Y. Skiba; *Métodos y Esquemas Numéricos, un Análisis Computacional*. UNAM, 2005.