



29
25

Universidad Nacional Autónoma de México

FACULTAD DE CIENCIAS

ANALISIS DE ALTERNATIVAS AL METODO DE
GRADIENTE CONJUGADO PARA
MATRICES NO SIMETRICAS

T E S I S

QUE PARA OBTENER EL TITULO DE

MATEMATICO

P R E S E N T A

GRACIELA DEL SOCORRO HERRERA ZAMARRON

MEXICO, D.

TESIS CON
FALLA LE ORIGEN

1989



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

Contenido

Introducción	1
Capítulo 1: Método tradicional de gradiente conjugado	2
1.1 Introducción	2
1.2 Conceptos fundamentales	3
1.3 Planteamiento general del problema	5
1.4 Minimización de la función cuadrática P en n -planos	11
1.5 Método de gradiente conjugado	14
Capítulo 2: Modificaciones para el caso no simétrico	18
2.1 Introducción	18
2.2 Extensión natural a matrices no simétricas	18
2.3 Método de residuo conjugado generalizado (GCR)	20
2.4 Métodos GCR(k) y ORTHOMIN(k)	22
Capítulo 3: Nuevo método para el caso no simétrico	23
3.1 Introducción	23
3.2 Planteamiento general de problema	23
3.3 Explicación del método	26
3.4 Construcción para la matriz no simétrica	32
Capítulo 4: Resultados numéricos y conclusiones	34
4.1 Introducción	31
4.2 Solución numérica de un problema específico	34
4.3 Resultados numéricos	37
4.4 Discusión y conclusiones	38
Apéndice: programas	58
Referencias	70

Introducción

El método de gradiente conjugado ha sido utilizado ampliamente en los últimos años. A pesar de que su formulación original fue hecha en 1952 por Hestenes y Stiefel [8], el interés actual se dió a partir de que Reid lo planteara como un método iterativo en 1971 [12], que es la forma en que se le utiliza con mayor frecuencia en la actualidad.

El método de gradiente conjugado (CG) se aplica para la ecuación $\mathbf{A}\mathbf{u} = \mathbf{b}$ cuando \mathbf{A} es una matriz positiva definida y simétrica. Sin embargo, se han formulado varios métodos basados en el de gradiente conjugado para eliminar esta restricción y extender el método a matrices no simétricas. De estos métodos, el primero en aparecer, que fuera formulado por los mismos Hestenes y Stiefel [8], se basa en la idea de resolver la ecuación $\mathbf{P}\mathbf{u} = \mathbf{a}$, donde \mathbf{P} es no simétrica y no singular, aplicando el método de gradiente conjugado a la ecuación equivalente $\mathbf{P}^T\mathbf{P}\mathbf{u} = \mathbf{P}^T\mathbf{a}$. Nos referiremos a este método como la extensión natural del gradiente conjugado. Entre los métodos más recientes, se ha trabajado ampliamente con ORTHOMIN(k) y GCR(k), formulados en 1976 por Vinsome [11].

En esta tesis se expone un nuevo método, propuesto por I. Herrera [10], basado en CG, que se puede aplicar al caso no simétrico. Los objetivos que nos proponemos son, por un lado, explicar de forma clara y detallada el método, y por otro hacer un análisis de la eficiencia de éste comparada con la de ORTHOMIN(k), GCR(k) y la extensión natural del gradiente conjugado en varios problemas particulares de transporte con difusión.

En el capítulo 1 se presentan algunos conceptos básicos del álgebra y el análisis que se utilizarán en el texto, además de una presentación del CG, siguiendo el texto de Hestenes [9]. El capítulo 2 presenta los métodos ORTHOMIN(k), GCR(k) y la extensión natural del CG. El capítulo 3 se inicia con una exposición del CG desde una óptica distinta a la planteada por Hestenes, siguiendo el mismo artículo de Herrera [10], con base en la cual se introduce el nuevo método. El capítulo 4 presenta los resultados numéricos y las conclusiones.

Capítulo 1

Método tradicional de gradiente conjugado

1.1 Introducción

En este capítulo revisaremos el método de gradiente conjugado, que en su forma original [8], se utiliza para resolver la ecuación

$$\mathbf{A}\mathbf{u} = \mathbf{b},$$

donde \mathbf{A} es una matriz $N \times N$ dimensional positiva definida y simétrica. Expondremos el método siguiendo a Hestenes [9] (otras presentaciones del método se pueden encontrar en [1], [2], [4] o [6]), el cual lo desarrolla al minimizar la función cuadrática

$$F(\mathbf{u}) = \frac{1}{2} \mathbf{u}^* \mathbf{A} \mathbf{u} - \mathbf{b}^* \mathbf{u} + c$$

positiva definida. La idea básica consiste en construir aproximaciones al vector solución \mathbf{u}_* , tomándolas como el mínimo de la función F sobre ciertos hiperplanos anidados de \mathbb{R}^N ,

$$\mathbb{H}_1 \subset \mathbb{H}_2 \subset \dots \subset \mathbb{H}_N,$$

donde el subíndice corresponde a la dimensión.

Para expresar las aproximaciones es necesario construir una base \mathbf{A} -ortogonal de cada hiperplano. El método que se utiliza usualmente para ortogonalizar vectores es el de Gram-Schmidt. Sin embargo, al seleccionar un nuevo elemento de la base, se requiere garantizar la ortogonalidad con respecto a todos los construidos anteriormente. En contraste con esto, los hiperplanos seleccionados en el método de gradiente conjugado, permiten asegurar que al escoger un nuevo elemento de la base baste con verificar la ortogonalidad con respecto al último que se ha construido para que, automáticamente, esta condición se cumpla con respecto a todos los anteriores. Esta forma de proceder permite construir la base iterativamente, de tal forma que cada nueva aproximación dependa únicamente de la precedente. En términos computacionales esto es una gran ventaja.

1.2 Conceptos Fundamentales

En esta sección introduciremos notación y terminología que se utilizará en el transcurso del texto, además revisaremos conceptos básicos sobre los que se desarrollará la teoría ulterior [5], [11].

Trabajaremos con funciones vectoriales de valor real en \mathbb{R}^N . Las componentes de los vectores se designarán por medio de subíndices cuando sea necesario, de tal forma que u_i es la i -ésima componente de \mathbf{u} . Cuando se hagan operaciones matriciales el vector \mathbf{u} se considerará como un vector columna. Denotaremos (\mathbf{u}, \mathbf{v}) a un producto escalar cualquiera y

$$\mathbf{u} + \mathbf{v} = \sum_{i=1}^N u_i + v_i$$

al producto interior asociado a la norma euclidiana de \mathbf{u} y \mathbf{v} en \mathbb{R}^N . La norma del vector \mathbf{u} respecto a un producto escalar cualquiera es $\|\mathbf{u}\| = (\mathbf{u}, \mathbf{u})^{1/2}$. Específicamente, denotaremos $\|\mathbf{u}\|_2$ a la norma euclidiana del vector \mathbf{u} y será

$$\|\mathbf{u}\|_2 = \left(\sum_{i=1}^N u_i^2 \right)^{1/2}.$$

La distancia entre dos vectores \mathbf{u} y \mathbf{v} es $\|\mathbf{u} - \mathbf{v}\|$. Una vecindad de \mathbf{u}^1 de radio δ es el conjunto de todos los puntos \mathbf{u} tales que $\|\mathbf{u} - \mathbf{u}^1\| < \delta$. Un conjunto S es abierto si todo punto \mathbf{u}^1 de S tiene una vecindad, contenida en éste, en la que está \mathbf{u}^1 .

Es necesario introducir también algunas relaciones entre matrices y funciones cuadráticas. $\mathbf{A}^T = (a_{ji})$ será la traspuesta de la matriz $\mathbf{A} = (a_{ij})$. Una matriz cuadrada es no singular si su determinante es distinto de cero. En este caso la inversa de \mathbf{A} existe y se denota \mathbf{A}^{-1} . Dado un producto interior (\cdot, \cdot) , la adjunta \mathbf{A}^* de la matriz \mathbf{A} es aquella que satisface $(\mathbf{u}, \mathbf{A}^* \mathbf{v}) = (\mathbf{v}, \mathbf{A} \mathbf{u})$. Utilizando esta noción decimos que una matriz \mathbf{A} es autoadjunta si $\mathbf{A}^* = \mathbf{A}$. Si $\mathbf{A}^T = \mathbf{A}$ decimos que \mathbf{A} es simétrica. Una matriz autoadjunta es no negativa si la desigualdad $(\mathbf{u}, \mathbf{A} \mathbf{u}) \geq 0$ se cumple para todo vector \mathbf{u} y es positiva definida si $(\mathbf{u}, \mathbf{A} \mathbf{u}) > 0$ cuando $\mathbf{u} \neq 0$. Si una matriz \mathbf{A} es positiva definida, \mathbf{A}^{-1} también lo es. La parte simétrica de la matriz \mathbf{A} está dada por $M = \frac{1}{2}(\mathbf{A} + \mathbf{A}^T)$. Trabajaremos con funciones definidas por medio del producto interno \cdot . La igualdad $\mathbf{A} \mathbf{u} + \mathbf{v} = \mathbf{u} + \mathbf{A}^T \mathbf{v}$ siempre se cumple y si \mathbf{A} es simétrica $\mathbf{A} \mathbf{u} + \mathbf{v} = \mathbf{u} + \mathbf{A} \mathbf{v}$. La función $Q(\mathbf{u}) = \mathbf{u} + \mathbf{A} \mathbf{u}$ es llamada forma cuadrática y es positiva definida si \mathbf{A} lo es. Una función F expresable en la forma

$$F(\mathbf{u}) = \frac{1}{2} \mathbf{u} + \mathbf{A} \mathbf{u} - \mathbf{b} + \mathbf{v} + c$$

donde \mathbf{A} es simétrica es llamada función cuadrática.

Utilizaremos algunos conceptos del cálculo para minimizar a la función cuadrá-

tica F . Consideremos una función continua f de valor real en un conjunto abierto S que tenga derivadas parciales continuas de primer y segundo orden en S . Una función tal se dice que es de clase $C^{(2)}$. La primera diferencial $f'(u, v)$ y segunda diferencial $f''(u, v)$ de f en u están dadas por las fórmulas

$$f'(u, v) = \sum_{i=1}^N \frac{\partial f(u)}{\partial u_i} v_i = f'(u) \cdot v$$

y

$$f''(u, v) = \sum_{i=1}^N \sum_{j=1}^N \frac{\partial^2 f(u)}{\partial u_i \partial u_j} v_i v_j$$

respectivamente. El vector

$$f'(u) = \left(\frac{\partial f(u)}{\partial u_i} \right) \quad (i = 1, \dots, N)$$

es el gradiente de f en u y se denotará $\nabla f(u)$. La matriz

$$f''(u) = \left(\frac{\partial^2 f(u)}{\partial u_i \partial u_j} \right) \quad (i, j = 1, \dots, N)$$

es el hessiano de f en u .

Un punto u^0 proporciona un mínimo local de f en S si u^0 está en S y la desigualdad $f(u) \geq f(u^0)$ se cumple para toda u en S que esté en una vecindad V de u^0 . Si $f(u) \geq f(u^0)$ para toda u en S , entonces u^0 minimiza a f globalmente en S . En cada caso el mínimo es estricto si la igualdad es válida sólo si $u = u^0$. Si f es dos veces diferenciable en un conjunto abierto S , entonces las relaciones

$$f'(u^0) = 0, \quad f''(u^0, v) \geq 0 \quad \forall v \neq 0$$

se cumplen en un punto mínimo local u^0 de f . Inversamente, si

$$f'(u^0) = 0, \quad f''(u^0, v) > 0 \quad \forall v \neq 0,$$

entonces u^0 proporciona un mínimo local estricto de f . Un punto u^0 en donde $f'(u^0) = 0$ es un punto crítico de f y $f(u^0)$ es el correspondiente valor crítico de f .

1.3 Planteamiento General del Problema

Supongamos que queremos minimizar una función $f : \mathbb{R}^N \rightarrow \mathbb{R}$. Una forma de hacerlo es minimizarla en hiperplanos anidados

$$\Pi_1 \subset \Pi_2 \subset \dots \subset \Pi_N,$$

donde el subíndice indica la dimensión. Llamaremos a Π_n n -plano. Si en cada uno de estos hiperplanos el mínimo u^i en Π_i es único, tendremos una sucesión de aproximaciones u^1, \dots, u^N tales que

$$f(u^1) \geq f(u^2) \geq \dots \geq f(u^N)$$

de tal forma que, a lo más, en N pasos encontramos el punto u_* = u^N que proporciona un mínimo estricto de f en \mathbb{R}^N , ya que, $\Pi_N = \mathbb{R}^N$.

En este caso queremos minimizar la función cuadrática

$$F(u) = \frac{1}{2}u^T A u - b^T u + c$$

en \mathbb{R}^N , donde A es una matriz simétrica positiva definida ($N \times N$)-dimensional, b es un vector fijo N -dimensional y c es un real. El valor de la constante c no jugará un papel importante en los algoritmos, por lo que no aparecerá más.

Para minimizar a la función F debemos encontrar los puntos en los que el gradiente de F se anula, es decir,

$$\nabla F(u) \equiv F'(u) = A u - b = 0.$$

Entonces u es un punto crítico de F si y solo si es solución del sistema lineal de ecuaciones

$$A u = b. \tag{1.3.1}$$

Como A no es singular (por ser positiva definida), existe exactamente una solución a esta ecuación, que es

$$u_* = A^{-1}b,$$

y por lo tanto un solo punto crítico de F . El hessiano es $F''(u) = A$, por lo que

$$F''(u_*, v) = v^T A v > 0 \quad \forall v \neq 0$$

y u_* es un mínimo estricto de F .

En el estudio de sistemas de ecuaciones lineales se acostumbra definir el residuo del sistema (1.3.1) en \mathbf{u} como el vector

$$\mathbf{r} = \mathbf{b} - \mathbf{A}\mathbf{u},$$

o bien,

$$\mathbf{r} = \mathbf{A}(\mathbf{u}_s - \mathbf{u}).$$

En nuestro caso, el vector \mathbf{r} es el gradiente negativo

$$\mathbf{r} = -F'(\mathbf{u})$$

de la función cuadrática F .

Hablaremos de puntos y vectores en un hiperplano. Definimos al hiperplano Π_n como el conjunto de puntos \mathbf{u} que satisfacen las siguientes ecuaciones

$$\mathbf{q}^i \cdot \mathbf{u} = \rho_i \quad (i = 1, \dots, N-n, 1 \leq n \leq N).$$

donde los vectores $\mathbf{q}^1, \dots, \mathbf{q}^{N-n}$ son linealmente independientes y $\rho_1, \dots, \rho_{N-n}$ son escalares dados. Si \mathbf{u}^0 es un punto en Π_n estas relaciones se pueden escribir como

$$\mathbf{q}^i \cdot (\mathbf{u} - \mathbf{u}^0) = 0 \quad (i = 1, \dots, N-n)$$

que explícitamente establece que Π_n pasa por \mathbf{u}^0 . Diremos que un vector \mathbf{w} está en Π_n si y solo si

$$\mathbf{q}^i \cdot \mathbf{w} = 0, \quad (i = 1, \dots, N-n). \quad (1.3.2)$$

La ecuación (1.3.2) tiene n soluciones linealmente independientes. Por lo que todo vector \mathbf{u} en Π_n se puede expresar como una combinación lineal

$$\mathbf{u} = \alpha_1 \mathbf{u}^1 + \dots + \alpha_n \mathbf{u}^n$$

de $\mathbf{u}^1, \dots, \mathbf{u}^n$, si estos son vectores linealmente independientes en Π_n . Sean \mathbf{u}^0 y $\tilde{\mathbf{u}}$ dos puntos en Π_n . El vector $\mathbf{u} = \tilde{\mathbf{u}} - \mathbf{u}^0$ está en Π_n y $\tilde{\mathbf{u}} = \mathbf{u} + \mathbf{u}^0$ se puede expresar en la forma

$$\tilde{\mathbf{u}} = \mathbf{u}^0 + \alpha_1 \mathbf{u}^1 + \dots + \alpha_n \mathbf{u}^n. \quad (1.3.3)$$

para alguna elección de los parámetros $\alpha_1, \dots, \alpha_n$. La ecuación (1.3.3) con $\alpha_1, \dots, \alpha_n$ parámetros arbitrarios se llama representación paramétrica de Π_n .

Conviene ahora dar una caracterización de los puntos mínimos de F sobre un n -plano Π_n , que sirva como guía para encontrarlos. Sea \mathbf{U} la matriz con columnas

u^1, u^2, \dots, u^n . Entonces (1.3.3) se puede escribir como

$$u = u^0 + Uv,$$

donde v es el vector cuyas componentes son $\alpha_1, \dots, \alpha_n$.

En el n -plano Π_n la función F es una función cuadrática, ya que si G es la función F restringida al n -plano Π_n tenemos

$$\begin{aligned} G(v) = F(u^0 + Uv) &= \frac{1}{2}(u^0 + Uv) * A(u^0 + Uv) - b * (u^0 + Uv) \\ &= \frac{1}{2}u^0 A u^0 - b * u^0 + \frac{1}{2}v * U^T A U v + (A u^0 - b) * U v \\ &= \frac{1}{2}v * B v - g * v + F(u^0) \end{aligned}$$

donde

$$\begin{aligned} B &= U^T A U, \\ g &= -U^T F'(u^0). \end{aligned}$$

Debido a que A es positiva definida y simétrica también lo es B . El punto

$$\tilde{v} = B^{-1}g$$

es solución de la ecuación $Bv - g = 0$ y, por lo tanto, es un mínimo estricto de G . Entonces el punto

$$\tilde{u} = u^0 + U\tilde{v}$$

es un punto mínimo estricto de F en Π_n . Además, de la relación

$$0 = G'(\tilde{v}) = U^T F'(u^0 + U\tilde{v}) = U^T F'(\tilde{u}),$$

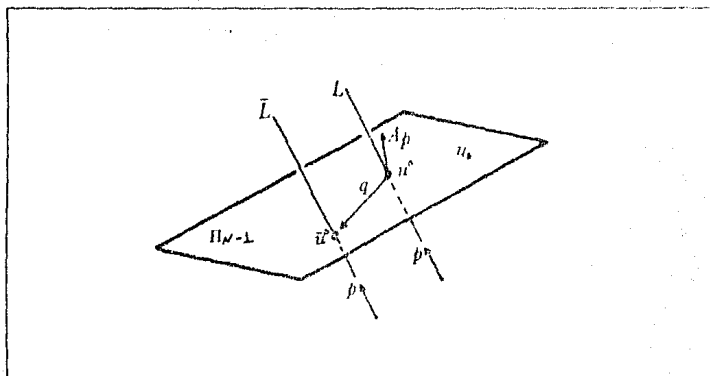
obtenemos que, el punto mínimo de F en Π_n se caracteriza por las ecuaciones

$$u^i * F'(\tilde{u}) = 0, \quad (i = 1, \dots, n),$$

o por la condición de que $F'(\tilde{u})$ sea ortogonal a Π_n .

Los resultados descritos se resumen en la siguiente Proposición.

PROPOSICIÓN 1.3.1 *Supongamos que el hessiano A de una de una función cuadrática F es positivo definido. Un punto \tilde{u} en un n -plano minimiza a F en Π_n si el gradiente $F'(\tilde{u})$ es ortogonal a Π_n , además existe un único punto mínimo de F en Π_n .*


 FIGURA 1. Puntos mínimos de F sobre dos líneas paralelas L y \bar{L} .

Ahora veamos cómo desarrollar un procedimiento para encontrar los mínimos de F en $\Pi_1, \Pi_2, \dots, \Pi_N$, hiperplanos dados.

PROPOSICIÓN 1.3.2 *Los puntos mínimos de F en líneas paralelas están sobre un hiperplano $\hat{\Pi}_{N-1}$ que pasa por el punto mínimo u_* de F . El $(N-1)$ -plano $\hat{\Pi}_{N-1}$ se define por medio de la ecuación*

$$\mathbf{p} \cdot (\mathbf{A}\mathbf{u} - \mathbf{b}) = 0, \quad (1.3.4)$$

donde \mathbf{p} es un vector dirección para estas líneas paralelas. El vector $\mathbf{A}\mathbf{p}$ es normal a $\hat{\Pi}_{N-1}$.

Demostración:

Sean \mathbf{u}^0 y $\bar{\mathbf{u}}^0$ puntos mínimos de F sobre dos líneas paralelas L y \bar{L} respectivamente, como se muestra en la Figura 1. La dirección de estas líneas se puede representar por medio de un vector \mathbf{p} no nulo. En el punto mínimo de F en L , el gradiente $F'(\mathbf{u}^0) = \mathbf{A}\mathbf{u}^0 - \mathbf{b}$ es ortogonal a L y por lo tanto a \mathbf{p} . El punto \mathbf{u}^0 satisface, entonces, la ecuación (1.3.4). De igual forma, $\bar{\mathbf{u}}^0$ satisface la ecuación, la cual representa un $(N-1)$ -plano $\hat{\Pi}_{N-1}$, cuya normal es $\mathbf{A}\mathbf{p}$, ya que

$$0 = \mathbf{p} \cdot (\mathbf{A}\mathbf{u} - \mathbf{b}) = \mathbf{p} \cdot \mathbf{A}(\mathbf{u} - \mathbf{u}_*) = \mathbf{A}\mathbf{p} \cdot (\mathbf{u} - \mathbf{u}_*),$$

además la ecuación indica explícitamente que \mathbf{u}_* está en el plano $\hat{\Pi}$.

Como se demostró antes, $\mathbf{A}\mathbf{p}$ es ortogonal a $\hat{\Pi}_{N-1}$; cuando esto sucede se dice que \mathbf{p} es conjugado (\mathbf{A} -ortogonal) a $\hat{\Pi}_{N-1}$ y que $\hat{\Pi}_{N-1}$ es conjugado a \mathbf{p} .

La representación paramétrica de una línea L es de la forma $\mathbf{u} = \mathbf{u}^0 + \alpha\mathbf{p}$, donde \mathbf{u}^0 es un punto en L y α es un parámetro que toma valores de $-\infty$ a ∞ . El residuo

de u^0 es

$$r^0 = -F'(u^0) = b - Au^0,$$

entonces

$$\begin{aligned} F(u^0 + \alpha p) &= \frac{1}{2}(u^0 + \alpha p)^T A(u^0 + \alpha p) - b^T (u^0 + \alpha p) \\ &= F(u^0) + \alpha p^T (Au^0 - b) + \frac{1}{2}\alpha^2 p^T A p \\ &= F(u^0) - \alpha p^T r^0 + \frac{1}{2}\alpha^2 p^T A p. \end{aligned} \quad (1.3.5)$$

La función (1.3.5) de α tiene un valor mínimo cuando $\alpha = p^T r^0 / p^T A p$. Tenemos el siguiente resultado.

PROPOSICIÓN 1.3.3 *El punto mínimo u^1 de F en la línea $u = u^0 + \alpha p$ está dado por la ecuación*

$$u^1 = u^0 + \alpha_1 p,$$

donde

$$\alpha_1 = \frac{c}{d}, \quad c = p^T r^0, \quad d = p^T A p, \quad r^0 = -F'(u^0) = b - Au^0.$$

Una Proposición relacionada con 1.3.2 es

PROPOSICIÓN 1.3.4 *Los puntos mínimos de F en $(N-1)$ -planos Π_{N-1} paralelos están en una línea L , conjugada a estos hiperplanos y que pasa por el punto mínimo u^* de F .*

Demostración:

Por la PROPOSICIÓN 1.3.1 el gradiente $F'(u^0) = Au^0 - b$ del punto mínimo u^0 de F en Π_{N-1} es ortogonal a este hiperplano y por lo tanto debe ser múltiplo de la normal q de Π_{N-1} . De acuerdo con esto existe un número α tal que

$$Au^0 - b = \alpha q,$$

o bien,

$$u^0 = A^{-1}b + \alpha A^{-1}q = u_* + \alpha A^{-1}q. \quad (1.3.6)$$

La ecuación (1.3.6) representa paramétricamente a una línea que pasa por el punto mínimo u_* de F en \mathbb{R}^N . Además el vector $p = A^{-1}q$ tiene la propiedad de ser conjugado a Π_{N-1} por ser q la normal de este hiperplano \mathfrak{H} .

Como una extensión de la PROPOSICIÓN 1.3.4 tenemos

PROPOSICIÓN 1.3.5 *Los puntos mínimos de F en n -planos paralelos están en un $(N - n)$ -plano conjugado a estos n -planos que pasa por el punto mínimo u_s de F . En otras palabras dado un conjunto de $N - n$ vectores linealmente independientes q^1, \dots, q^{N-n} , para cualquier conjunto de números reales $\rho_1, \dots, \rho_{N-n}$, el punto mínimo u^0 de F en el n -plano*

$$H_n : q^i \cdot u = \rho_i \quad (i = 1, \dots, N - n)$$

está sobre el $(N - n)$ -plano

$$\hat{H}_{N-n} : u = u_s + \alpha_1 A^{-1} q^1 + \dots + \alpha_{N-n} A^{-1} q^{N-n}$$

que pasa por el punto mínimo u_s de F . \hat{H}_{N-n} es conjugado a H_n .

La demostración es muy parecida a la de la PROPOSICIÓN 1.3.3 por lo que no la desarrollaremos.

PROPOSICIÓN 1.3.6 *Dado un n -plano H_n y un vector w , el conjunto $H_n + w$ de puntos u expresables como $u = u + w$ con u en H_n es un n -plano \hat{H}_n paralelo a H_n . Inversamente, si \hat{H}_n es un n -plano paralelo a H_n y $w = \hat{u} - u$ es un vector que une un punto u en H_n a un punto \hat{u} en \hat{H}_n , entonces $H_n = H_n + w$ por lo que existe un vector único p conjugado a H_n tal que $H_n = H_n + p$. Si u^0 minimiza a F en H_n , entonces $u^0 = u^0 + p$ minimiza a F en \hat{H}_n .*

Demostración:

Sea u^0 un punto en H_n y u^1, \dots, u^n vectores linealmente independientes en H_n . Entonces H_n se puede representar paramétricamente como

$$u = u^0 + \alpha_1 u^1 + \dots + \alpha_n u^n.$$

Si \hat{u}^0 es un punto en un segundo n -plano \hat{H}_n , entonces este hiperplano es paralelo a H_n si y solo si se puede representar paramétricamente por medio de la ecuación

$$\hat{u} = \hat{u}^0 + \alpha_1 u^1 + \dots + \alpha_n u^n.$$

El conjunto $H_n + w$ será

$$\begin{aligned} \hat{u} &= u^0 + \alpha_1 u^1 + \dots + \alpha_n u^n + w \\ &= \hat{u}^0 + \alpha_1 u^1 + \dots + \alpha_n u^n, \end{aligned}$$

donde $\hat{u}^0 = u^0 + w$ por lo que \hat{H}_n es paralelo a H_n . Ahora si \hat{H}_n es paralelo a H_n y $w = \hat{u}^0 - u^0$ donde \hat{u}^0 es un punto en \hat{H}_n y u^0 en H_n , entonces $\hat{u}^0 = w + u^0$ y un

punto \tilde{u} en $\tilde{\Pi}_n$ se puede escribir como

$$\begin{aligned}\tilde{u} &= u^0 + \alpha_1 u^1 + \dots + \alpha_n u^n \\ &= w + u^0 + \alpha_1 u^1 + \dots + \alpha_n u^n \\ &= w + u,\end{aligned}$$

donde u es un punto de Π_n . Si u^0 y u^0 son, respectivamente, los puntos mínimos de Π_n y $\tilde{\Pi}_n$, entonces el vector $p = \tilde{u}^0 - u^0$ está en el \mathbb{R}^{N-n} plano conjugado a Π_n y a $\tilde{\Pi}_n$ (PROPOSICIÓN 1.3.5) y es conjugado a estos n -planos. Además $\Pi_n = \Pi_n + p$. Inversamente, supongamos que p es conjugado a Π_n y que $\Pi_n = \Pi_n + p$. Si u^0 minimiza a F en Π_n , entonces $F'(u^0)$ y $\mathbb{R}p$ son ambos ortogonales a Π_n , de tal forma que $F'(u^0 + p) = F'(u^0) + \mathbb{R}p$ es ortogonal a Π_n y entonces al n plano paralelo $\Pi_n = \Pi_n + p$. Se sigue que el punto $u^0 + p$ es el punto mínimo u^0 de F en Π_n . Esto establece la última parte de la Proposición 1.3.6.

PROPOSICIÓN 1.3.7 Dado un n -plano Π_n y un vector w fuera de éste, el conjunto $\Pi_n + \beta w$ de todos los puntos $u + \beta w$, determinados por los puntos u en Π_n y los números reales β , es un $(n+1)$ -plano Π_{n+1} que expande a Π_n y a $\Pi_n = \Pi_n + w$. Existe un vector único p conjugado a Π_n tal que $\Pi_n = \Pi_n + p$ y $\Pi_{n+1} = \Pi_n + \beta p$. El punto mínimo u^1 de F en Π_{n+1} está en la línea $u = u^0 + \alpha p$, donde u^0 es el punto mínimo de F en Π_n .

Demostración:

Si u^0 es el punto mínimo de F en Π_n y u^1, \dots, u^n son vectores linealmente independientes en Π_n , entonces Π_n se puede representar paramétricamente por medio de la ecuación $u = u^0 + \alpha_1 u^1 + \dots + \alpha_n u^n$. Si w no está en Π_n , los vectores u^1, \dots, u^n , y son linealmente independientes y la ecuación $v = u^0 + \alpha_1 u^1 + \dots + \alpha_n u^n + \beta w$ con parámetros arbitrarios $\alpha_1, \dots, \alpha_n, \beta$ definen un $(n+1)$ -plano Π_{n+1} que cumple las condiciones descritas en la Proposición. Por la PROPOSICIÓN 1.3.6 existe un vector único p conjugado a Π_n tal que $\Pi_n = \Pi_n + p$. En virtud de la PROPOSICIÓN 1.3.4 con Π_{n+1} jugando el papel \mathbb{R}^N , el punto mínimo u^1 de F en Π_{n+1} está en la línea $u = u^0 + \alpha p$ que pasa por el punto mínimo de Π_n y es conjugado al mismo \mathbb{R} .

1.4 Minimización de la Función Cuadrática F en n -Planos

Sea $\{p^1, \dots, p^n\}$ un conjunto de vectores no nulos mutuamente conjugados, es decir, se satisfacen las relaciones

$$p^i \cdot \mathbb{R}p^j = 0 \quad (i \neq j, i = 1, \dots, n). \quad (1.4.1)$$

PROPOSICIÓN 1.4.1 Sea

$$\Pi_n : u = u^0 + \alpha_1 p^1 + \dots + \alpha_n p^n$$

el n -plano que contiene al punto u^0 y está determinado por un sistema conjugado p^1, \dots, p^n . El punto mínimo u^n de F en Π_n está dado por la fórmula

$$u^n = u^0 + \alpha_1 p^1 + \dots + \alpha_n p^n, \quad (1.4.2a)$$

donde

$$\alpha_i = \frac{c_i}{d_i}, \quad c_i = p^i * r^0, \quad d_i = p^i * A p^i, \quad (i = 1, \dots, n) \quad (1.4.2b)$$

y $r^0 = -F'(u^0) = b - Au^0$ es el residuo de F en u^0 . El residuo de $r^n = -F'(u^n)$ de F en u^n está determinado por

$$r^n = r^0 - \alpha_1 A p^1 - \dots - \alpha_n A p^n, \quad (1.4.3)$$

y es ortogonal a Π_n , de tal forma que

$$p^i * r^n = 0 \quad (i = 1, \dots, n). \quad (1.4.4)$$

El valor mínimo de F en Π_n es

$$F(u^n) = F(u^0) - \frac{1}{2}(\alpha_1 c_1 + \dots + \alpha_n c_n). \quad (1.4.5)$$

Demostración:

Por la PROPOSICIÓN 1.3.1 el punto mínimo u^n de F en Π_n se caracteriza por el hecho de que su residuo $r^n = -F'(u^n)$ es ortogonal a Π_n y por lo tanto a los vectores p^1, \dots, p^n en Π_n . Como

$$r^n = b - Au^n = b - A(u^0 + \alpha_1 p^1 + \dots + \alpha_n p^n),$$

la fórmula (1.4.3) se satisface. Combinando (1.4.3) y (1.4.4) con las relaciones (1.4.1) encontramos que para $i = 1, \dots, n$

$$0 = p^i * r^n = p^i * r^0 - \sum_{j=1}^n \alpha_j p^i * A p^j = c_i - \alpha_i d_i$$

por lo que

$$\alpha_i = \frac{c_i}{d_i}.$$

Para obtener (1.4.5) usamos la identidad

$$F(u + p) - F(u) = p * F'(u) + \frac{1}{2} p * A p$$

con $u = u^n$ y

$$p = u^0 - u^n = -\alpha_1 p^1 - \dots - \alpha_{n+1} p^n.$$

Ya que p es un vector sobre Π_n , es ortogonal a $F'(u^n) = -r^n$, de tal forma que

$$\begin{aligned} F(u^0) - F(u^n) &= p \cdot F'(u^n) + \frac{1}{2} p \cdot A p \\ &= \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j p^i \cdot A p^j \\ &= \frac{1}{2} \sum_{i=1}^n \alpha_i^2 p^i \cdot A p^i = \frac{1}{2} \sum_{i=1}^n \alpha_i^2 d_i = \frac{1}{2} \sum_{i=1}^n \alpha_i c_i, \end{aligned}$$

por las relaciones de conjugación (1.4.1) y (1.4.2b) \square

PROPOSICIÓN 1.4.2 Sea $\{p^1, \dots, p^m\}$ un sistema conjugado de vectores. Para un punto dado u^0 , sean u^1, \dots, u^m los puntos definidos recursivamente por el hecho de que para cada $n+1$, $1 \leq n+1 \leq m$, el punto u^{n+1} minimice a F sobre la línea

$$u = u^n + \alpha p^n. \quad (1.4.6)$$

Entonces u^n y el residuo $r^n = -F'(u^n)$ son

$$\begin{aligned} u^n &= u^{n-1} + \alpha_n p^n, \\ r^n &= r^{n-1} - \alpha_n A p^n, \end{aligned} \quad (1.4.7a)$$

donde

$$\alpha_n = \frac{c_n}{d_n}, \quad c_n = p^n \cdot r^{n-1}, \quad d_n = p^n \cdot A p^n. \quad (1.4.7b)$$

El punto u^n minimiza a F en el n -plano

$$\Pi_n : u = u^0 + \alpha_1 p^1 + \dots + \alpha_n p^n.$$

Tenemos las relaciones

$$\begin{aligned} p^n \cdot r^j &= c_n \quad (j < n \leq m), \\ p^n \cdot r^j &= 0, \quad (n+1 \leq j \leq m). \end{aligned} \quad (1.4.8)$$

Demostración:

Como u^n minimiza a F en el 1-plano (1.1.6), se sigue de la PROPOSICIÓN 1.4.1, con u^{n-1} tomando el papel de u^n , que u^n y r^n se pueden escribir como en (1.4.7) y que

$$p^n \star r^n = 0, \quad (n = 1, \dots, m).$$

Por las relaciones de conjugación $p^n \star Ap^j = 0, (n \neq j)$ tenemos

$$p^n \star r^{j+1} = p^n \star (r^j - \alpha_j Ap^j) = p^n \star r^j, \quad (j \neq n, j = 1, \dots, m).$$

En consecuencia, usando $j = 0, \dots, n-2$

$$p^n \star r^0 = p^n \star r^1 = \dots = p^n \star r^{n-1} = c_n \quad (n \leq m),$$

tomando $j = n, \dots, m-1$

$$p^n \star r^m = p^n \star r^{m-1} = \dots = p^n \star r^n = 0 \quad (n \leq m).$$

Esto demuestra la relación (1.4.8). Como $c_j = p^j \star r^0$, se sigue de la PROPOSICIÓN 1.4.1 que u^n minimiza a F en el n -plano \mathfrak{B} .

1.5 Método de Gradiente Conjugado

Como consecuencia de los resultados anteriores, para minimizar la función cuadrática F se puede utilizar una rutina iterativa en la que habiendo encontrado un punto u^{n-1} que minimice a la función F sobre Π_{n-1} y una dirección p^n conjugada a Π_{n-1} , encontramos el punto mínimo u^n en Π_n minimizando la función sobre la línea $u = u^{n-1} + \alpha p^n$.

Este método, llamado de direcciones conjugadas, se puede formalizar como sigue.

Algoritmo CG

PASO INICIAL: Seleccionar una estimación inicial u^0 del punto mínimo u_s . Calcular el residuo

$$r^0 = b - Au^0$$

y seleccionar una dirección inicial $p^1 \neq 0$.

PASOS ITERATIVOS: Habiendo obtenido la estimación u^{n-1} de u_s , su residuo r^{n-1} y la dirección p^n , calcular una nueva aproximación u^n y su residuo r^n por medio

de las fórmulas

$$\alpha_n = \frac{\mathbf{p}^n \cdot \mathbf{r}^{n-1}}{\mathbf{p}^n \cdot \mathbf{A}\mathbf{p}^n}, \quad (1.5.1c)$$

$$\mathbf{u}^n = \mathbf{u}^{n-1} + \alpha_n \mathbf{p}^n,$$

$$\mathbf{r}^n = \mathbf{r}^{n-1} + \alpha_n,$$

$$\mathbf{A}\mathbf{p}^n = \mathbf{b} - \mathbf{A}\mathbf{u}^n. \quad (1.5.1b)$$

Seleccionar entonces un vector \mathbf{p}^{n+1} conjugado a $\mathbf{p}^1, \dots, \mathbf{p}^n$, es decir, un vector \mathbf{p}^{n+1} tal que se cumplan las relaciones

$$\mathbf{p}^j \cdot \mathbf{A}\mathbf{p}^{n+1} = 0, \quad (j = 1, \dots, n). \quad (1.5.1e)$$

TERMINACIÓN: Terminar en el m -ésimo paso si

$$\mathbf{r}^m = \mathbf{0}.$$

Hasta ahora no hemos definido cuáles direcciones conjugadas $\mathbf{p}^1, \dots, \mathbf{p}^n$ utilizaremos, ni cómo las obtendremos. A continuación desarrollaremos un conjunto de direcciones conjugadas partiendo de vectores $\mathbf{w}^0, \dots, \mathbf{w}^n$ linealmente independientes utilizando el método de Gram-Schmidt:

$$\mathbf{p}^1 = \mathbf{w}^0 \quad (1.5.2a)$$

$$\mathbf{p}^{n+1} = \mathbf{w}^n + \sum_{j=1}^n \beta_{n+1,j} \mathbf{p}^j, \quad (1.5.2b)$$

donde, para $j = 1, \dots, n$,

$$\beta_{n+1,j} = -\frac{\mathbf{p}^j \cdot \mathbf{A}\mathbf{w}^n}{d_j}, \quad d_j = \mathbf{p}^j \cdot \mathbf{A}\mathbf{p}^j. \quad (1.5.2c)$$

El método de Gram-Schmidt asegura que se satisfacen las relaciones conjugadas [5]

$$\mathbf{p}^j \cdot \mathbf{A}\mathbf{p}^n = 0, \quad (j \neq n).$$

El algoritmo (1.5.2) tiene la propiedad de que el vector \mathbf{w}^n se puede seleccionar después de que los vectores $\mathbf{p}^1, \dots, \mathbf{p}^n$ se han calculado. Por supuesto que \mathbf{w}^n se debe escoger de tal forma que $\mathbf{w}^0, \dots, \mathbf{w}^n$ o equivalentemente $\mathbf{p}^1, \dots, \mathbf{p}^n, \mathbf{w}^n$ sean linealmente independientes.

En el algoritmo CD se han seleccionado los puntos u^n de tal forma que su residuo r^n sea ortogonal a p^1 . ¿Cómo podemos utilizar este hecho para generar las direcciones p^1, \dots, p^{n+1} ? Supongamos que hemos escogido el punto u^0 y la dirección p^1 , además hemos calculado r^0 , entonces $u^1 = u^0 + \alpha_1 p^1$ donde α_1 se define por medio de la ecuación (1.5.1a). Esta información permite calcular $r^1 = b - A u^1$, como habíamos señalado antes r^1 es ortogonal a p^1 y por lo tanto estos dos vectores son linealmente independientes. Es por esto que podemos generar $p^2 = r^1 - \beta_{2,1} p^1$, donde $\beta_{2,1}$ está definido por la fórmula (1.5.2c). Así p^2 será un vector conjugado a p^1 . Siguiendo así podemos generar la base p^1, \dots, p^{n+1} . Formalizando:

$$p^1 = r^0 \tag{1.5.3a}$$

$$p^{n+1} = r^n + \sum_{j=1}^n \beta_{n+1,j} p^j, \tag{1.5.3b}$$

donde, para $j = 1, \dots, n$,

$$\beta_{n+1,j} = -\frac{p^j \cdot A r^n}{p^j \cdot A p^j}. \tag{1.5.3c}$$

Es importante recalcar que el hiperplano H_n se puede escribir paramétricamente como

$$H_{n+1} : u = u^0 + a_0 r^0 + \dots + a_n r^n,$$

es decir, $H_{n+1} = u^0 + \text{gen} \{r^0, \dots, r^n\}$. Estos hiperplanos son conocidos como espacios afines de Krylov. Por otro lado sabemos que r^{n+1} es ortogonal a este hiperplano o, lo que es lo mismo, es ortogonal a r^0, \dots, r^n . Como esto sucede para cualquier n tenemos las relaciones

$$r^i \cdot r^j = 0, \quad (i \neq j).$$

El costo computacional del algoritmo (1.5.3) es muy alto porque es necesario guardar la información de todos los vectores p^1, \dots, p^{n+1} que se van generando. Veamos si podemos reducir la expresión general del vector p^{n+1} .

PROPOSICIÓN 1.5.1 En el algoritmo (1.5.3) $p^j \cdot A r^n = 0$ si $j < n$.

Demostración:

De (1.4.7a) sabemos que $r^j = r^{j-1} - \alpha_j A p^j$, entonces $\alpha_j A p^j = r^{j-1} - r^j$ y tenemos

$$\alpha_j p^j \cdot A r^n = \alpha_j A p^j \cdot r^n = (r^{j-1} - r^j) \cdot r^n = 0,$$

si $j < n$. Como α_j no es cero, $p^j \cdot A r^n = 0$ ■

Como consecuencia de esta Proposición, modificamos el siguiente término del algoritmo (1.5.3): $p^{n+1} = r^n + \beta_{n+1}p^n$, donde $\beta_{n+1} = -\frac{p^n * \mathbb{A}r^n}{p^n * p^n}$.

Es importante observar que el valor de la función (1.3.1) en un vector u es igual a un medio de la norma (asociada al producto interior $(v, w) = v * \mathbb{A}^{-1}w$) de su residuo $r = \mathbb{A}(u_s - u)$ más una constante, ya que

$$\begin{aligned} F(u) + \frac{1}{2}u_s * b &= \frac{1}{2}(u_s - u) * \mathbb{A}(u_s - u) \\ &= \frac{1}{2}\mathbb{A}e * \mathbb{A}^{-1}\mathbb{A}e = \frac{1}{2}r * \mathbb{A}^{-1}r \\ &= \frac{1}{2}\|r\|_{\mathbb{A}^{-1}}^2. \end{aligned}$$

Entonces, al minimizar la función (1.3.1) sobre el n -ésimo espacio afín de Krylov, implícitamente estamos minimizando la norma del residuo.

El algoritmo de gradiente conjugado que ya incluye el cálculo de la base conjugada es:

Algoritmo CG

PASO INICIAL: Seleccionar un punto u^0 y calcular

$$p^1 = r^0 = b - \mathbb{A}u^0.$$

PASOS ITERATIVOS: Habiendo obtenido u^{n-1} , r^{n-1} y p^n calcular u^n , r^n y p^{n+1} por medio de las fórmulas

$$\begin{aligned} \alpha_n &= \frac{c_n}{d_n}, \\ d_n &= p^n * \mathbb{A}p^n, \\ c_n &= r^{n-1} * r^{n-1} = p^n * r^{n-1} \\ u^n &= u^{n-1} + \alpha_n p^n, \\ r^n &= b - \mathbb{A}u^n, \\ \beta_{n+1} &= -\frac{p^n * \mathbb{A}r^n}{d_n}, \\ p^{n+1} &= r^n + \beta_{n+1}p^n. \end{aligned}$$

TERMINACIÓN: Terminar en el m -ésimo paso si $r^m = 0$.

Capítulo 2

Modificaciones para el caso no simétrico

2.1 Introducción

Discutiremos cuatro métodos derivados del método de gradiente conjugado para sistemas de ecuaciones lineales no simétricos, que en su forma matricial escribiremos como

$$\mathbb{P}\mathbf{u} = \mathbf{a}, \quad (2.1.1)$$

donde \mathbb{P} es no simétrica y no singular. Primero discutiremos una extensión obvia o natural [8] a matrices no simétricas, que se obtiene al aplicar el método de gradiente conjugado a la ecuación resultante de aplicar \mathbb{P}^T a la ecuación (2.1.1); i.e.,

$$\mathbb{K}\mathbf{u} = \mathbb{P}^T\mathbf{a} = \mathbf{b}, \quad (2.1.2)$$

en la que $\mathbb{K} = \mathbb{P}^T\mathbb{P}$. Obsérvese que la matriz \mathbb{K} así definida, es necesariamente de dimensión $N \times N$, positiva definida y simétrica. Por esto mismo, el método de gradiente conjugado es aplicable a la ecuación (2.1.2). Ya que la ecuación (2.1.2) es equivalente* a la original (2.1.1), esta es una forma de extender el método de gradiente conjugado a matrices no simétricas.

Los otros tres métodos son el de residuo conjugado generalizado (GCR), GCR(k) y ORTHOMIN(k). En realidad GCR es el método básico, mientras que GCR(k) y ORTHOMIN(k), son dos simplificaciones (y en cierto sentido aproximaciones de GCR), que disminuyen el trabajo de procesamiento y el espacio de memoria requeridos en cada paso de iteración. Desde luego, estas simplificaciones tienen un precio: mientras que con GCR, si no existieran errores de redondeo, la solución se obtendría en N iteraciones cuando más, en cambio con ORTHOMIN(k) o GCR(k), no hay una garantía similar.

2.2 Extensión natural a matrices no simétricas

Sea \mathbf{u}_n la aproximación enésima a la solución \mathbf{u} , del sistema de ecuaciones lineales (2.1.1), donde \mathbb{P} es una matriz $N \times N$ dimensional no simétrica. Sea \mathbf{r}^n el residuo

*Dos sistemas de ecuaciones son equivalentes cuando toda solución \mathbf{u}_i de una de ellas, es también solución de la otra.

correspondiente, dado por

$$\mathbf{r}^n = \mathbf{a} - \mathbf{P}\mathbf{u}^n.$$

Si aplicamos CG a (2.1.2) obtenemos el algoritmo siguiente:

Algoritmo EN

PASO INICIAL: Escoger una aproximación \mathbf{u}^0 y calcular

$$\mathbf{r}^0 = \mathbf{a} - \mathbf{P}\mathbf{u}^0, \quad \mathbf{p}^1 = \mathbb{P}^T \mathbf{r}^0 = \mathbb{P}^T (\mathbf{a} - \mathbf{P}\mathbf{u}^0).$$

PASOS ITERATIVOS: Habiendo obtenido \mathbf{u}^{n-1} , \mathbf{r}^{n-1} y \mathbf{p}^n calcular \mathbf{u}^n , \mathbf{r}^n y \mathbf{p}^{n+1} por medio de las fórmulas

$$\alpha_n = \frac{c_n}{d_n},$$

$$d_n = \mathbb{P}^2 \mathbf{p}^n + \mathbb{P}^2 \mathbf{p}^n,$$

$$c_n = \mathbb{P}^T \mathbf{r}^{n-1} + \mathbb{P}^T \mathbf{r}^{n-1} = \mathbb{P}^2 \mathbf{r}^{n-1} + \mathbf{r}^{n-1},$$

$$\mathbf{u}^n = \mathbf{u}^{n-1} + \alpha_n \mathbf{p}^n, \tag{2.2.1a}$$

$$\mathbf{r}^n = \mathbf{a} - \mathbb{P}^2 \mathbf{u}^n = \mathbf{r}^{n-1} - \alpha_n \mathbb{P}^2 \mathbf{p}^n, \tag{2.2.1b}$$

$$\beta_{n+1} = -\frac{\mathbb{P}^2 \mathbf{p}^n + \mathbb{P}^2 \mathbf{r}^n}{d_n}, \tag{2.2.1c}$$

$$\mathbf{p}^{n+1} = \mathbb{P}^T \mathbf{r}^n + \beta_{n+1} \mathbf{p}^n. \tag{2.2.1d}$$

TERMINACIÓN Terminar al m -ésimo paso si $\mathbf{r}^m = \mathbf{0}$.

Este algoritmo es aplicable a una matriz \mathbb{P}^2 no simétrica y no singular cualquiera. Sin embargo tiene la desventaja de que la n -ésima aproximación \mathbf{u}^n minimiza a la función (1.3.1) sobre el espacio

$$\text{gen} \{ \mathbb{P}^T \mathbf{r}^0, \dots, \mathbb{P}^T \mathbf{r}^{n-1} \} = \text{gen} \{ \mathbb{P}^T \mathcal{P} \mathbf{c}^0, \dots, (\mathbb{P}^T \mathbb{P})^n \mathbf{u}^0 \}.$$

Es decir que para generar los espacios de búsqueda se ha utilizado la matriz $\mathbb{P}^T \mathbb{P}$ en lugar de \mathbb{P} . Esta multiplicación por \mathbb{P}^T tiende a disminuir la velocidad de convergencia del método.

2.3 Método de residuo conjugado generalizado (GCR)

El método GCR minimiza a la función (1.3.1) sobre los espacios

$$\text{gen} \{r^0, \dots, r^{n-1}\} = \text{gen} \{P^0 e^0, \dots, P^{n-1} e^{n-1}\} = \text{gen} \{P^0 e^0, \dots, P^n e^0\}. \quad (2.3.1)$$

El algoritmo para realizar esta minimización es similar al (2.2.1), pero más laborioso. Supongamos que hemos tomado $p^1 = r^0 = a - P^0 u^0$ y hemos calculado

$$p^i = r^{i-1} + \sum_{j=1}^{i-1} \beta_{ij} p^j \quad (i = 1, \dots, n)$$

donde

$$\beta_{ij} = -\frac{P^i r^{i-1} \cdot P^j p^j}{P^j p^j \cdot P^j p^j}$$

y r^i ha sido calculado por medio de las fórmulas (2.2.1a), (2.2.1b). El método de gradiente conjugado asegura que el vector $P^T r^n$ es ortogonal a p^1, \dots, p^n y por lo tanto a r^0, \dots, r^{n-1} . Para generar un vector p^{n+1} ortogonal a p^1, \dots, p^n necesitamos que r^n sea linealmente independiente de r^0, \dots, r^{n-1} . ¿Qué condiciones necesitamos imponer sobre P para que el vector r^n cumpla con esta condición?

Sea $\alpha_0 r^0 + \dots + \alpha_n r^n = 0$, ya que $P^T r^i$ es ortogonal a r^j si i es mayor que j tenemos

$$\mathbb{P}^T r^n \cdot (\alpha_0 r^0 + \dots + \alpha_n r^n) = \alpha_n r^n \cdot \mathbb{P}^T r^n = 0,$$

si $r^n \cdot \mathbb{P}^T r^n$ es distinto de cero, $\alpha_n = 0$. Entonces

$$\alpha_0 r^0 + \dots + \alpha_{n-1} r^{n-1} = 0,$$

por lo que $\alpha_0 = \dots = \alpha_{n-1} = 0$ porque r^0, \dots, r^{n-1} son linealmente independientes. Entonces basta con pedir que $u \cdot \mathbb{P}^T u$ sea distinto de cero para toda $u \neq 0$ para que r^n sea linealmente independiente de r^0, \dots, r^{n-1} . Si suponemos que la parte simétrica de P es positiva definida tenemos

$$0 < u \cdot \frac{1}{2} (\mathbb{P}^T + P) u = \frac{1}{2} (u \cdot \mathbb{P}^T u + u \cdot P u) = u \cdot \mathbb{P}^T u$$

y la condición se cumple.

En el método de gradiente conjugado inicialmente aparecieron los términos β_{ij} pero en ese caso simplificamos la expresión (1.5.3) porque β_{ij} era cero si $j < i-1$.

Ahora tenemos

$$\beta_{ij} = -\frac{\mathbf{P}\mathbf{r}^{i-1} \star \mathbb{P}\mathbf{p}^j}{\mathbb{P}\mathbf{p}^j \star \mathbb{P}\mathbf{p}^j} \quad (j = 1, \dots, i-1)$$

desarrollando el término $\mathbf{P}\mathbf{r}^{i-1} \star \mathbb{P}\mathbf{p}^j$, utilizando (2.2.1b), obtenemos

$$\begin{aligned} \mathbf{P}\mathbf{r}^{i-1} \star \mathbf{p}^j &= \alpha_j \mathbb{P}^2 \mathbf{r}^{i-1} \star (\mathbf{r}^{j-1} - \mathbf{r}^j) \\ &= \mathbf{r}^{i-1} \star (\mathbb{P}^T \mathbf{r}^{j-1} - \mathbb{P}^T \mathbf{r}^j). \end{aligned}$$

Sabemos que $\mathbf{r}^i \star \mathbb{P}^T \mathbf{r}^j$ es cero si $j > i-1$ pero eso no sucede en este caso.

Si \mathbf{P} es una matriz no simétrica con parte simétrica positiva definida obtenemos el algoritmo del residuo conjugado generalizado (GCR)

Algoritmo GCR

PASO INICIAL: Escoger una aproximación \mathbf{u}^0 . Calcular $\mathbf{p}^0 = \mathbf{r}^0 = \mathbf{a} - \mathbb{P}\mathbf{u}^0$.

PASOS ITERATIVOS: Habiendo obtenido \mathbf{u}^{n-1} , \mathbf{r}^{n-1} y \mathbf{p}^n calcular α_n , \mathbf{r}^n y \mathbf{p}^{n+1} como sigue:

$$\begin{aligned} \alpha_n &= \frac{\mathbb{P}\mathbf{p}^n \star \mathbf{r}^{n-1}}{\mathbb{P}\mathbf{p}^n \star \mathbb{P}\mathbf{p}^n}, \\ \mathbf{u}^n &= \mathbf{u}^{n-1} + \alpha_n \mathbf{p}^n, \\ \mathbf{r}^n &= \mathbf{a} - \mathbb{P}\mathbf{u}^n = \mathbf{r}^{n-1} - \alpha_n \mathbb{P}^2 \mathbf{p}^n \\ \beta_{n+1,j} &= -\frac{\mathbb{P}^2 \mathbf{r}^n \star \mathbb{P}\mathbf{p}^j}{\mathbb{P}\mathbf{p}^j \star \mathbb{P}\mathbf{p}^j}, \\ \mathbf{p}^{n+1} &= \mathbf{r}^n + \sum_{j=1}^n \beta_{n+1,j} \mathbf{p}^j \end{aligned}$$

TERMINACIÓN: Terminar en el m -ésimo paso si $\mathbf{r}^m = \mathbf{0}$

Si no existieran errores de redondeo, GCR obtendría la solución exacta, al igual que gradiente conjugado, en a lo más N iteraciones. Sin embargo al trabajar con el sistema de vectores (2.3.1) es necesario mantener en la memoria de la máquina a la base de vectores del espacio generado por éstos, por lo que el trabajo y espacio de memoria requeridos en cada iteración es muy grande si N lo es. Los métodos de GCR(k) y ORTHOMIN(k) son dos modificaciones a GCR para disminuir el trabajo de procesamiento y el espacio de memoria requeridos.

A diferencia de lo dicho en la Sección 1.5, en el método de GCR se minimiza la norma euclidiana del residuo de la función (1.3.1) sobre cada subespacio afín de

Krylov en cada paso del algoritmo, ya que en este caso

$$F(\mathbf{u}) + \frac{1}{2}\mathbf{u}_k \cdot \mathbf{b} = \frac{1}{2}\mathbf{P}^T \mathbf{e} \cdot \mathbf{P}^T \mathbf{e} = \frac{1}{2}\|\mathbf{r}\|_2^2.$$

2.4 Métodos GCR(k) y ORTHOMIN(k)

Vinsonne [14] propuso un método llamado ORTHOMIN(k) que es significativamente menos costoso por iteración. En vez de hacer \mathbf{p}^{n+1} $\mathbf{P}^T \mathbf{P}$ -ortogonal a todas las direcciones precedentes $\{\mathbf{p}^1, \dots, \mathbf{p}^n\}$, se puede hacer $\mathbf{P}^T \mathbf{P}$ -ortogonal únicamente a los últimos k vectores $\{\mathbf{p}^{n+1-k}, \dots, \mathbf{p}^n\}$, de tal forma que

$$\mathbf{p}^{n+1} = \mathbf{r}^n + \sum_{j=n-k+1}^n \beta_{n+1,j} \mathbf{p}^j,$$

con $\beta_{n+1,j}$, $j = n-k+1, \dots, n$ definidos como en GCR. Así, únicamente tenemos que guardar la información de, a lo más, k vectores.

Otra posibilidad es recomenzar con cada $k+1$ iteraciones, tomando como aproximación inicial al último vector encontrado. Nuevamente, a lo más k vectores dirección tienen que ser salvados, de tal forma que el espacio de memoria utilizado es el mismo que el de ORTHOMIN(k). Sin embargo el costo por iteración es menor ya que en general un número menor que k vectores se usan para encontrar \mathbf{p}^{n+1} . Este método se conoce como GCR(k).

Capítulo 3

Nuevo método para el caso no simétrico

3.1 Introducción

Revisaremos nuevamente el método de gradiente conjugado bajo una óptica distinta, que nos permitirá introducir una idea propuesta por I. Herrera [10] y cuyo desarrollo, análisis y puesta a prueba como método de solución de sistemas no simétricos, es uno de los objetivos centrales de esta tesis. Se construirán aproximaciones a la solución de la ecuación

$$Au = b$$

(A positiva definida y simétrica) que minimicen el error directamente, sin utilizar la función auxiliar (1.3.1) definida en el capítulo 1. Minimizaremos la ecuación en subespacios anidados

$$H_1 \subset \dots \subset H_n$$

de la siguiente manera: suponiendo que el vector u_1 es conocido, encontraremos el vector u^0 en H_n que deje el mínimo error $e^0 = u_n - u^0$ en ese subespacio. Para esto haremos uso de un producto escalar cualquiera (\cdot, \cdot) y al encontrar la expresión general del algoritmo definiremos cual es el producto interior que conviene utilizar, para que, todos los términos que aparezcan se puedan calcular.

En el caso no simétrico modificaremos únicamente el producto escalar a utilizar, es decir, los espacios de búsqueda serán los mismos que en el caso simétrico, pero tendremos la ventaja de que el producto interior definiéndolo por la norma euclidiana, al que hemos denotado (\cdot, \cdot) , permitirá conocer todos los términos que necesitamos para iniciar el algoritmo.

En este Capítulo modificaremos el lenguaje que se ha utilizado, propio del cálculo y la geometría analítica, pues es más cómodo para el enfoque escogido utilizar ahora el lenguaje del álgebra lineal.

3.2 Planteamiento general del problema

Queremos resolver la ecuación

$$Au = b, \quad (3.2.1)$$

donde A es una matriz positiva definida y simétrica de $N \times N$.

Sea u_n la solución de la ecuación (3.2.1) (Ver. 2). Tomemos un subespacio vectorial H_1 de \mathbb{R}^N de una dimensión y supongamos que (\cdot, \cdot) es un producto escalar que permite calcular la proyección u^1 de u_n sobre H_1 . u^1 es la primera aproximación a la solución u_n . Es importante hacer notar que u^1 es el vector de H_1 que deja el

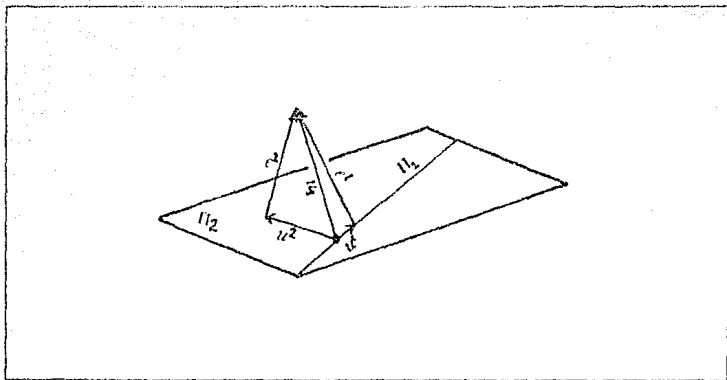


FIGURA 2. u^1 y u^2 , proyecciones de u , sobre Π_1 y Π_2 , respectivamente.

error de magnitud menor respecto al producto escalar (\cdot, \cdot) . Es decir si el error que corresponde a u^1 es

$$e^1 = u_s - u^1$$

y v es un elemento de Π_1 con error

$$e = u_s - v, \quad (3.2.2)$$

entonces $\|e^1\| \leq \|e\|$, donde $\|u\|^2 = (u, u)$.

Ahora queremos encontrar un vector u^2 tal que la norma del error correspondiente e^2 sea menor que la de e^1 . Para esto necesitamos considerar un espacio Π_2 distinto de Π_1 , ya que ningún vector de éste último cumple con esta condición. Sea Π_2 un subespacio de \mathbb{R}^N de dimensión dos que contenga a Π_1 y u^2 la proyección de u_s sobre Π_2 . Aseguramos que u^2 satisface la condición pedida. Veamos:

Sean $\beta_1 = \{p^1\}$ y $\beta_2 = \{p^1, p^2\}$ bases ortogonales de Π_1 y Π_2 respectivamente, entonces

$$u^1 = \frac{(u_s, p^1)}{(p^1, p^1)} p^1,$$

$$u^2 = \frac{(u_s, p^1)}{(p^1, p^1)} p^1 + \frac{(u_s, p^2)}{(p^2, p^2)} p^2.$$

Usando la ecuación (3.2.2) tenemos

$$\begin{aligned} \mathbf{e}^1 &= \mathbf{u}_s - \frac{(\mathbf{u}_s, \mathbf{p}^1)}{(\mathbf{p}^1, \mathbf{p}^1)} \mathbf{p}^1, \\ \mathbf{e}^2 &= \mathbf{u}_s - \frac{(\mathbf{u}_s, \mathbf{p}^1)}{(\mathbf{p}^1, \mathbf{p}^1)} \mathbf{p}^1 - \frac{(\mathbf{u}_s, \mathbf{p}^2)}{(\mathbf{p}^2, \mathbf{p}^2)} \mathbf{p}^2. \end{aligned}$$

Las normas al cuadrado de \mathbf{e}^1 y \mathbf{e}^2 son

$$\begin{aligned} \|\mathbf{e}^1\|^2 &= (\mathbf{u}_s, \mathbf{u}_s) - \frac{(\mathbf{u}_s, \mathbf{p}^1)^2}{(\mathbf{p}^1, \mathbf{p}^1)}, \\ \|\mathbf{e}^2\|^2 &= (\mathbf{u}_s, \mathbf{u}_s) - \frac{(\mathbf{u}_s, \mathbf{p}^1)^2}{(\mathbf{p}^1, \mathbf{p}^1)} - \frac{(\mathbf{u}_s, \mathbf{p}^2)^2}{(\mathbf{p}^2, \mathbf{p}^2)} \end{aligned}$$

entonces

$$\|\mathbf{e}^2\|^2 = \|\mathbf{e}^1\|^2 - \frac{(\mathbf{u}_s, \mathbf{p}^2)^2}{(\mathbf{p}^2, \mathbf{p}^2)}. \quad (3.2.3)$$

Como $\frac{(\mathbf{u}_s, \mathbf{p}^2)^2}{(\mathbf{p}^2, \mathbf{p}^2)} \geq 0$ se desprende de (3.2.3) que

$$\|\mathbf{e}^2\| \leq \|\mathbf{e}^1\|.$$

De esta forma podemos construir una sucesión de aproximaciones $\mathbf{u}^1, \dots, \mathbf{u}^n$ tales que:

$$\mathbf{u}^k \in \Pi_k \quad (k = 1, \dots, n)$$

y cuyos errores correspondientes cumplan con la siguiente relación

$$\|\mathbf{e}^1\| \geq \|\mathbf{e}^2\| \geq \dots \geq \|\mathbf{e}^n\|. \quad (3.2.4)$$

Además

$$\dim(\Pi_k) = k \quad (k = 1, \dots, n), \quad (3.2.5)$$

$$\Pi_1 \subset \Pi_2 \subset \dots \subset \Pi_n. \quad (3.2.6)$$

Si construyéramos N aproximaciones, \mathbf{u}^N sería la solución exacta.

Una manera de resolver la ecuación (3.2.1) que se basa en la misma idea es

tomar una primera aproximación u^0 a la solución y calcular su error

$$e^0 = u_s - u^0. \quad (3.2.7)$$

Sabemos entonces que $u_s = u^0 + e^0$. Si encontramos aproximaciones w^k a e^0 por medio de las proyecciones de este último a los espacios Π_k , como lo hicimos anteriormente con u_s , y en este caso

$$u^k = u^0 + w^k, \quad w^k \in \Pi_k \quad (3.2.8)$$

estos vectores son aproximaciones a u_s .

Todo lo demostrado para u^k es válido para w^k , a saber:

$$w^N = e^0,$$

es decir,

$$u^N = u_s;$$

$$\|e^0 - w^k\| \geq \|e^0 - w^{k+1}\|. \quad (3.2.9)$$

Pero, ¿qué pasa con $e^k = u_s - u^k$? Por (3.2.8)

$$e^k = u_s - u^0 - w^k = e^0 - w^k \quad (3.2.10)$$

y su norma, tomando en cuenta (3.2.9), satisface

$$\|e^k\| = \|e^0 - w^k\| \geq \|e^0 - w^{k+1}\| = \|e^{k+1}\|.$$

Se cumple la condición (3.2.4) impuesta sobre e^k . Esta última será la forma de proceder en este trabajo.

Planteado así el problema quedan varios puntos fundamentales por resolver:

- 1) ¿Qué espacios vectoriales conviene escoger?
- 2) ¿Cómo podemos construir las bases de estos espacios de forma eficiente?
- 3) ¿Qué productos escalares permiten encontrar la proyección de u_s sobre Π_k ?

La forma de responder a estas preguntas es lo que caracteriza al método de gradiente conjugado.

3.3 Explicación del método

Queremos escoger los espacios Π_n de tal forma que construir la base ortogonal $\{p^1, \dots, p^n\}$ de cada uno de ellos requiera poco trabajo computacional. Analicemos

los espacios de Krylov definidos como sigue

$$\Pi_n = \text{gen} \{ \mathbf{A}e^0, \mathbf{A}^2e^0, \dots, \mathbf{A}^ne^0 \} \quad (3.3.1)$$

PROPOSICIÓN 3.3.1 *Los vectores $\mathbf{A}e^0, \dots, \mathbf{A}^ne^0$ son linealmente independientes y, por lo tanto, la dimensión del n -ésimo espacio de Krylov es n si no se ha encontrado la solución.*

Demostración:

Supongamos que $\mathbf{A}e^0, \mathbf{A}^2e^0, \dots, \mathbf{A}^ne^0$ son linealmente dependientes. Entonces existen a_1, a_2, \dots, a_n no todos cero tales que

$$a_1\mathbf{A}e^0 + a_2\mathbf{A}^2e^0 + \dots + a_n\mathbf{A}^ne^0 = 0$$

y

$$\mathbf{A}(a_1e^0 + a_2\mathbf{A}e^0 + \dots + a_n\mathbf{A}^{n-1}e^0) = 0. \quad (3.3.2)$$

Debido a que \mathbf{A} es una matriz no singular (por ser positiva definida)

$$a_1e^0 + a_2\mathbf{A}e^0 + \dots + a_n\mathbf{A}^{n-1}e^0 = 0.$$

Si $a_1 \neq 0$ (si fuera cero haríamos el paso (3.3.2) nuevamente hasta encontrar $a_k \neq 0$)

$$e^0 = -\frac{a_2}{a_1}\mathbf{A}e^0 - \dots - \frac{a_n}{a_1}\mathbf{A}^{n-1}e^0.$$

Tomando en cuenta la ecuación (3.2.7) tenemos

$$u_n = u^0 - \frac{a_2}{a_1}\mathbf{A}e^0 - \dots - \frac{a_n}{a_1}\mathbf{A}^{n-1}e^0.$$

Tenemos, entonces, a u_n en el espacio trasladado de Krylov de orden $n-1$, lo que implica que $u^{n-1} = u_n$, que contradice nuestra hipótesis. Por lo cual $\mathbf{A}e^0, \dots, \mathbf{A}^ne^0$ son linealmente independientes y la dimensión de Π_n es n . \square

Si las proyecciones w^n de e^0 están en Π_n las aproximaciones

$$u^n = u^0 + w^n$$

pertenecen al espacio trasladado (espacio afín) de Krylov

$$\Pi_{n,T} = u^0 + \Pi_n.$$

Comprobemos que estos espacios satisfacen las condiciones (3.2.4) y (3.2.5) impuestas sobre los Π_n :

PROPOSICIÓN 3.3.2 *Los espacios de Krylov están anidados, es decir*

$$\Pi_1 \subset \Pi_2 \subset \dots \subset \Pi_N.$$

Demostración:

Por (3.3.1)

$$\Pi_n = \text{gen} \{Ae^0, A^2e^0, \dots, A^ne^0\},$$

$$\Pi_{n+1} = \text{gen} \{Ae^0, A^2e^0, \dots, A^ne^0, A^{n+1}e^0\}$$

entonces $\Pi_n \subset \Pi_{n+1}$ y por lo tanto

$$\Pi_1 \subset \Pi_2 \subset \dots \subset \Pi_N \quad \square$$

Para demostrar las propiedades de los espacios de Krylov que nos interesan necesitamos usar varios resultados que presentamos a continuación.

PROPOSICIÓN 3.3.3 *La transformación del espacio n -ésimo de Krylov, bajo A , está contenida en el espacio de Krylov una dimensión mayor. Es decir*

$$A\Pi_n \subset \Pi_{n+1}$$

Demostración:

$$\Pi_n = \text{gen} \{Ae^0, \dots, A^ne^0\}$$

o, lo que es lo mismo

$$\Pi_n = \{\alpha_1 Ae^0 + \dots + \alpha_n A^ne^0 / \alpha_i \in \mathbb{R}, \quad i = 1, \dots, n\}$$

entonces

$$\tilde{\Pi}_n \Pi_n = \{\alpha_1 A^2e^0 + \dots + \alpha_n A^{n+1}e^0 / \alpha_i \in \mathbb{R}, \quad i = 1, \dots, n\}$$

$$\subset \{\beta_0 Ae^0 + \beta_1 A^2e^0 + \dots + \beta_n A^{n+1}e^0 / \beta_i \in \mathbb{R}, \quad i = 1, \dots, n\} = \Pi_{n+1} \quad \square$$

PROPOSICIÓN 3.3.4 *Si el transformado de e^0 bajo A es elemento de Π_n , entonces la solución u_s es elemento de $\Pi_{n,T}$.*

Demostración:

Se demuestra de forma análoga a la PROPOSICIÓN 3.3.1, pero en este caso

obtenemos

$$\begin{aligned} \mathbf{u}_s &= \mathbf{u}^n - \frac{\alpha_1}{a_{n+1}} \mathbf{e}^0 - \dots - \frac{\alpha_n}{a_{n+1}} \mathbf{A}^{n-1} \mathbf{e}^0 \\ &= \mathbf{u}^0 + \mathbf{w}^n - \frac{\alpha_1}{a_{n+1}} \mathbf{e}^0 - \dots - \frac{\alpha_n}{a_{n+1}} \mathbf{A}^{n-1} \mathbf{e}^0, \end{aligned}$$

como \mathbf{w}^n es elemento de Π_n , entonces \mathbf{u}_s está en $\Pi_{n,T} \boxplus$

Con esto podemos demostrar la dos propiedades de los espacio de Krylov antes mencionadas.

PROPOSICIÓN 3.3.5. *Propiedad de ortogonalidad. Si \mathbf{v} es ortogonal a Π_n , entonces $\mathbf{A}\mathbf{v}$ es ortogonal Π_{n-1} .*

Demostración:

Sea \mathbf{w}^{n-1} un elemento de Π_{n-1} . Tenemos, entonces,

$$(\mathbf{A}\mathbf{v}, \mathbf{w}^{n-1}) = (\mathbf{v}, \mathbf{A}\mathbf{w}^{n-1}) = 0$$

ya que $\mathbf{A}\mathbf{w}^{n-1}$ es elemento de Π_n por la PROPOSICIÓN 3.3.3. Así $\mathbf{A}\mathbf{v}$ es ortogonal a Π_n como queríamos demostrar \square

PROPOSICIÓN 3.3.6 *Propiedad de pertenencia. Sea $\mathbf{e} = \mathbf{u}_s - \mathbf{u}^n$, donde \mathbf{u}^n es elemento de $\Pi_{n,T}$. Entonces $\mathbf{A}\mathbf{e}$ pertenece a Π_{n+1} .*

Demostración:

Ya que $\mathbf{u}^n \in \Pi_{n,T}$, $\mathbf{u}^n = \mathbf{u}^0 + \mathbf{w}^n$ para algún $\mathbf{w}^n \in \Pi_n$.

$$\mathbf{A}\mathbf{e} = \mathbf{A}(\mathbf{u}_s - \mathbf{u}^n - \mathbf{w}^n) = \mathbf{A}(\mathbf{u}_s - \mathbf{u}^0) - \mathbf{A}\mathbf{w}^n = \mathbf{A}\mathbf{e}^0 - \mathbf{A}\mathbf{w}^n.$$

Tenemos $\mathbf{A}\mathbf{e}^0 \in \Pi_{n+1}$ por (3.3.1) y $\mathbf{A}\mathbf{w}^n \in \Pi_{n+1}$ por la PROPOSICIÓN 3.3.3, esto indica que $\mathbf{A}\mathbf{e}$ es elemento de $\Pi_{n+1} \boxplus$

Como habíamos visto antes, construiremos aproximaciones \mathbf{u}^n con elementos del espacio trasladado de Krylov correspondiente. El error de estas aproximaciones es un vector ortogonal a este espacio, que parte de éste y va hasta la solución exacta \mathbf{u}_s de la ecuación (3.2.1). Las propiedades de pertenencia y ortogonalidad nos dicen que en esta clase de aproximaciones el transformado del error es ortogonal a los espacios de Krylov de orden mayor y pertenece al siguiente de estos espacios. Veamos como estos resultados permiten construir iterativamente la base ortogonal que buscábamos.

Sabemos, entonces, que

$$\mathbf{A}\mathbf{e}^n \perp \Pi_{n-1} \quad (3.3.3)$$

y

$$\mathbf{A}\mathbf{e}^n \in \Pi_{n+1} \quad (3.3.4)$$

Supongamos que $\{p^1, \dots, p^n\}$ es una base ortogonal de Π_n , donde p^m pertenece a Π_m , $m = 1, \dots, n$. Queremos encontrar un vector p^{n+1} en Π_{n+1} que sea ortogonal a p^m , $m = 1, \dots, n$ y tal que $\{p^1, \dots, p^{n+1}\}$ genere a Π_{n+1} .

p^n es un elemento de Π_{n+1} por ser Π_n subespacio de Π_{n+1} ; $\mathbf{A}e^n$ pertenece también a ese espacio por (3.3.4). Además, tanto p^n como $\mathbf{A}e^n$ son ortogonales a Π_{n-1} por (3.3.3) y son linealmente independientes (suponiendo que no hemos encontrado la solución) por la PROPOSICIÓN 3.3.5. Entonces cualquier elemento en el espacio generado por estos dos vectores es ortogonal a Π_{n-1} y pertenece a Π_{n+1} . Proponemos p^{n+1} como

$$p^{n+1} = \mathbf{A}e^n + \beta_{n+1}p^n. \quad (3.3.5)$$

Definido así, el único requisito que falta satisfacer es que p^{n+1} sea ortogonal a p^n . Esto lo logramos haciendo $(p^{n+1}, p^n) = 0$, veamos:

$$(p^{n+1}, p^n) = (\mathbf{A}e^n, p^n) + \beta_{n+1}(p^n, p^n) = 0$$

de lo que resulta

$$\beta_{n+1} = -\frac{(\mathbf{A}e^n, p^n)}{(p^n, p^n)} \quad (3.3.6)$$

Podemos construir la base de Π_n aplicando la construcción anterior iterativamente. Como primer elemento tomaremos

$$p^1 = b - \mathbf{A}u^0.$$

Con esto hemos respondido las dos primeras preguntas de la sección anterior. Para responder la tercera, veamos cómo se expresan las aproximaciones u^{n+1} en esta base.

Ya que $u^{n+1} = u^0 + w^{n+1}$, donde w^{n+1} es la proyección de e^0 sobre Π_{n+1} , tenemos

$$\begin{aligned} u^{n+1} &= u^0 + \sum_{i=1}^{n+1} \frac{(e^0, p^i)}{(p^i, p^i)} p^i \\ &= u^0 + \sum_{i=1}^n \frac{(e^0, p^i)}{(p^i, p^i)} p^i + \frac{(e^0, p^{n+1})}{(p^{n+1}, p^{n+1})} p^{n+1} = u^n + \alpha_{n+1} p^{n+1} \end{aligned} \quad (3.3.7)$$

donde

$$\alpha_{n+1} = \frac{(e^0, p^{n+1})}{(p^{n+1}, p^{n+1})}. \quad (3.3.8)$$

Para obtener las aproximaciones iterativamente conviene expresar (e^0, p^{n+1}) en términos de (e^n, p^{n+1}) , trataremos de establecer una relación entre estos términos. De (3.2.10) obtenemos

$$e^0 = e^n + w^n,$$

por tanto

$$(e^0, p^{n+1}) = (e^n, p^{n+1}) + (w^n, p^{n+1}).$$

w^n es elemento de H_n y p^{n+1} es ortogonal a este espacio por lo cual $(w^n, p^{n+1}) = 0$ y

$$(e^0, p^{n+1}) = (e^n, p^{n+1}).$$

Substituyendo (3.3.5) en esta ecuación obtenemos

$$(e^0, p^{n+1}) = (e^n, \mathcal{A}e^n) : \beta_{n+1}(e^n, p^n) = (e^n, \mathcal{A}e^n)$$

ya que $(e^n, p^n) = 0$ porque e^n es ortogonal a H_n . La nueva expresión para (3.3.8) es

$$\alpha_{n+1} = \frac{(e^n, \mathcal{A}e^n)}{(p^{n+1}, p^{n+1})} \quad (3.3.9).$$

Para obtener β_{n+1} podríamos utilizar cualquier producto escalar, pues cada término en la expresión (3.3.6) es conocido. No es el caso para α_{n+1} ya que en (3.3.9) aparece el vector desconocido e^n .

¿Qué productos escalares permiten conocer $(e^n, \mathcal{A}e^n)$?

Cuando \mathcal{A} es una matriz simétrica y positiva definida, \mathcal{R} y cada una de sus potencias definen productos interiores que hacen posible calcular α_{n+1} . Utilizaremos el más sencillo de ellos, dado por

$$(u, v) = u \cdot \mathcal{R}v$$

donde \cdot es el producto interior usual en los reales.

Con esto las ecuaciones (3.3.6) y (3.3.9) se transforman en

$$\alpha_{n+1} = \frac{r^n \cdot r^n}{p^{n+1} \cdot \mathcal{R}p^{n+1}},$$

$$\beta_{n+1} = -\frac{p^n \cdot \mathcal{R}r^n}{p^n \cdot \mathcal{R}p^n},$$

donde $r^n = \mathcal{A}e^n$.

Al hacer el planteamiento general del problema únicamente imponemos la condición de que el producto escalar (\cdot, \cdot) permitiera calcular la proyección de e^n sobre los Π_n . Al hacer el desarrollo del método hemos presupuesto que si la magnitud de e^n decrece con respecto a la norma dada por (\cdot, \cdot) , también decrece con respecto a la dada por \ast . Esta condición no se expuso explícitamente para no complicar el texto, sin embargo, el producto escalar $u \ast Av$ permite asegurar que esto sucede.

3.4 Construcción para la matriz no simétrica

Ismael Herrera ha observado que en aplicaciones a matrices no simétricas es posible minimizar directamente la norma euclídea del error y no solamente el residuo, como se hizo en el Capítulo II (Sección 2.3). Esto da lugar a simplificaciones importantes en los cálculos y aceleración de la convergencia, cuyo análisis y puesta a prueba, como se ha dicho, ha sido uno de los objetivos principales de esta tesis.

Consideremos nuevamente la ecuación (2.1.1), entonces todos los desarrollos anteriores son aplicables a la ecuación (2.1.2) asociada con ésta, incluyendo (3.3.5), (3.3.6), (3.3.7) y (3.3.9). Si definimos

$$r^n = a - \mathbb{P}u^n = \mathbb{P}e^n, \quad q^1 = r^0, \quad q^{n+1} = \mathbb{P}e^n + \beta_{n+1}q^n,$$

entonces $p^{n+1} = \mathbb{P}^T q^{n+1}$ y podemos reescribir estas ecuaciones como

$$\begin{aligned} p^1 &= \mathbb{P}^T q^1, \\ p^{n+1} &= Ae^n + \beta_{n+1}p^n = \mathbb{P}^T(\mathbb{P}e^n + \beta_{n+1}q^n), \\ \beta_{n+1} &= -\frac{(Ae^n, p^n)}{(p^n, p^n)} = -\frac{(\mathbb{P}^T \mathbb{P}e^n, \mathbb{P}^T q^n)}{(\mathbb{P}^T q^n, \mathbb{P}^T q^n)} = -\frac{(\mathbb{P}^T r^n, \mathbb{P}^T q^n)}{(\mathbb{P}^T q^n, \mathbb{P}^T q^n)}, \\ u^{n+1} &= u^n + \alpha_{n+1} \mathbb{P}^T q^{n+1}, \\ \alpha_{n+1} &= \frac{(e^n, Ae^n)}{(p^{n+1}, p^{n+1})} = \frac{(e^n, \mathbb{P}^T p^n)}{(\mathbb{P}^T q^{n+1}, \mathbb{P}^T q^{n+1})} = \frac{(r^n, r^n)}{(\mathbb{P}^T q^{n+1}, \mathbb{P}^T q^{n+1})}. \end{aligned}$$

En este caso todos los términos que aparecen al calcular α_n y β_n son conocidas por lo que se pueden calcular independientemente del producto interno que se utilice, es por esto que podemos utilizar el producto interior \ast y minimizar directamente la norma euclídea del error sobre cada subespacio afín de Krylov.

Algoritmo NM

PASO INICIAL: Seleccionar un punto u^0 y calcular

$$q^1 = r^0 = a - \mathbb{P}u^0.$$

PASOS ITERATIVOS: Habiendo obtenido u^{n-1}, r^{n-1} y q^n calcular u^n, r^n y q^{n+1} por medio de las fórmulas:

$$\alpha_n = \frac{r^{n-1} + r^{n-1}}{P^T q^n + P^T q^n}$$

$$u^n = u^{n-1} + \alpha_n P^T q^n,$$

$$r^n = b - P u^n,$$

$$\beta_{n+1} = -\frac{P^T r^n + P^T q^n}{P^T q^n + P^T q^n},$$

$$q^{n+1} = r^n + \beta_{n+1} q^n.$$

TERMINACIÓN: Terminar en el m -ésimo paso si $r^m = 0$.

En el trabajo numérico de esta tesis obtendremos soluciones aproximadas dando un margen máximo al residuo r^n . Esto es necesario porque el error mismo e^n se desconoce. Sin embargo, es conveniente observar que el residuo constituye una medida indirecta del error, ya que, bajo las hipótesis establecidas en esta tesis (P no singular) siempre existe un número M tal que

$$\|e\|_2 \leq M \|r\|_2.$$

Así, al dar una cota a $\|r^n\|_2$ ella conlleva otra de $\|e^n\|_2$.

Capítulo 4

Resultados numéricos y conclusiones

4.1 Introducción

Probaremos el nuevo método (NM) con un problema de transporte con difusión en estado estacionario y compararemos resultados numéricos con GCR(3), ORTHOMX(3) y la extensión natural del método de gradiente conjugado.

4.2 Solución numérica de un problema específico

Trabajaremos con un problema de transporte con difusión [1]:

$$\frac{\partial c}{\partial t} + \nabla \cdot \mathbf{vc} - \sum_{i=1}^3 \frac{\partial}{\partial x_i} \left(K \frac{\partial c}{\partial x_i} \right) = 0, \quad (4.2.1)$$

donde c es la concentración, \mathbf{v} es la velocidad, y K el coeficiente de difusión. La concentración será independiente del tiempo y K constante, igual a uno, con lo que la ecuación (4.2.1) se transforma en

$$\nabla \cdot \mathbf{vc} - \sum_{i=1}^3 \left(\frac{\partial^2 c}{\partial x_i^2} \right) = 0. \quad (4.2.2)$$

Las condiciones en la frontera están dadas como se muestran en la Figura 3.

Resolveremos la ecuación (4.2.2) sobre una región cúbica con celdas de lado h , tomando constante la velocidad en la dirección $(1, 1, 1)$ y con $\|\mathbf{v}\|_2 = Pe/h$, donde Pe es el número de Peclet.

Discretizaremos la ecuación usando el método de celdas. Sean x_{ijk} ($i, j, k = 1, \dots, N$) los puntos en los que queremos aproximar el valor de la concentración cada uno de los cuales es centro de la celda E_{ijk} . Denotaremos $x_{i+\frac{1}{2}, j, k}$, $x_{i-\frac{1}{2}, j, k}$, $x_{i, j+\frac{1}{2}, k}$, $x_{i, j-\frac{1}{2}, k}$, $x_{i, j, k+\frac{1}{2}}$ y $x_{i, j, k-\frac{1}{2}}$ a los puntos sobre los lados de la celda, que están colocados como se ilustra en la Figura 4.

Por el teorema de la divergencia

$$\int_{E_{ijk}} \nabla \cdot \mathbf{vc} \, dx = \int_{\partial E_{ijk}} \mathbf{vc} \cdot \mathbf{n} \, dx,$$

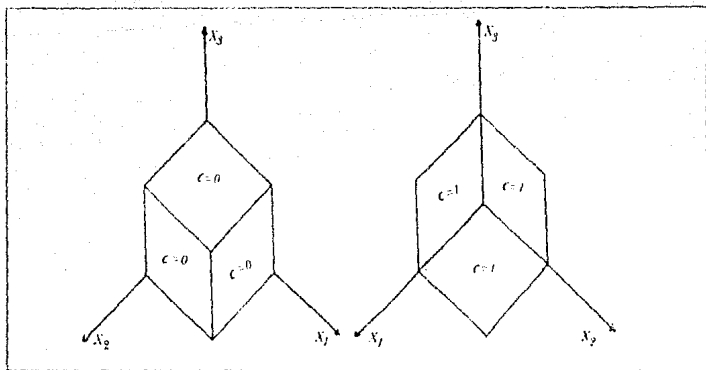


FIGURA 3. Condiciones a la frontera para el problema de transporte con difusión.

donde n es un vector normal en cada punto de la frontera δE_{ijk} . En este caso E_{ijk} es un cubo y la integral sobre C_1 , uno de sus caras (como se muestra en el Figura 3) está determinada por el vector $n = (1, 0, 0)$, por lo que

$$\int_{C_1} cv \cdot n \, dx = \int_{C_1} cv \, dx,$$

donde el vector $v = (v, v, v)$. Si aproximamos el valor de c sobre la cara C_1 con el valor en el punto medio $x_{i+\frac{1}{2},j,k}$ obtenemos

$$\int_{C_1} cv \cdot n \, dx \approx v h^2 c_{i+\frac{1}{2},j,k}$$

y, por lo tanto,

$$\int_{C_1} cv \cdot n \, dx + \int_{C_2} cv \cdot n \, dx \approx v h^2 \{c_{i+\frac{1}{2},j,k} - c_{i-\frac{1}{2},j,k}\}.$$

Sin embargo en esta ecuación aparecen los términos desconocidos $c_{i+\frac{1}{2},j,k}$ y $c_{i-\frac{1}{2},j,k}$ por lo que los aproximaremos utilizando el promedio de los valores $c_{i+1,j,k}$ y $c_{i,j,k}$ o $c_{i-1,j,k}$ y $c_{i,j,k}$ según sea el caso, con esto obtenemos la siguiente aproximación al valor de la integral

$$\int_{C_1} cv \cdot n \, dx + \int_{C_2} cv \cdot n \, dx \approx v \frac{h^2}{2} \{c_{i+1,j,k} - c_{i-1,j,k}\}. \quad (4.2.3)$$

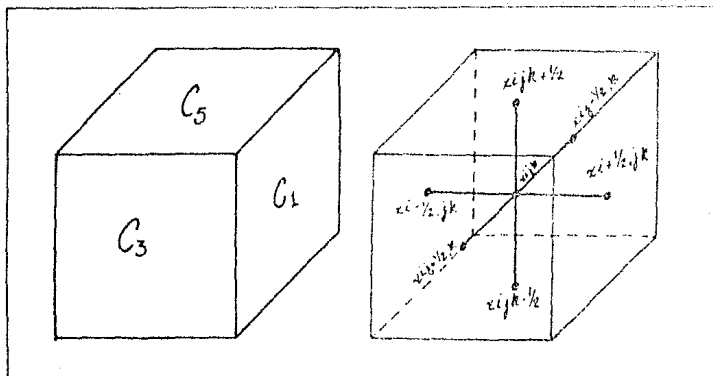


FIGURA 4. Notación utilizada para el método de células.

Por otro lado, si integramos la segunda parcial de la concentración con respecto a una de las variables, por ejemplo x_1 , tenemos

$$\int_{E_{ijk}} \frac{\partial^2 c}{\partial x_1^2} dx = \int_{E_{jk}} \frac{\partial c}{\partial x_1}(x_{i+1/2,j,k}) - \frac{\partial c}{\partial x_1}(x_{i-1/2,j,k}) dx \quad (4.2.4)$$

$$\approx h(c_{i+1,j,k} + c_{i-1,j,k} - 2c_{i,j,k}),$$

si aproximamos la $\frac{\partial c}{\partial x_1}(x_{i+1/2,j,k})$ por medio de la pendiente de la cuerda que une las concentraciones en los puntos $x_{i+1,j,k}$ con $x_{i,j,k}$ y análogamente para $\frac{\partial c}{\partial x_1}(x_{i-1/2,j,k})$. Si unimos los resultados (4.2.3) y (4.2.4) para cada una de las variables x_1 , x_2 y x_3 obtenemos la ecuación

$$\int_{E_{ijk}} \left(\nabla \cdot \mathbf{vc} - \sum_{i=1}^3 \frac{\partial^2 c}{\partial x_i^2} \right) dx \approx -h^2 \left(\frac{v}{2} + \frac{1}{h} \right) c_{i,j,k-1}$$

$$- h^2 \left(\frac{v}{2} + \frac{1}{h} \right) c_{i,j-1,k} - h^2 \left(\frac{v}{2} + \frac{1}{h} \right) c_{i-1,j,k}$$

$$+ 6hc_{i,j,k} + h^2 \left(\frac{v}{2} - \frac{1}{h} \right) c_{i+1,j,k} + h^2 \left(\frac{v}{2} - \frac{1}{h} \right) c_{i,j+1,k}$$

$$+ h^2 \left(\frac{v}{2} - \frac{1}{h} \right) c_{i,j,k+1} \quad (i, j, k = 1, \dots, N) \quad (4.2.5)$$

Hemos formulado nuestro problema en términos de un sistema de ecuaciones lineales, que podemos escribir por medio de una matriz heptadiagonal de dimensión N^3 . La matriz \mathbb{P} será

$$\mathbb{P} = \begin{pmatrix} \beta & \alpha & 0 & \dots & 0 & \alpha & 0 & \dots & 0 & \alpha & 0 & \dots \\ \gamma & \beta & \alpha & 0 & \dots & 0 & \alpha & 0 & \dots & 0 & \alpha & 0 & \dots \\ 0 & \gamma & \beta & \alpha & 0 & \dots & 0 & \alpha & 0 & \dots & 0 & \alpha & 0 & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \dots & 0 & \gamma & \beta & \alpha & 0 & \dots & 0 & \alpha & 0 & \dots & 0 & \alpha & 0 \\ \gamma & 0 & \dots & 0 & \gamma & \beta & \alpha & 0 & \dots & 0 & \alpha & 0 & \dots & 0 & \alpha & 0 \\ 0 & \gamma & 0 & \dots & 0 & \gamma & \beta & \alpha & 0 & \dots & 0 & \alpha & 0 & \dots & 0 & \alpha & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \dots & 0 & \gamma & 0 & \dots & 0 & \gamma & \beta & \alpha & 0 & \dots & 0 & \alpha & 0 & \dots & 0 & \alpha & 0 \\ \gamma & 0 & \dots & 0 & \gamma & 0 & \dots & 0 & \gamma & \beta & \alpha & 0 & \dots & 0 & \alpha & 0 & \dots & 0 & \alpha \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{pmatrix}$$

donde

$$\beta = \frac{6}{h}, \quad \alpha = \frac{v}{2} - \frac{1}{h}, \quad \gamma = -\frac{v}{2} - \frac{1}{h}, \quad v = \sqrt{\frac{Pe}{3h}}$$

4.3 Resultados numéricos

Damos los valores $h = 0.1$ y $u^0 = 0$ para todos los casos. Variamos la velocidad, cambiando el número de Peclet ($Pe=10,100,1000,10000$) como se hizo en el artículo de Celia *et al.* [3]. El criterio para terminar los algoritmos es $\|r^k\|_2 \leq 10^{-4}$. Todos los experimentos se realizaron en una microcomputadora Tandy de 8 MHz, utilizando doble precisión. Los programas están escritos en FORTRAN.

La siguiente Tabla resume el trabajo (número de operaciones en cada iteración) y memoria requeridos por cada método.

	MÉTODO NUEVO	ORTHOMIN(3)	GCR(3)	EXT NAT
RAM (bytes)	120 N^3	136 N^3	136 N^3	120 N^3
No. it.	20 N^3	69 N^3	69 N^3	55 N^3

PRIMERA PRUEBA

$Pe = 10$, $N = 3, 4, 5, 6, 7, 8$. Contamos iteraciones y tiempo de CPU.

En la primera prueba se varió la dimensión de la matriz para obtener una gráfica que muestra como crece el número de iteraciones con respecto a la dimensión (Figs. 5 y 6). Por otro lado, se midió el tiempo que tarda cada algoritmo en encontrar la aproximación (Figs. 7 y 8). Otra forma de medir el tiempo, independiente de la computadora utilizada, consiste en medir el número de operaciones (multiplicaciones) para cada algoritmo (Fig. 9).

SEGUNDA PRUEBA

$N = 8$, $Pe = 10, 100, 1000, 10000$. Contamos iteraciones y tiempo de CPU.

La segunda prueba consistió en trabajar con matrices de dimensión fija y variar el número de Peclet, de tal forma que las matrices obtenidas tuviesen asimetrías distintas, esto es, que $\gamma - \alpha$ tenga distintas magnitudes. Nuevamente medimos iteraciones, tiempo de CPU y operaciones (Figs. 10-12).

TERCERA PRUEBA

$N = 8$, $Pe = 10$. Medimos $\|r^k\|_2^2$ en la k -ésima iteración.

En la tercera prueba dejamos fija $N = 8$, $Pe = 10$ y medimos la magnitud del residuo $\|r^k\|_2$ a la k -ésima iteración. Graficamos operaciones contra $\|r^k\|_2^2$ (Figs. 13-17) y, calculando el error utilizando alguna de las aproximaciones obtenidas, operaciones contra $\|e\|_2^2$ (Fig. 18 y 19).

4.4 Discusión y conclusiones

Las Tablas 2-7 presentan los valores graficados en las Figuras 5-19.

De la gráfica 5 se obtiene que $I \approx 4.2N$ para ORTHOMIN(3) y GCR(3), mientras que de la gráfica 6 obtenemos $I \approx KN^{1.8}$ (K constante) para NM y EN. Entonces, ORTHOMIN(3) y GCR(3) requieren menos iteraciones para obtener una aproximación cuyo residuo $\|r^n\|_2$ cumpla la condición $\|r^n\|_2^2 \leq 10^{-1}$, que NM y EN, sin embargo, el tiempo y trabajo que utiliza NM es siempre menor de la mitad que el utilizado por cualquiera de los otros métodos (Figs. 7-9). Esto se debe al número de operaciones que realiza cada método en una iteración (ver Tabla 1). Al graficar el logaritmo natural de las iteraciones contra el del trabajo (Fig. 8) obtenemos cuatro curvas que se pueden considerar rectas con pendiente aproximada de 4.7, por lo que, $T \approx KN^{0.7}$ para los cuatro métodos, y el valor más pequeño lo toma K en el método nuevo.

Al variar el número de Peclet, la cantidad de iteraciones que requiere cada método para terminar también varía (Fig. 10), sin embargo, mientras que NM y EN tienen una variación máxima de alrededor de quince iteraciones, los métodos GCR(3) y ORTHOMIN(3) la tienen de alrededor de noventa iteraciones, esto indica que NM y EN son más estables con respecto a asimetrías que GCR(3) y ORTHOMIN(3). Esto se refleja también en el tiempo utilizado y el trabajo realizado por cada algoritmo

N	NM	ORTHOMIN(3)	GCR(3)	EN
	I	I	I	I
3	10	8	7	10
4	16	13	13	19
5	24	17	17	28
6	33	21	22	46
7	44	26	26	53
8	56	30	30	66

TABLA 2. Valores de N contra iteraciones para cada método. Ver Figura 5.

ln N	NM	ORTHOMIN(3)	GCR(3)	EN
	ln I	ln I	ln I	ln I
1.0286	2.3026	2.0794	1.9459	2.3026
1.3863	2.7726	2.5649	2.5649	2.9444
1.6091	3.1780	2.8332	2.8332	3.3322
1.7917	3.4905	3.0445	3.0910	3.6889
1.9459	3.7842	3.2581	3.2581	3.9703
2.0794	4.0253	3.4012	3.4012	4.1896

TABLA 3. Valores de ln N contra ln I para cada uno de los métodos. Ver Figura 6.

para terminar el proceso (Figs. 11, 12). El tiempo utilizado en el caso del NM nunca es mayor a veinte minutos y en cualquiera de los otros métodos nunca es menor de treinta y cinco minutos. GCR(3) y ORTHOMIN(3) llegan a tardar alrededor de dos horas cuando el número de Péclet vale 10000.

Se graficaron juntos los datos obtenidos para NM y ORTHOMIN(3) (Figs. 13-15), GCR(3) no se graficó porque su tabla de resultados es muy parecida a la de ORTHOMIN(3). La gráfica de EN (Figs. 16-17) se hizo aparte debido a que las magnitudes de las normas de sus residuos son muy disímiles. En la gráfica 13 podemos observar la diferencia que existe entre el descenso de la norma del residuo en los métodos GCR(3) y NM, ya que $\|r^n\|_2$ decrece monótonamente en el caso de GCR(3), cosa que no sucede para NM, más aún, la norma del residuo en este método da saltos abruptos. A pesar de esto y de que en un inicio la norma del residuo de NM es mucho mayor que la de GCR(3), conforme el número de operaciones aumenta, la norma del NM disminuye más rápidamente que la de ORTHOMIN(3) (Figs. 14, 15). Este comportamiento tan distinto se debe a la diferencia estructural de los métodos: GCR minimiza a $\|r^n\|_2$, por lo que está obligada a decrecer monótonamente y NM minimiza a $\|e^n\|_2$ por lo que $\|r^n\|_2$ puede tener un comportamiento distinto. Lo mismo se observa en el caso de EN (Figs. 16-17), ya que este método minimiza el residuo respecto a la norma $\| \cdot \|_{A^{-1}}$ y $\|r^n\|$ puede tener también saltos (aunque en este caso no se presentan tan marcados como en NM). Al graficar operaciones contra $\|e^n\|_2^2$ se confirma que la norma del error en NM (Figs. 18-19) decrece monótonamente y siempre es más pequeña que la del error de ORTHOMIN(3).

Los resultados obtenidos muestran al nuevo método como un método competitivo e inclusive superior a ORTHOMIN(k), GCR(k) y por supuesto a la extensión

N	NM T	ORTHOMIN(3) T	GCR(3) T	EN T
3	0:11	0:21	0:17	0:20
4	0:36	1:26	1:23	1:31
5	1:44	3:47	3:37	4:29
6	4:08	8:18	8:21	10:15
7	8:48	16:39	15:50	23:57
8	16:45	29:01	27:44	45:01

TABLA 4. Valores de N contra tiempo para cada uno de los métodos. Ver Figura 7.

$\ln N$	NM $\ln T$	ORTHOMIN(3) $\ln T$	GCR(3) $\ln T$	EN $\ln T$
1.0986	-1.7148	-1.0498	-1.2730	-1.1087
1.3863	-0.5108	0.3577	0.3221	0.4187
1.6094	0.5481	1.3297	1.2865	1.4996
1.7917	1.4183	2.1162	2.1223	2.3273
1.9459	2.1747	2.8124	2.7619	3.1760
2.0794	2.8184	3.368	3.3225	3.8071

TABLA 5. Valores de $\ln N$ contra $\ln T$ para cada uno de los métodos. Ver Figura 8.

natural del gradiente conjugado. Además, como se mencionó anteriormente, NM es más adecuado para problemas no simétricos debido a su estabilidad ante cambios en la asimetría de las matrices. Sin embargo no se puede aún establecer una conclusión definitiva al respecto porque es necesario analizar un número más amplio de casos y también introducir preconditionadores, ya que esta es la forma que con mayor frecuencia se utiliza los métodos comparados en esta tesis. Por otro lado, falta incluir comparaciones con el método GMRES de Saad y Schultz [13]. Sin embargo, como se muestra en el trabajo de Susana Gómez y José Luis Morales [7], tiene eficiencia muy cercana a ORTHOMIN(k) y GCR(k).

I	NM	ORTHOMR(5)	CCR(3)	EN
1	40787.9738077400000	11366.8008132930000	11376.8008132930000	14270.6332177000000
2	80324.2242747400000	5739.3184474000000	5739.3184474000000	93655457.45242130000
3	68114.5002875000000	3484.7836481620000	3484.7836481620000	7531464.3847253000000
4	51095.8927597440000	2384.3159237520000	2389.3156237520000	2624167.4541460000000
5	48124.5051919140000	1721.1194435664000	1722.4953575810000	4152029.4598304600000
6	44477.7307976300000	1278.6199560370000	1282.4427896817000	2919018.4217480000000
7	39604.5926542500000	956.1703643530000	956.3620019721000	2493189.1822997000000
8	43541.8511509730000	713.9082746952000	713.5147674633000	2354455.8654240000000
9	39829.8444444440000	530.5333745184800	530.3938484906000	1584285.3254054000000
10	30734.5610084100000	366.2816615970000	366.2547942900000	1740034.9199136000000
11	30387.6343115330000	243.1455091316400	243.1099937525000	1000252.1101012600000
12	42991.3793279560000	147.8664407145600	147.4452227523000	1012761.2603500000000
13	35580.8411943240000	79.21106671613100	79.17345579971800	616841.6684304000000
14	34840.8946445200000	37.84012824747200	37.51566866627000	724548.5849561300000
15	37751.4408484900000	18.36442197636000	18.19457844277400	608400.3776741000000
16	34794.3483402300000	7.04292294363000	6.9422113019000	593816.3832600000000
17	26242.3104261500000	3.34400840422740	3.2796186533100	473221.4352179200000
18	23971.4264495110000	1.96801140119570	1.91211765113360	515306.3494242700000
19	21430.1094958420000	1.38841181127510	1.33745148209480	570257.1044309650000
20	11226.6532242730000	W 7638190094702E-001	6.4527467015259E-001	761241.5046952700000
21	7614.3757603545000	5.4389831798224E-001	5.2679335474812E-001	760853.6373414100000
22	7850.3870001760000	2.1184976492253E-001	2.0419060108577E-001	444323.3283623700000
23	6648.4544924610000	7.5862518206183E-002	7.1057262811729E-002	314867.0166414000000
24	9654.7107751120000	3.6273240406011E-002	3.8707066040028E-002	321010.77060000000
25	14710.1180072040000	1.3703592765247E-002	1.1823065115187E-002	162845.8300863100000
26	7784.5873307020000	7.2846651026431E-003	6.9224406550643E-003	126117.5019061409000
27	4167.4832418719000	6.1858592018742E-003	6.0103687622403E-003	229548.8443622200000
28	2564.7674710314000	3.2127124985400E-003	3.1320907483013E-003	309058.8068073000000
29	1956.0555347830000	1.886610269842E-003	1.4120134322413E-003	250348.1631851400000
30	1348.2307563536000	7.937529365932E-004	7.2022466975949E-004	196574.5677514000000
31	628.7811214772477			273901.1153180000000
32	472.3548195136200			114795.5444636000000
33	267.4849076742300			89633.2665933100000
34	270.2198428379300			64066.0301646100000
35	192.3102902867300			34821.4134733500000
36	180.02201564042200			20001.9023351800000
37	188.0645931930000			19650.8284404700000
38	92.9310849800000			22271.1558942300000
39	41.2624234249600			14051.0204506500000
40	26.2022573871500			1574.40154153200000
41	7.1972546970650			8510.7136442280000
42	3.20737486614050			4251.5143846416000
43	8.1959618097949E-001			870.96357128052000
44	3.1847610517142E-001			262.92249717830000
45	2.1942360493944E-001			96.6364734903600
46	2.362448218132E-001			8.856666964531E-001
47	9.403933183019E-002			87.10758764752000
48	4.4389638714741E-002			56.27919154092000
49	3.3671600574927E-002			23.94492016958000
50	2.0193259465234E-002			12.00180448486600
51	8.1173031263230E-003			6.6153431107720
52	4.194701930793E-003			4.76031436551310
53	3.388415974593E-003			2.0873849697249
54	2.838451344183E-003			1.01680008682210
55	2.7258842491371E-003			7.6300179243028E-001
56	7.3421501271362E-004			4.9147167694587E-001
57				2.5807293249180E-001
58				2.0064131767440E-001
59				6.2477810618077E-001
60				3.9399351384177E-001
61				7.407626103202E-001
62				1.9082978622122E-001
63				8.4746602613052E-001
64				5.720624512665E-001
65				2.1357458431616E-001
66				5.438630353154E-001

TABLA 6. Número de iteraciones I contra $\|r^I\|_2$ para cada uno de los métodos. Ver Figuras 15-17.

<i>I</i>	NM	ORTHOMIN(3)	EN
1	274.47413766800000	233.45377606710000	249.03040571320000
2	247.14780419000000	174.02145348440000	244.50226582930000
3	225.03263465760000	133.01675557320000	253.22711141050000
4	207.20161592270000	100.21279333700000	237.74094870300000
5	191.34357466342000	73.96237121827000	223.79120514510000
6	178.32369493310000	52.59424531334000	210.55616326200000
7	160.91814519300000	35.46596919320000	187.29937775822000
8	147.69480003940000	22.403784014074000	166.43375602633000
9	136.27563655060000	12.974620492427000	173.74029583544000
10	123.49185030349000	6.743251307037300	163.24532300000000
11	114.66530459041000	3.021870016847100	154.83277048793000
12	101.92558328540006	1.119712604119350	147.03135546242000
13	91.250275320985000	3.80028794377850E-001	137.97402600785000
14	80.110127396125000	1.5792355468119E-001	120.12203140290000
15	69.474574443423000	1.0846043918768E-001	121.23083002648000
16	59.72623630560000	6.73662253600390E-002	112.95572344566000
17	49.981655910563500	7.05408194896050E-002	102.95034334080000
18	39.071953041673600	5.8154324942290E-002	92.73216841818000
19	31.719121208413800	4.1638457714100E-002	82.67076943306000
20	25.739987463303100	2.8062101074194E-002	67.70716722865000
21	22.436764554242000	8.1335597429959E-003	53.402649446346000
22	20.196416146988000	1.52941160264304E-003	44.40357668480000
23	17.353664477047000	3.4532033197283E-004	37.44095577452000
24	14.436454129168000	3.3189692525593E-004	33.47916441448000
25	10.829906017504000	3.1917216100985E-004	30.92094603028200
26	6.612082823416200	2.3684623782720E-004	28.51682794924100
27	4.048876414700500	1.7073780290719E-004	20.21773624554200
28	2.648140072163600	8.1406447272388E-005	13.843106760305000
29	1.855040055623300	2.8187248511548E-005	6.498345114000000
30	1.269820914272600	4.0131851823334E-006	5.277309455800800
31	8.1768067891946E-001		3.091959367334480
32	6.4335713727267E-001		1.844991145263900
33	4.7414929409792E-001		1.188662507466640
34	3.5773870184008E-001		7.8544746132998E-001
35	2.7320961195752E-001		3.4524441354712E-001
36	1.90981164474761E-001		4.06477603004971E-001
37	1.0680395671405E-001		2.9108027938147E-001
38	2.00070774701160E-002		1.8200937497674E-001
39	2.066336596092E-002		9.096401785411E-002
40	9.9614349353783E-003		4.2971787010459E-002
41	3.4974796676190E-003		1.0464901408079E-002
42	1.3819416400022E-003		3.0653279785474E-002
43	6.8088197642624E-004		7.4210490138693E-004
44	3.9589691493127E-004		4.5066151487580E-004
45	2.7020416972127E-004		3.3900422996723E-004
46	1.5442144319785E-004		2.4095457726414E-004
47	7.1252520042673E-005		1.2823301713314E-004
48	2.8732721591609E-005		7.1279420979764E-005
49	1.3702318048364E-005		3.8969288716906E-005
50	1.113253608789E-005		1.8781270153925E-005
51	8.4498725710081E-006		9.7430824013635E-006
52	3.7264184364384E-006		6.7235402213182E-006
53	2.1777567418031E-006		4.0679460126376E-006
54	1.1412063765040E-006		2.6504502213264E-006
55	4.7603264633696E-007		1.4726106404653E-006
56			5.9744737370122E-007
57			1.3710071732354E-007
58			1.84370115300077E-007
59			2.3652220645758E-007
60			3.0111527124877E-007
61			3.8071298501690E-007
62			3.8484673290224E-007
63			4.2241099200404E-007
64			4.8048031530851E-007
65			4.7059391778512E-007
66			4.7722275649218E-007

TABLA 7. Número de iteraciones *I* contra $\|e^I\|_2^2$ para NM, ORTHOMIN(3) y EN. Ver Figuras 18 y 19.

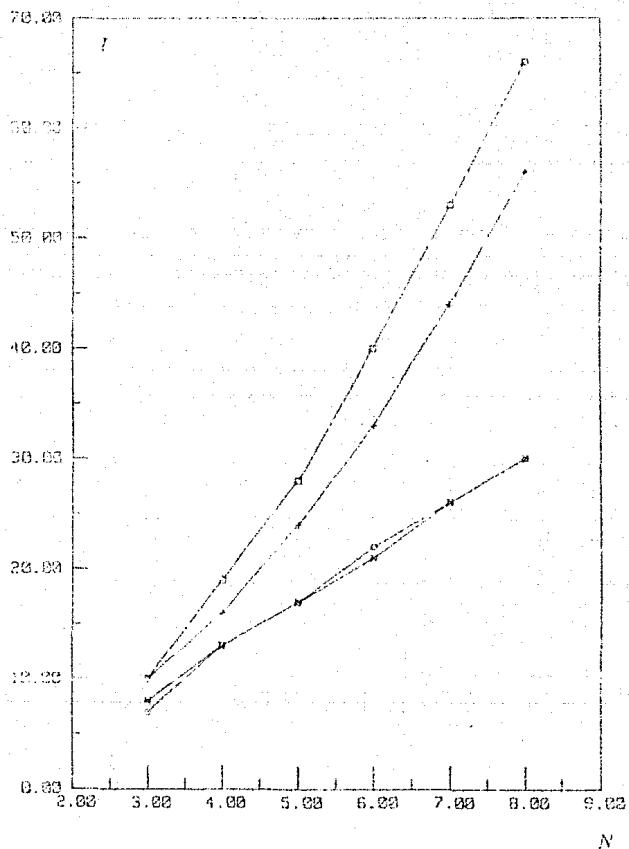


FIGURA 5. Gráfica de N contra I (iteraciones) para los cuatro métodos, donde N^3 es la dimensión de la matriz. (+): NM; (*): ORTHOMIN(3); (O): GCR(3); (□): EN

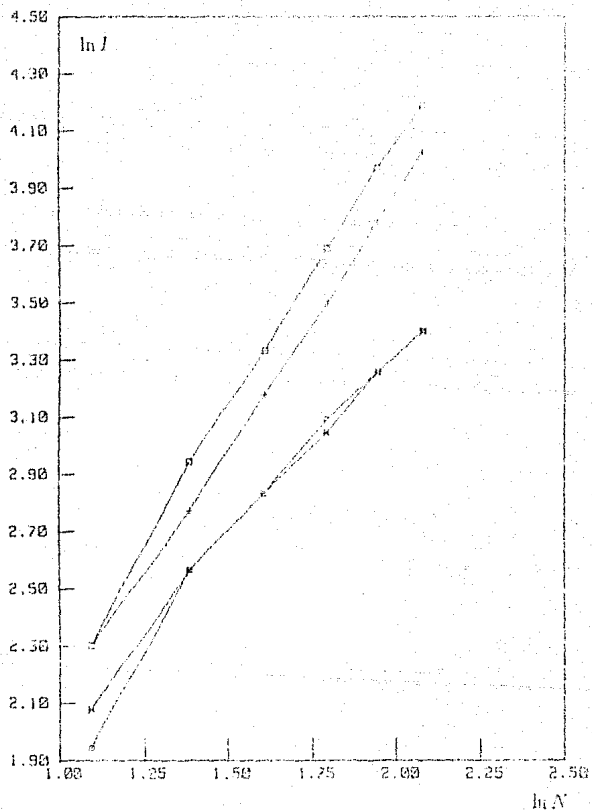


FIGURA 6. Gráfica de $\ln J$ contra $\ln N$ para los cuatro métodos, donde N^3 es la dimensión de la matriz. (+: NM; •: ORTHOMIN(3); ○: GCR(3); □: EN)

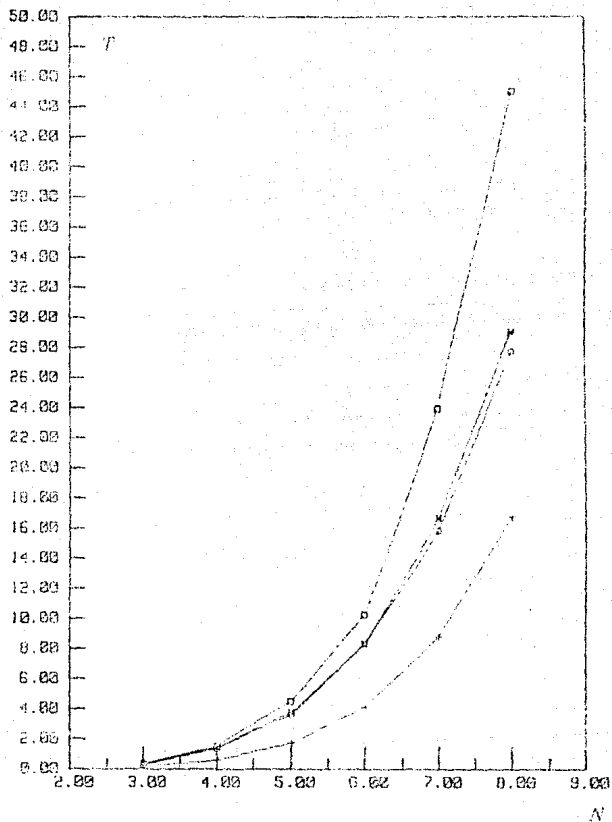


FIGURA 7. Gráfica de N contra T (tiempo) para los cuatro métodos, donde N^d es la dimensión de la matriz. (+: NM; *: ORTHOMIN(3); ○: GCR(3); □: EN)

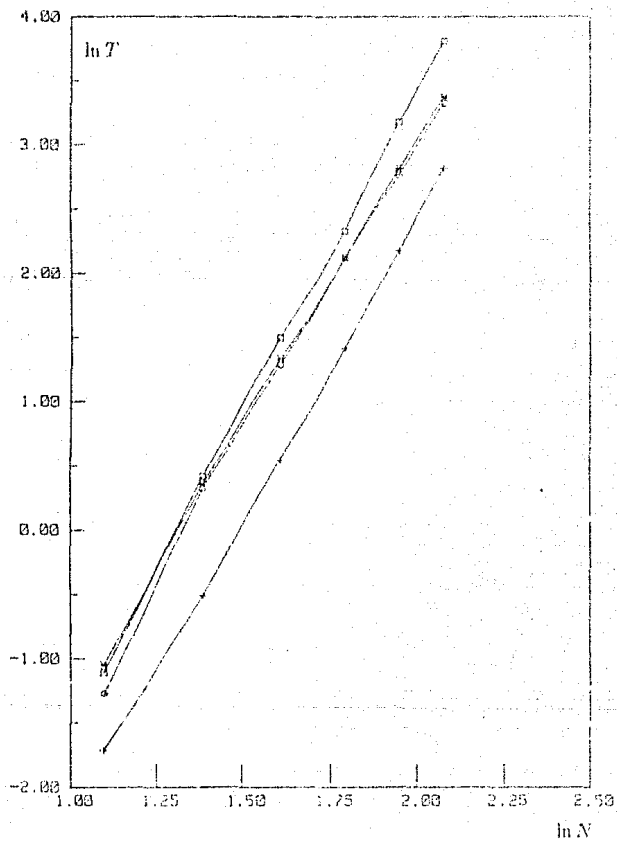


FIGURA 8. Gráfica de $\ln N$ contra $\ln T$ para los cuatro métodos, donde N^3 es la dimensión de la matriz. (4: SM; *: ORTHOMIS(3); O: GCR(3); □: ES)

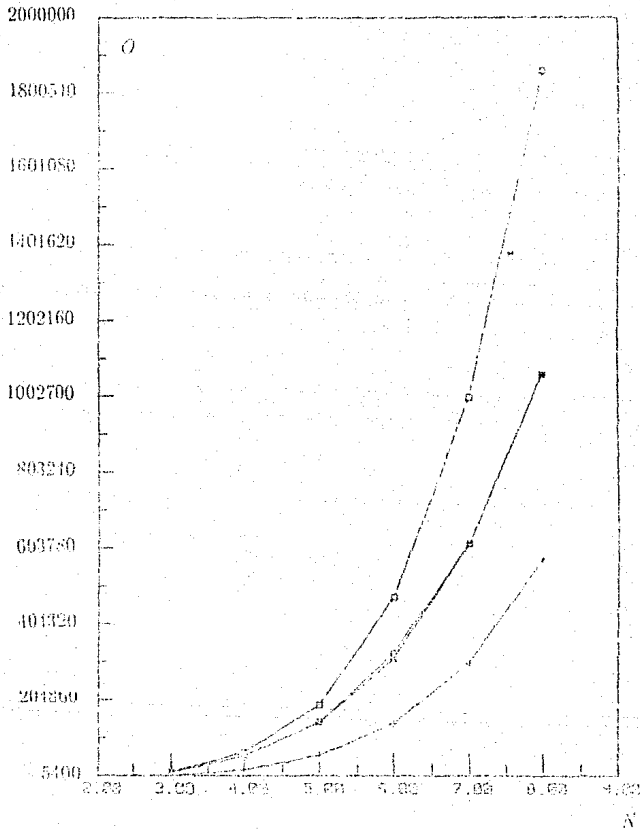


FIGURA 9. Gráfica de N contra O (número de operaciones) para los cuatro métodos. (+: NM; * : ORTHOMIS(3); ○: GCR(3); □: ES)

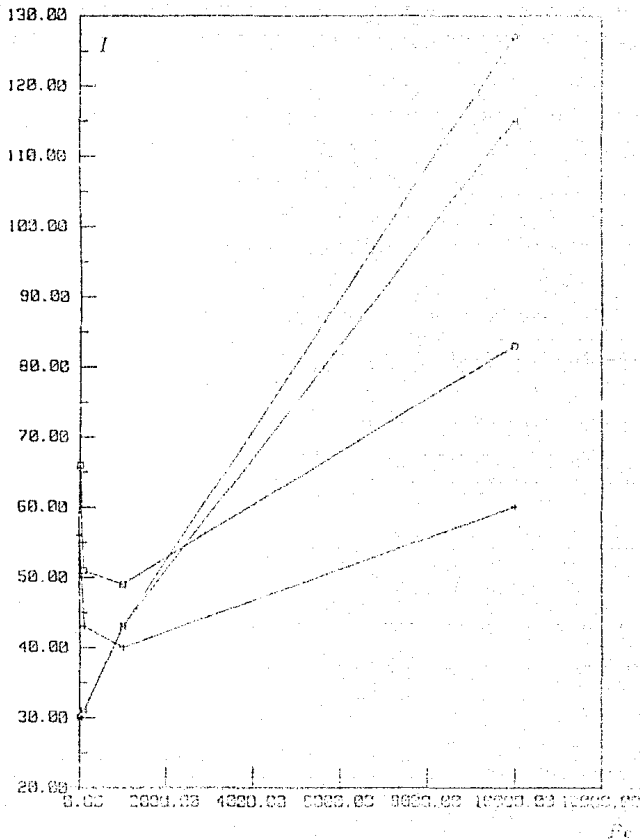


FIGURA 10. Gráfica de Pz (número de Peckel) contra I (iteraciones) para los cuatro métodos.
 (\triangle : NM; ∇ : ORTHOMIN(3); \circ : GCR(3); \square : EN)

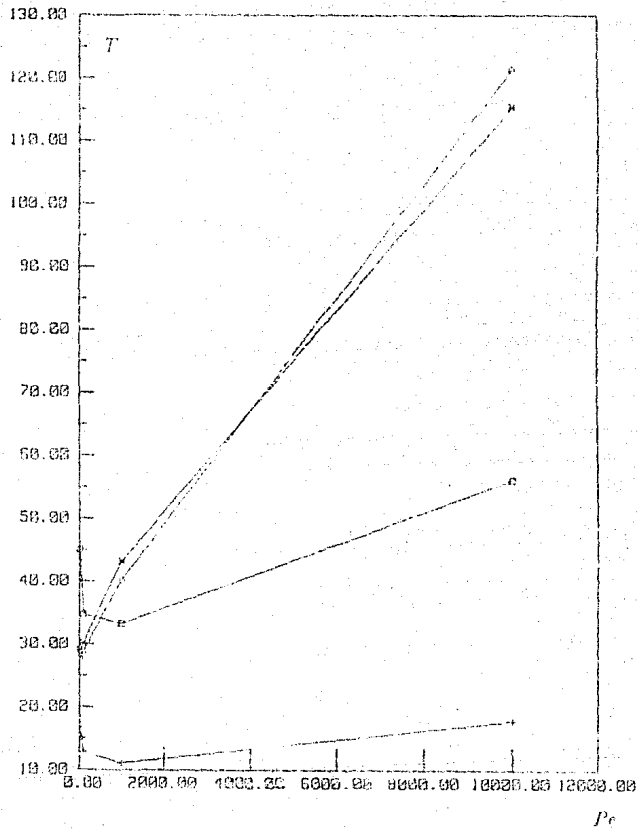


FIGURA 11. Gráfica de Pe (número de Peclet) contra T (tiempo) para los cuatro métodos. (+: NM; *: ORTHOMIS(3); O: GCR(3); □: EX)

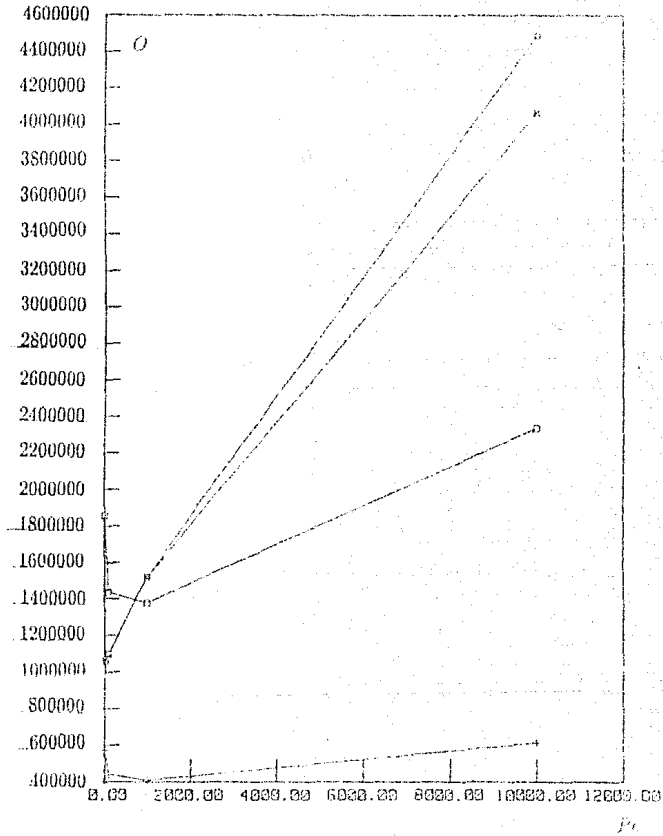


FIGURA 12. Gráfica de Pe (número de Peclet) contra O (número de operaciones) para los cuatro métodos. (+: NM; *: ORTHOMIN(3); O: GER(3); □: EX)

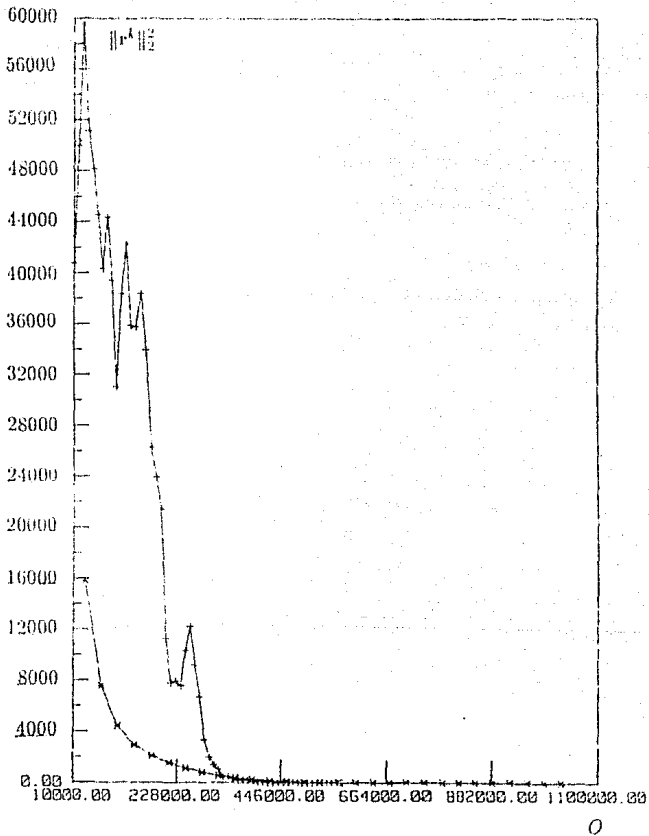


FIGURA 13. Gráfica de O (número de operaciones a la k -ésima iteración) contra $\|r^k\|_2^2$ (norma del residuo al cuadrado) para SM y ORTHOMIN(3). (+: SM; *: ORTHOMIN(3))

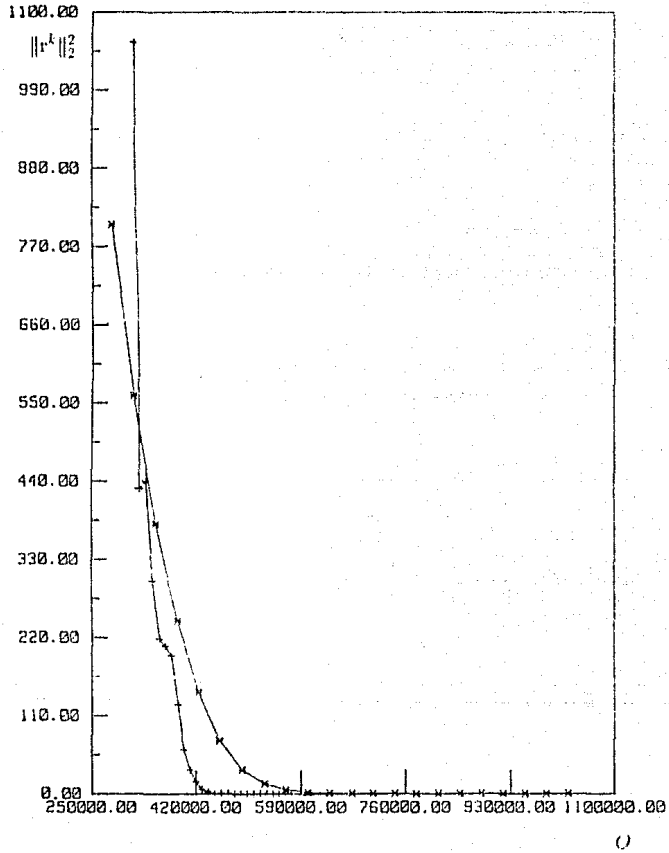


FIGURA 14. Acercamiento de la Figura 13.

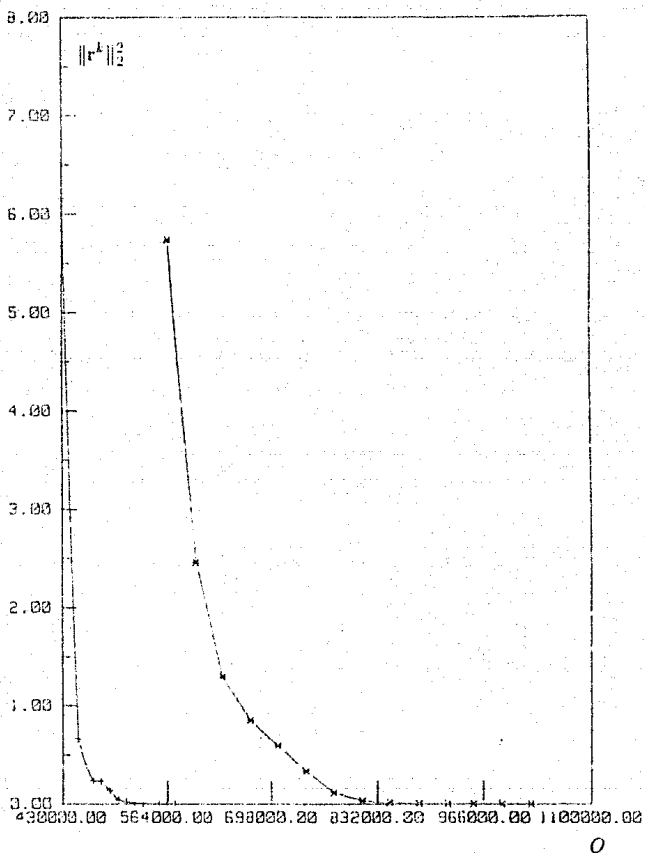


FIGURA 15. Acercamiento de la Figura 13.

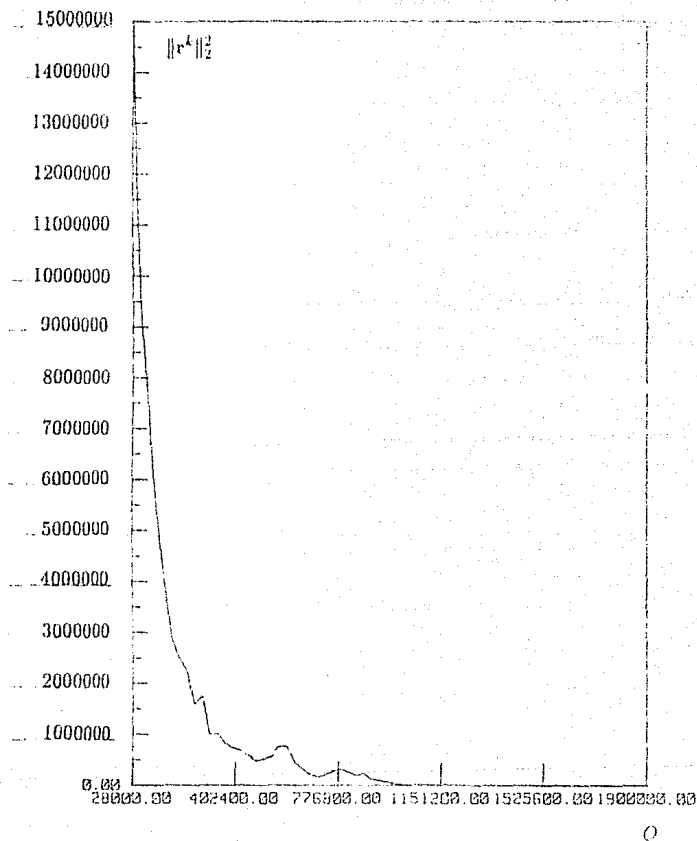


FIGURA 16. Gráfica de O (número de operaciones a la k -ésima iteración) contra $\|r^k\|_2^2$ (norma del residuo al cuadrado) para EN.

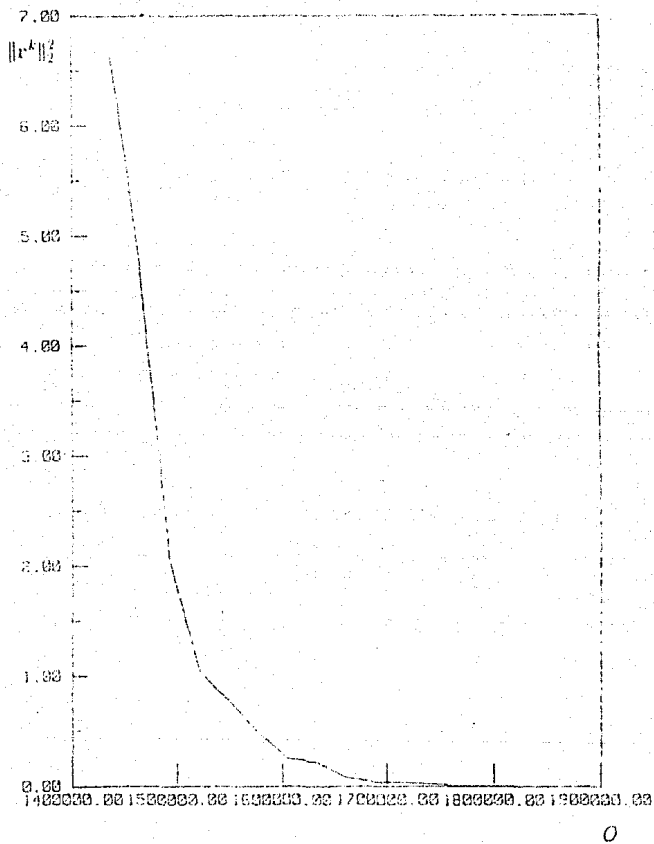


FIGURA 17. Acercamiento de la Figura 16.

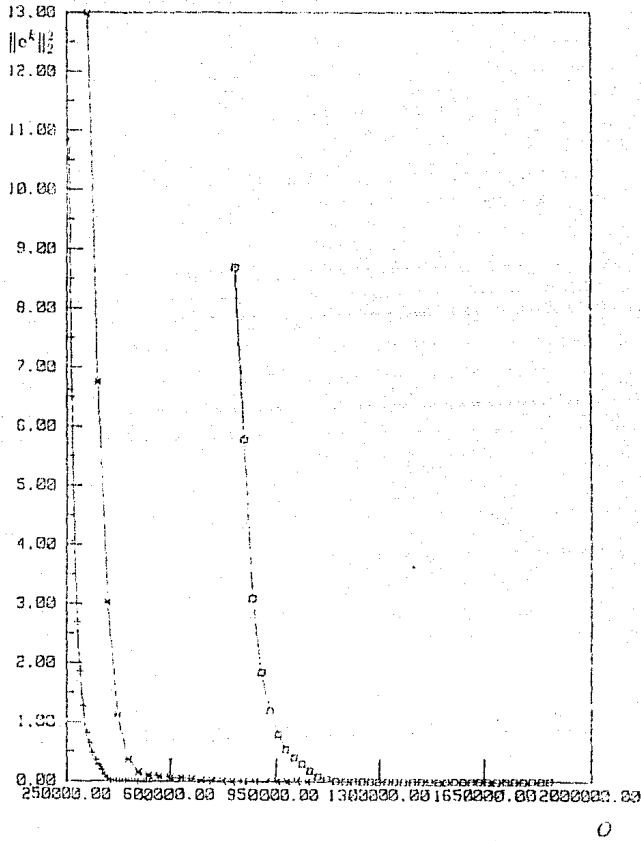


FIGURA 19. Acercamiento de la Figura 18.

Apéndice: programas

Los programas son versiones originales, tomando como referencia un programa desarrollado por el Dr. Robert Yates para el nuevo método. Están escritos en FORTRAN, utilizando las instrucciones básicas, a sugerencia del mismo Dr. Yates.

```
C.....
C.....
C**
C**          PROGRAMA PROPGCHE.FOR
C**
C**
C.....
C.....
C RESUELVE EL PROBLEMA AU=B, CON A UNA MATRIZ NO SIMETRICA.
C UTILIZANDO EL METODO TRADICIONAL DE GRADIENTE CONJUGADO PARA
C EL SISTEMA AT*AU=AT*B, PERO UTILIZANDO EL PRODUCTO ESCALAR USUAL
C DE LOS REALES. LA MATRIZ A DEBE SER HEPTADIAGONAL CON ELEMENTOS
C IGUALES EN CADA RENGLON.
C IMPLICIT REAL*8 (A-N,O-Z)
C DIMENSION A1(30000)
C OPEN(UNIT=1,FILE='MATRIZ.DAT',STATUS='OLD',FORM='BINARY')
C OPEN(UNIT=2,FILE='VECTOR.DAT',STATUS='OLD',FORM='BINARY')
C OPEN(UNIT=3,FILE='PIOPR.OP',STATUS='NEW')
C WRITE(*,*) 'CUAL ES LA DIMENSION X,Y,Z'
C READ(*,*) II,JJ,KK
C IJ=II*JJ
C NP=IJ*KK
C IULT=1
C IA=IULT
C IU=IA+NP*7
C IB=IU+NP
C IW=IB+NP
C IR=IW+NP
C IV=IR+NP
C IP=IV+NP
C IS=IP+NP
C IULT=IS+NP
C IF(IULT.GT.30000) CALL EXIT
C CALL GRAL(IJ,IJ,NP,A1(IA),A1(IU),A1(IB),A1(IW),A1(IR),A1(IV),
C * A1(IP),A1(IS))
C CLOSE(UNIT=1)
C CLOSE(UNIT=2)
C CLOSE(UNIT=3)
C END

C
C .....
C LEE LAS ENTRADAS DE LA MATRIZ A DEL ARCHIVO 1 Y LAS DE B
C DEL ARCHIVO 2. DEFINE U=0 COMO PRIMERA APROXIMACION. LLAMA
C A LA SUBROUTINA
C SUBROUTINE GRAL(IJ,IJ,NP,A,U,B,W,R,V,P,S)
C IMPLICIT REAL*8 (A-N,O-Z)
```

```

DIMENSION A(NP,7),U(NP),B(NP),W(NP),K(NP),F(NP),P(NP),
* S(NP)
READ(2) (B(I),I=1,NP)
READ(1) ((A(I,J), J=1,7),I=1,NP)
DO 200 I=1,NP
200 U(I)=0.DO
CALL ASYM(NP,IJ,II,A,U,B,R,P,V,W,S)
RETURN
END
.....
C DESARROLLA EL ALGORITMO DE GRADIENTE CONJUGADO MODIFICADO.
C DEFINIMOS U COMO LA N-ESIMA APROXIMACION, R SU RESIDUO, P LA
C DIRECCION DE BUSQUEDA Y Q1 LA NORMA DE R. EL FIN DE LA SUBROUTINA
C SE DA DESDE DE LA PANTALLA.
SUBROUTINE ASYM(NP,IJ,II,A,U,B,R,P,V,W,S)
IMPLICIT REAL*8 (A-H,G-Z)
DIMENSION A(NP,7),U(NP),B(NP),R(NP),P(NP),W(NP),V(NP),S(NP)
READ(*,*)
N=0
10 CALL HUL3D(NP,IJ,II,A,U,W)
DO 20 I=1,NP
K(I)=B(I)-W(I)
P(I)=0.DO
20 CONTINUE
CALL HUL3DA(NP,IJ,II,A,R,S)
BETA=0.DO
30 DO 40 I=1,NP
V(I)=P(I)
P(I)=S(I)-BETA*P(I)
40 CONTINUE
ALPHA=0.DO
D=0.DO
DO 50 I=1,NP
ALPHA=ALPHA+R(I)*R(I)
D=D+P(I)*P(I)
50 CONTINUE
ALPHA=ALPHA/D
DO 60 I=1,NP
U(I)=U(I)+ALPHA*P(I)
60 CONTINUE
CALL HUL3D(NP,IJ,II,A,U,W)
Q1=0.DO
DO 70 I=1,NP
R(I)=B(I)-W(I)
Q1=Q1+R(I)**2
70 CONTINUE
CALL HUL3DA(NP,IJ,II,A,R,S)
N=N+1
WRITE(*,*)'RESID',N,Q1
WRITE(S,*) N*20*NP,Q1
IF((N.GE.200).OR.(Q1.LEQ.10E-04)) RETURN
BETA=0.DO
DO 80 I=1,NP
BETA=BETA+S(I)*P(I)
80 CONTINUE

```

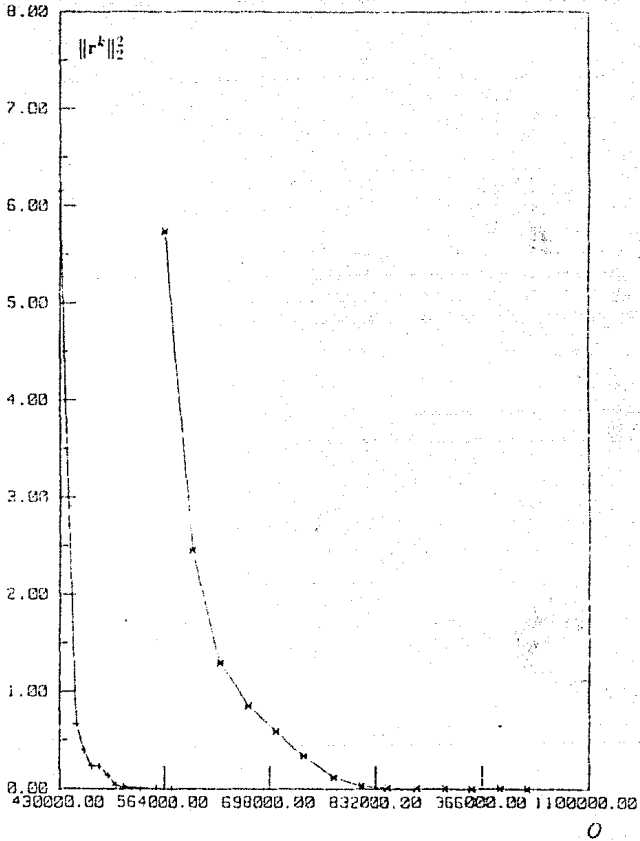


FIGURA 15. Acercamiento de la Figura 13.

```

CALL ORTHOMIN(NP,IJ,II,K,A,U,B,R,P,V,S,AP,V1)
RETURN
END
C*****
C  DEFINIMOS U COMO LA N-ESIMA APROXIMACION, R SU RESIDUO, P LA
C  MATRIZ QUE GUARDA A LA BASE Y Q1 LA FORMA DE R. EL FIN DE LA
C  SUBROUTINA SE DA DESDE DE LA PANTALLA. LLAMA A LA SUBROUTINA
C  BASE CADA VEZ QUE CONSTRUYE UNA NUEVA APROXIMACION.
SUBROUTINE ORTHOMIN(NP,IJ,II,K,A,U,B,R,P,V,S,AP,V1)
IMPLICIT REAL*8 (A-H,Q-Z)
DIMENSION A(NP,7),U(NP),B(NP),R(NP),P(K,NP),V(NP),S(NP),
* AP(NP),V1(NP)
READ(*,*)
N=0
10  CALL MUL3D(NP,IJ,II,A,U,V)
DO 20 I=1,NP
R(I)=B(1)-V(I)
P(K,I)=R(I)
DO 25 J=1,K-1
P(J,I)=0.DO
25  CONTINUE
20  CONTINUE
30  DO 40 I=1,NP
V(I)=P(K,I)
40  CONTINUE
ALPHA=0.DO
D=0.DO
CALL MUL3D(NP,IJ,II,A,V,AP)
DO 50 I=1,NP
ALPHA=ALPHA+R(I)*AP(I)
V=D*AP(I)+AP(I)
50  CONTINUE
ALPHA=ALPHA/D
DO 60 I=1,NP
U(I)=U(I)+ALPHA*P(K,I)
60  CONTINUE
Q1=0.DO
DO 70 I=1,NP
R(I)=R(I)-ALPHA*AP(I)
Q1=Q1+R(I)**2
70  CONTINUE
N=N+1
WRITE(3,*) N,Q1
WRITE(*,*)'RESID',N,Q1
IF((N.EQ.200).OR.(Q1.LE.10E-04)) RETURN
CALL BASE(NP,IJ,II,K,A,R,S,AP,P,V,V1)
GOTO 30
END

```

```

C.....
C.....
C**
C**          PROGRAMA GCRK.FOR          **
C**
C.....
C.....
C RESUELVE EL PROBLEMA  $AU=B$ , CON A UNA MATRIZ NO SIMETRICA,
C UTILIZANDO EL METODO TRADICIONAL DE RESIDUO CONJUGADO GENERALIZADO R
C (GCR(K)). K ES EL NUMERO DE ELEMENTOS DE LA BASE A UTILIZAR. LA
C MATRIZ A DEBE SER EPTADIAGONAL CON ELEMENTOS IGUALES EN CADA BENGLOS.
IMPLICIT REAL*8 (A-H,O-Z)
DIMENSION A1(30000)
OPEN(UNIT=1,FILE='MATRIZ.DAT',STATUS='OLD',FORM='BINARY')
OPEN(UNIT=2,FILE='VECTOR.DAT',STATUS='OLD',FORM='BINARY')
OPEN(UNIT=3,FILE='P10000GR.DAT',STATUS='NEW')
WRITE(*,*) 'CUALES SON LAS DIMENSIONES X,Y,Z?'
READ(*,*) II,JJ,KK
WRITE(*,*) 'CUAL ES EL VALOR DE K?'
READ(*,*) K
IJ=II*JJ
NP=IJ*KK
IULT=1
IA=IULT
IU=IA+NP*7
IB=IU+NP
IR=IB+NP
IV=IR+NP
IP=IV+NP
IS=IP+NP*K
IAP=IS+NP
IV1=IAP+NP
IULT=IV1+NP
IF(IULT.LE.30000) GO TO 100
WRITE(*,*) 'NO CABE'
CALL EXIT
100 CALL GRAL(NP,IJ,II,K,A1(IA),A1(IU),A1(IB),A1(IR),A1(IV),
* A1(IP),A1(IS),A1(IAP),A1(IV1))
CLOSE(UNIT=1)
CLOSE(UNIT=2)
CLOSE(UNIT=3)
END
C.....
C LEE LAS ENTRADAS DE LA MATRIZ A DEL ARCHIVO 1 Y LAS DE B
C DEL ARCHIVO 2. DEFINE U=0 COMO PRIMERA APROXIMACION. LLAMA
C A LA SUBROUTINA GCRK.
SUBROUTINE GRAL(NP,IJ,II,K,A,U,B,R,V,P,S,AP,V1)
IMPLICIT REAL*8 (A-H,O-Z)
DIMENSION A(NP,7),U(NP),B(NP),R(NP),V(NP),P(K,NP),
* S(NP),AP(NP),V1(NP)
READ(2) (B(I),I=1,NP)
HEAD(1) ((A(I,J),J=1,7),I=1,NP)
DO 200 I=1,NP
200 U(I)=0.DO
WRITE(*,*) 'SOL'

```



```

CALL GCRK(NP,IJ,II,K,A,U,B,R,P,V,S,AP,V1)
RETURN
END
C.....
C  CADA K ITERACIONES (CONTADAS POR LA VARIABLE IC) TODOS LOS ELEMENTOS
C  DE LA BASE SE HACEN CERO EXCEPTO EL K-ESIMO QUE PASA A SER EL
C  PRIMERO. DEFINIMOS U COMO LA N-ESIMA APROXIMACION, R SU RESIDUO,
C  P LA DIRECCION DE BUSQUEDA Y Q1 LA NORMA DE R. EL FIN DE LA
C  SUBROUTINA SE DA DESDE DE LA PANTALLA. LLAMA A LA SUBROUTINA BASE
C  CADA VEZ QUE CONSTRUYE UNA NUEVA APROXIMACION.
SUBROUTINE GCRK(NP,IJ,II,K,A,U,B,R,P,V,S,AP,V1)
IMPLICIT REAL*8 (A-U,Z-Z)
DIMENSION A(NP,7),U(NP),B(NP),R(NP),P(K,NP),V(NP),S(NP),
* AP(NP),V1(NP)
READ(*,*)
N=0
CALL MUL3D(NP,IJ,II,A,U,V)
DO 20 I=1,NP
R(I)=B(I)-V(I)
P(K,I)=R(I)
20 CONTINUE
IC=K
30 IF (IC.NE.K) GO TO 32
DO 35 J=1,K-1
DO 37 I=1,NP
P(J,I)=0.DO
37 CONTINUE
35 CONTINUE
IC=5
32 IC=IC+1
DO 40 I=1,NP
V(I)=P(K,I)
40 CONTINUE
Q=0.DO
ALPHA=0.DO
D=0.DO
CALL MUL3D(NP,IJ,II,A,V,AP)
DO 50 I=1,NP
ALPHA=ALPHA+R(I)*AP(I)
D=D+AP(I)*AP(I)
50 CONTINUE
ALPHA=ALPHA/D
DO 60 I=1,NP
U(I)=U(I)+ALPHA*P(K,I)
60 CONTINUE
Q1=0.DO
DO 70 I=1,NP
R(I)=R(I)-ALPHA*AP(I)
Q1=Q1+R(I)**2
70 CONTINUE
N=N+1
WRITE(*,*) 'RESID',N,Q1
WRITE(3,*) N,Q1
IF ((N.EQ.200).OR.(Q1.LE.10E-04)) RETURN
CALL BASE(NP,IJ,II,K,A,R,S,AP,P,V,V1)

```

```

GOTO 30
END
C*****
C*****
C**
C**          PROGRAM GRADSG FOR
C**
C*****
C*****
C  RESUELVE EL PROBLEMA AU=B, CON A UNA MATRIZ NO SIMETRICA,
C  UTILIZANDO EL METODO TRADICIONAL DE GRADIENTE CONJUGADO PARA
C  EL SISTEMA A*AU=AT*B. LA MATRIZ A DEBE SER EPTAD(AGONAL
C  CON ELEMENTOS IGUALES EN CADA RENGLON
IMPLICIT REAL*8 (A-H,O-Z)
DIMENSION A1(30000)
OPEN(UNIT=1,FILE='MATRIZ.DAT',STATUS='OLD',FORM='BINARY')
OPEN(UNIT=2,FILE='VECTOR.DAT',STATUS='OLD',FORM='BINARY')
OPEN(UNIT=3,FILE='PI0000GG.DAT',STATUS='NEW')
WRITE(*,*) 'CUAL ES LA DIMENSION X,Y,Z'
READ(*,*) II,JJ,KK
IJ=I1*JJ
NP=IJ*KK
IULT=1
IA=IULT
IU=IA+NP*7
IB=IU*NP
IW=IB*NP
IR=IW*NP
IV=IR*NP
IP=IV*NP
IS=IP*NP
IAP=IS*NP
IULT=IAP*NP
IF(IULT.LE.30000) GO TO 100
WRITE(*,*) 'NO CABE'
CALL EXIT
100 CALL GRADSG(II, JJ, NP, A1(IA), A1(IU), A1(IB), A1(IW), A1(IR),
* A1(IV), A1(IP), A1(IS), A1(IAP))
CLOSE (UNIT=1)
CLOSE (UNIT=2)
CLOSE (UNIT=3)
END
C*****
C  LEE LAS ENTRADAS DE LA MATRIZ A DEL ARCHIVO 1 Y LAS DE B
C  DEL ARCHIVO 2. DEFINE U=0 COMO PRIMERA APROXIMACION. LLAMA
C  A LA SUBRUTINA SYH.
SUBROUTINE GRADSG(II, JJ, NP, X, U, B, W, R, V, P, S, AP)
IMPLICIT REAL*8 (A-H,O-Z)
DIMENSION A(NP,7),U(NP),B(NP),U(NP),R(NP),V(NP),P(NP),
+ S(NP),AP(NP)
READ(2) (V(I),I=1,NP)
READ(1) ((A(I,J),J=1,7),I=1,NP)
CALL MUL3DA(NP, JJ, II, A, V, B)
DO 200 I=1, NP
U(I)=0.00

```

```

200 CONTINUE
    CALL SYM(NP,IJ,II,A,U,B,R,P,U,V,S,AP)
    RETURN
    END
.....
C   DESARROLLA EL ALGORITMO DE GRADIENTE CONJUGADO. DEFINIMOS U
C   COMO LA N-ESIMA APROXIMACION, R SU RESIDUO, P LA DIRECCION
C   DE BUSQUERA Y Q1 LA NORMA DE R. EL FIN DE LA SUBROUTINA SE DA
C   DESDE DE LA PANTALLA.
    SUBROUTINE SYM(NP,IJ,II,A,U,B,R,P,U,V,S,AP)
    IMPLICIT REAL*8 (A-H,O-Z)
    DIMENSION A(NP,?),U(NP),B(NP),R(NP),P(NP),W(NP),V(NP),S(NP),
    - AP(NP)
    READ(*,*)
    N=0
    CALL MUL3D(NP,IJ,II,A,U,V)
    CALL MUL3DA(NP,IJ,II,A,V,W)
    DO 20 I=1,NP
    R(I)=B(I)-W(I)
    P(I)=0.00
20 CONTINUE
    BETA=0.00
    DO 40 I=1,NP
    V(I)=P(I)
    P(I)=R(I)-BETA*P(I)
40 CONTINUE
    ALPHA=0.00
    D=0.00
    CALL MUL3D(NP,IJ,II,A,P,V)
    CALL MUL3DA(NP,IJ,II,A,V,AP)
    DO 50 I=1,NP
    ALPHA=ALPHA+R(I)*R(I)
    D=D+P(I)*AP(I)
50 CONTINUE
    ALPHA=ALPHA/D
    DO 60 I=1,NP
    U(I)=U(I)+ALPHA*P(I)
60 CONTINUE
    CALL MUL3D(NP,IJ,II,A,U,V)
    CALL MUL3DA(NP,IJ,II,A,V,W)
    Q1=0.00
    DO 70 I=1,NP
    R(I)=B(I)-W(I)
    Q1=Q1+R(I)*R(I)
70 CONTINUE
    CALL MUL3D(NP,IJ,II,A,R,V)
    CALL MUL3DA(NP,IJ,II,A,V,S)
    N=N+1
    WRITE(*,*)'RESID',N,Q1
    WRITE(3,*) H,Q1
    IF((N.EQ.200).OR.(Q1.LE.10E-04)) RETURN
200 BETA=0.00
    DO 80 I=1,NP
    BETA=BETA+S(I)*P(I)
80 CONTINUE

```

```
BETA=BETA/D
GOTO 30
END
```

Subrutinas

```
C.....
C*
C*                                     B A S E
C*
C.....
C  CONSTRUYE UN VECTOR P(K,J) ORTOGONAL A P(1,J), P(2,J),...,P(K-1,J).
C  J=1,...,NP, POR MEDIO DEL METODO GRAM-SCHMIDT.
SUBROUTINE BASE(NP,IJ,II,K,A,R,S,AP,P,V,V1)
IMPLICIT REAL*8 (A-H,O-Z)
DIMENSION A(NP,7),P(K,NP),R(NP),S(NP),AP(NP),V(NP),V1(NP)
DO 10 J=1,NP
DO 20 I=1,K-1
P(I,J)=P(I+1,J)
20 CONTINUE
P(K,J)=0.D0
10 CONTINUE
DO 30 I=1,K-1
BETA=0.D0
D=0.D0
DO 40 J=1,NP
V(J)=P(I,J)
40 CONTINUE
CALL MUL3D(NP,IJ,II,A,V,AP)
CALL MUL3D(NP,IJ,II,A,R,S)
DO 50 J=1,NP
BETA=BETA*S(J)*AP(J)
D=D+AP(J)**2
50 CONTINUE
IF (D.EQ.0.D0) BETA=0.D0
IF (D.NE.0.D0) BETA=BETA/D
DO 60 J=1,NP
P(K,J)=P(K,J)-BETA*P(I,J)
60 CONTINUE
30 CONTINUE
DO 70 J=1,NP
P(K,J)=P(K,J)+R(J)
70 CONTINUE
DO 75 I=1,NP
V(I)=P(K,I)
75 CONTINUE
CALL MUL3D(NP,IJ,II,A,V,AP)
DO 80 I=1,K-1
DO 100 J=1,NP
V(J)=P(I,J)
100 CONTINUE
BETA=0.D0
```

```

CALL MUL3D(NP,IJ,II,A,Y,V1)
DO 95 J=1,NP
BETA=BETA+V1(J)*AP(J)
95 CONTINUE
90 CONTINUE
RETURN
END

```

C*.....
C*
C* MUL3D
C*
C*.....

```

C MULTIPLICA AM LA MATRIZ AM POR X Y EL RESULTADO SERA Y. AM
C ES UNA MATRIZ NP X 7 - DIMENSIONAL. LA DIMENSION EN X ES II,
C LA DIMENSION DE X POR LA DE Y ES IJ.
SUBROUTINE MUL3D(NP,IJ,II,AM,X,Y)
IMPLICIT REAL*8 (A-H,O-Z)
DIMENSION AM(NP,7),X(NP),Y(NP)
NP1=NP-II
NP2=NP-IJ
Y(1)=AM(1,4)*X(1)+AM(1,5)*X(2)+AM(1,6)*X(II+1)+AM(1,7)*X(IJ+1)
Y(NP)=AM(NP,4)*X(NP)+AM(NP,7)*X(NP1)+AM(NP,3)*X(NP-1)
* +AM(NP,1)*X(NP2)
DO 100 K=2,NP-1
Y(K)=AM(K,4)*X(K)+AM(K,3)*X(K-1)+AM(K,5)*X(K+1)
IF(K.GT.II) Y(K)=Y(K)+AM(K,2)*X(K-II)
IF(K.GT.IJ) Y(K)=Y(K)+AM(K,1)*X(K-IJ)
IF(K.LE.NP2) Y(K)=Y(K)+AM(K,7)*X(K+1J)
IF(K.LE.NP1) Y(K)=Y(K)+AM(K,6)*X(K+II)
100 CONTINUE
RETURN
END

```

C*.....
C*
C* MUL3DA
C*
C*.....

```

C MULTIPLICA AM LA MATRIZ AMT POR X Y EL RESULTADO SERA Y. AM
C ES UNA MATRIZ NP X 7 - DIMENSIONAL. LA DIMENSION EN X ES II,
C LA DIMENSION DE X POR LA DE Y ES IJ.
SUBROUTINE MUL3DA(NP,IJ,II,AM,X,Y)
IMPLICIT REAL*8 (A-H,O-Z)
DIMENSION AM(NP,7),X(NP),Y(NP)
NP1=NP-II
NP2=NP-IJ
Y(1)=AM(1,4)*X(1)+AM(2,3)*X(2)+AM(II+1,2)*X(II+1)
* +AM(IJ+1,1)*X(IJ+1)
Y(NP)=AM(NP,4)*X(NP)+AM(NP1,6)*X(NP1)+AM(NP-1,5)*X(NP-1)
* +AM(NP2,7)*X(NP2)
DO 100 K=2,NP-1
Y(K)=AM(K,4)*X(K)+AM(K-1,5)*X(K-1)+AM(K+1,3)*X(K+1)
IF(K.GT.II) Y(K)=Y(K)+AM(K-11,6)*X(K-11)
IF(K.GT.IJ) Y(K)=Y(K)+AM(K-IJ,7)*X(K-IJ)
IF(K.LE.NP2) Y(K)=Y(K)+AM(Y+IJ,1)*X(K+IJ)
IF(K.LE.NP1) Y(K)=Y(K)+AM(K+II,2)*X(K+II)

```

```

100 CONTINUE
    RETURN
    END
C*****
C*****
C**                                     **
C**                                     **
C**                                     **
C*****
C*****
C DA VALORES A LAS ENTRADAS DE LA MATRIZ A EN BASE A LA ECUACION
C DE TRANSPORTE CON DIFUSION EN ESTADO ESTACIONARIO.
  IMPLICIT REAL*8 (A-H,O-Z)
  DIMENSION A1(30000)
  OPEN(UNIT=1,FILE='MATRIZ.DAT',FORM='BINARY')
  OPEN(UNIT=2,FILE='VECTOR.DAT',FORM='BINARY')
  WRITE(*,*) 'CUALES SON LAS DIMENSIONES X,Y,Z'
  READ(*,*) IJ,JJ,KK
  IJ=IJ*JJ
  NP=IJ*KK
  IULT=1
  IA=IULT
  IU=IA+NP*7
  IB=IU+NP
  IULT=IB+NP
  IF(IULT.GT.30000) CALL EXIT
  CALL GENMAT(NP,IJ,IJ,A1(IA),A1(IU),A:(IB))
  CLOSE(UNIT=1)
  CLOSE(UNIT=2)
  END
C*****
C DA VALORES A LAS ENTRADAS DE LA MATRIZ A, DEFINE B=AU DONDE
C U TIENE VALOR 1 EN TODAS SUS COMPONENTES. DEFINE U=0 COMO
C PRIMERA APROXIMACION. LLAMA A LA SUBROUTINA GENM.
  SUBROUTINE GENMAT(NP,IJ,II,A,U,B)
  IMPLICIT REAL*8 (A-H,O-Z)
  DIMENSION A(NP,7),U(NP),B(NP)
  WRITE(*,*) 'CUAL ES EL NUMERO DE PECLET?'
  READ(*,*) PE
  WRITE(*,*) 'DE QUE TAMAÑO SON LOS LADOS DE LAS CELDAS DE LA RED?'
  READ(*,*) H
  T=3.DO*H
  V=SQRT(PE/T)
  Q=-V/2.DO-1.DO/H
  X1=Q
  X2=Q
  X3=Q
  A=Q DO/H
  Q=V/2.DO-1.DO/H
  X5=Q
  X6=Q
  X7=Q
  WRITE(*,*)X1,X2,X3,X4,X5,X6,X7
  DO 100 I=1,NP
  A(I,4)=X4
  
```

```

A(I,3)=X3
A(I,5)=X5
A(I,2)=X2
A(I,1)=X1
A(I,7)=XY
A(I,6)=X6
B(I)=0.D0
IF(I.GE.IJ+1) GO TO 10
B(I)=B(I)-1.D0*A(I,1)
A(I,1)=0.D0
10 IF(MOD(I-1,IJ).GE.IJ) GO TO 20
B(I)=B(I)-1.D0*A(I,2)
A(I,2)=0.D0
20 IF(MOD(I,II).NE.1) GO TO 30
B(I)=B(I)-1.D0*A(I,3)
A(I,3)=0.D0
30 IF(MOD(I-1,IJ).LE.IJ-II-1) GO TO 40
A(I,6)=0.D0
40 IF (MOD(I,II).NE.0) GO TO 50
A(I,5)=0.D0
50 IF (I.LE.NP-IJ) GO TO 100
A(I,7)=0.D0
100 CONTINUE
DO 150 I=1,NP
WRITE(2) B(I)
DO 200 J=1,7
WRITE(1) A(I,J)
200 CONTINUE
150 CONTINUE
RETURN
END

```

C*.....

ESTA TESIS NO DEBE
SALIR DE LA BIBLIOTECA

Referencias

- [1] Allen, M.B., Herrera, I. and Pinder, G.F., *Numerical Modeling in Science and Engineering*, John Wiley and Sons, 1988.
- [2] Birkhoff, G. and Lynch, R.E., *Numerical Solution of Elliptic Problems*, Philadelphia: SIAM, 1991.
- [3] Celis, M.A., Herrera I., Boulioutas, E.T., and Kindred, J.S., A New Numerical Approach for the Advective Diffusive Transport Equation, *Numer. Methods for Partial Differential Equations* 5 no.3, 1989.
- [4] Eisenstat, S.C., Elman, R.C. and Schultz, M.H., Variational Iterative Methods for Nonsymmetric Systems of Linear Equations, *SIAM J. Numer. Anal.* vol. 20, no.2, april 1983.
- [5] Freidberg, S.H., Insel, A.J. y Spence, L.E., *Algebra Lineal*, Publicaciones Cultura, S.A., 1982.
- [6] Golub, G.H., Van Loan, C.F. *Matrix Computations*, Baltimore: Johns Hopkins University Press, 1983.
- [7] Gómez, S. and Morales, J.L., Performance of the Chebyshev Iterative Method, GMRES and Orthomin on a Set of Oil Reservoir Simulation Problems, in: *Mathematics for Large Scale Computing*, capítulo 11, Díaz, J.C., ed., 1989.
- [8] Hestenes, M.R. and Stiefel, E., Methods of Conjugate Gradients for Solving Linear Systems, *J. Res. Nat. Bur. Standards*, 49 (1952) pp. 109-125.
- [9] Hestenes, M.E., *Applications of Mathematics Conjugate Direction Methods in Optimization*, Springer-Verlag, 1980.
- [10] Herrera, I., Un Análisis del Método de Gradiente Conjugado, *Comunicaciones Técnicas del Instituto de Geofísica*, USAM, Serie Investigación, no.7, 1988.
- [11] Maltsev, A.I. *Fundamentos de l Algebra Lineal*, Siela Veintiuno Editores S.A., 1970.
- [12] Reid, J.K., On the Method of Conjugate Gradients for the Solution of Large Sparse Systems of Linear Equations, in: *Large Sparse Sets of Linear Equations*, J.K. Reid, ed., Academic Press, New York, 1971, pp. 231-261.
- [13] Saad, Y. and Schultz M.H., GMRES: A generalized Residual Algorithm for Solving Nonsymmetric Linear Systems, *SIAM J. Sci. Stat. Comput.* 7, no.3, 1986.
- [14] Vinsome, P.K.W., ORTHOMIN, an Iterative Method for Solving Sparse Sets of Simultaneous Linear Equations, in: *Proc. Fourth Symposium on Reservoir Simulation*, Society of Petroleum Engineers of AIME, 1976.