Universidad de Las Palmas de Gran Canaria
Departamento de Matemáticas

Tesis Doctoral

Construcción de inversas aproximadas tipo *sparse* basada en la proyección ortogonal de Frobenius para el precondicionamiento de sistemas de ecuaciones no simétricos

Autor: Elizabeth Flórez Vázquez
Directores: Gustavo Montero García
Luis González Sánchez

La Doctorando	El Director	El Director
Fdo.: Elizabeth Flórez Vázquez	Fdo.: Gustavo Montero García	Fdo.: Luis González Sánchez

Las Palmas de Gran Canaria, Marzo de 2003

A mi hijo

Agradecimientos

Quisiera agradecer primeramente a Gustavo Montero y Luis González sin cuya ayuda hubiera sido imposible realizar de este trabajo.

Además a Eduardo Rodríguez Barrera que pacientemente me ha ayudado a resolver los imprevistos informáticos y a resolver todas las dudas en el tema.

A todos los que de una forma u otra han contribuido a la terminación de esta tesis.

A todos, muchas gracias.

Esta tesis ha sido desarrollada en el marco del proyecto subvencionado por el Ministerio de Ciencia y Tecnología, REN2001-0925-C03-02/CLI, titulado *Modelización numérica de transporte de contaminantes en la atmósfera.*

Índice general

Índice de figuras

Índice de tablas

Capítulo 1

Introducción

Uno de los problemas más importantes en la ciencia de la computación es el desarrollo de Métodos iterativos paralelizables y eficientes para resolver sistemas de ecuaciones lineales $Ax = b$ con matriz de coeficientes A de orden elevado y *sparse*. Los Métodos basados en los subespacios de Krylov efectúan productos matriz-vector en cada iteración y pocas operaciones con vectores (producto escalar y actualizaciones de vectores). Estos métodos pueden ser implementados eficientemente en ordenadores de gran capacidad pero necesitan del precondicionamiento para ser efectivos. La mayoría de los precondicionadores de propósitos generales tales como los basados en la factorización incompleta de A, son suficientemente robustos y permiten obtener una buena velocidad de convergencia, pero son altamente secuenciales lo que dificulta su implementación eficiente en ordenadores en paralelo, especialmente para problemas no estructurados. Así, el precondicionamiento es normalmente la mayor dificultad en la resolución de grandes sistemas *sparse*.

Se ha realizado una gran cantidad de trabajos desde los inicios del los procesadores vectoriales y paralelos, enfocados a paralelizar todo lo posible los mejores precondicionadores tales como SSOR y los métodos de factorización incompleta. Como resultado de esto, es posible lograr una buena ejecución para algunos tipos de problemas con matrices altamente estructuradas como los que se obtienen de la discretización de ecuaciones en derivadas parciales en mallas regulares. Por otra parte, todavía es muy difícil resolver eficientemente sistemas lineales generales con un patrón de *sparsidad* irregular en ordenadores vectoriales.

Otra línea de investigación consiste en el desarrollo alternativo de métodos de precondicionamiento que son paralelizables de forma natural. Dentro de las primeras técnicas de este tipo podemos mencionar los precondicionadores polinomiales, que se basan en aproximar la inversa de la matriz de coeficientes A con un polinomio de grado pequeño en la matriz. Estos métodos tienen una larga historia, [27, 30] pero se popularizaron sólo después que los primeros procesadores vectoriales aparecieran en el mercado, [42, 68]. Los precondicionadores polinomiales únicamente requieren productos matriz-vector con A y,

sin embargo, tienen un excelente potencial de paralelización, pero no son tan efectivos en la reducción del número de iteraciones como los métodos de factorización incompleta. Con los precondicionadores polinomiales, la reducción del número de iteraciones tiende a compensarse con los productos matriz-vector que se efectúan en cada iteración. Con más precisión, Axelsson [7] demostró que el costo por iteración aumenta linealmente con el número $m + 1$ de términos en el polinomio mientras que el número de iteraciones disminuye más lentamente que $O(1/m + 1)$. Por lo tanto, los precondicionadores polinomiales no pueden ser muy efectivos, especialmente en ordenadores en serie o de memoria compartida. Para ordenadores en paralelo con memoria distribuida, estos precondicionadores son un poco más atractivos debido al número reducido de productos escalares y accesos a la matriz. Otra característica atractiva es que el espacio de almacenamiento de los precondicionadores polinomiales no es mayor que el requerido para la matriz de los coeficientes A y requieren pequeños tiempos de preparación. También pueden usarse en un contexto libre de matrices puesto que se combinan fácilmente con otros precondicionadores. Por otra parte, los precondicionadores polinomiales tienen serias dificultades de manipulación en el caso de matrices con un espectro general complejo, por ejemplo, autovalores en ambos lados del eje inmaginario. Resumiendo, es justo decir que los precondicionadores polinomiales no parecen ser una solución satisfactoria del problema del precondicionamiento para matrices *sparse* en general.

En los últimos años, se ha incrementado el interés en otros tipos de precondicionadores algebraicos: las inversas aproximadas *sparse*. Estos métodos se basan en aproximar la inversa de la matriz directamente, al igual que los precondicionadores polinomiales. Sin embargo, con los precondicionadores polinomiales, las inversas aproximadas pueden encontrarse sólo implícitamente en forma de un polinomio en la matriz de coeficientes A, en contraste con el precondicionador del tipo inversa aproximada *sparse*, donde la matriz $M \approx A^{-1}$ se calcula explícitamente y se almacena. El precondicionamiento se reduce, pues, a un producto matriz-vector con M. Los métodos de este tipo fueron propuestos primeramente al principio de los 70 pero recibieron muy poca atención debido a la falta de estrategias efectivas para la determinación automática de un buen patrón de entradas no nulas para la inversa aproximada *sparse*. Recientemente se han desarrollado nuevas estrategias que han renovado el interés por este tipo de precondicionadores [60, 24, 84].

Mientras que los procesadores en paralelo han sido la principal motivavión del desarrollo de los métodos de inversas aproximadas, también ha habido al menos otra influencia. Es bien conocido que las técnicas de factorización incompleta fallan en matrices no simétricas y/o indefinidas. El fallo se debe comúnmente a una forma de inestabilidad, tanto en el paso de la factorización incompleta como tal (los ceros o pivotes muy pequeños), como en el paso de sustitución hacia atrás o en ambos. La mayoría de las técnicas de inversas aproximadas son inmunes a estos problemas, y por lo tanto constituyen un comple-

mento importante a los métodos de precondicionamiento más clásicos aún en ordenadores con arquitectura de von Neumann.

Debemos resaltar que las técnicas de inversas aproximadas se apoyan en asumir tácitamente que para una matriz *sparse* dada A es posible encontrar una matriz *sparse* M que sea una buena aproximación de A^{-1}. Sin embargo, esto no es tan obvio puesto que la inversa de una matriz *sparse* normalmente es densa. Con más precisión, puede probarse que la inversa de una matriz *sparse* irreducible es normalmente densa. Esto significa que para un patrón de *sparsidad* irreducible, siempre es posible asignar valores numéricos a las entradas no nulas de tal forma que todas las entradas de la inversa sean no nulas. No obstante, es muy común el caso de que muchas de las entradas en la inversa de una matriz *sparse* son pequeñas en valor absoluto, así, la aproximación de A^{-1} con una matriz *sparse* es posible. Recientemente se han realizado más estudios sobre el difícil problema de obtener las entradas más "importantes" de A^{-1} automáticamente [61, 84, 24].

Existen propuestas alternativas para construir precondicionadores del tipo inversa aproximada *sparse*, algunos de las cuales se han comparado con los métodos de factorización incompleta. Una comparación directa entre las diversas técnicas de inversas aproximadas puede verse en [24] donde se compara la efectividad de varios métodos para diferentes tipos de problemas.

Los técnicas de inversa aproximada pueden agruparse en tres categorías: los métodos de inversa aproximada basados en la minimización de la norma Frobenius, las inversas aproximadas factorizadas y los métodos de precondicionamiento consistentes en una factorización incompleta seguida de una obtención aproximada de las inversas de los factores incompletos. La obtención de la inversa aproximada puede realizarse de muchas formas cada una de las cuales nos lleva a la obtención de un precondicionador diferente. En la sección siguiente se dará una visión general sobre las distintas técnicas de inversas aproximadas y sus características. En toda la sección nos ocuparemos de un sistema lineal de la forma $Ax = b$ donde A es una matriz real de orden $n \times n$ y b es un vector real de n componentes.

1.1. Métodos basados en la minimización de la norma de Frobenius

Esta es la primera técnica de cálculo de inversa aproximada propuesta y es uno de los métodos mejor conocido. Este método es también altamente paralelizable. La idea básica es calcular una matriz *sparse* $M \approx A^{-1}$ que sea la solución del siguiente problema de minimización,

$$\min_{M \in \mathcal{S}} \|I - AM\|_F$$

donde $\mathcal{S}$ es el subespacio de las matrices *sparse* y $\|\cdot\|_F$ es la norma de Frobenius de una matriz. Como

$$\|I - AM\|_F^2 = \sum_{j=1}^{n} \|e_j - Am_j\|_2^2$$

donde e_j es la j-ésima columna de la matriz identidad, el cálculo de M se reduce a resolver n problemas de mínimos cuadrados independientes sujeto a las restricciones de *sparsidad*. Este camino fue propuesto inicialmente por Benson [12].

En los trabajos iniciales, el subespacio de partida fue el subespacio restringido $\mathcal{S}$ consistente en las matrices con un patrón de *sparsidad* dado. Para un subespacio $\mathcal{S}$ dado, el cálculo de M es directo y es posible implementarlo eficientemente en paralelo. En un medio de memoria distribuida, la matriz de coeficientes A puede distribuirse entre los procesadores antes de la ejecución de los cálculos y la construcción de M es un proceso local que puede realizarse con poca comunicación entre los procesadores, tal comunicación debe eliminarse completamente duplicando algunas de las columnas de A.

Cuando se fija el patrón de *sparsidad* por adelantado, la construcción del precondicionador puede realizarse como sigue. El patrón de entradas no nulas es el subconjunto $\mathcal{G} \subseteq \{(i,j)/1 \leq i,j \leq n\}$ tal que $m_{i,j} = 0$ si $(i,j) \notin \mathcal{G}$. Así, el subespacio restringido $\mathcal{S}$ es simplemente el conjunto de todas las matrices reales de orden n con patrón de entradas no nulas contenido en $\mathcal{G}$. Denotando como m_j la j-ésima columna de $M(1 \leq j \leq n)$ Para un j fijo, se considera el conjunto $\mathcal{J} = \{i/(i,j) \in \mathcal{G}\}$ el cual especifica el patrón de entradas no nulas de m_j. Claramente, las únicas columnas de A que entran en la definición de m_j son aquellas cuyo índice está en $\mathcal{J}$. Sea $A(:,\mathcal{J})$ la submatriz de A formada por tales columnas y sea $\mathcal{I}$ el conjunto de índices de filas no nulas de $A(:,\mathcal{J})$. Entonces podemos restringir nuestra atención a la matriz $\hat{A} = A(\mathcal{I},\mathcal{J})$ y al vector incógnita $\hat{m} = m_j(\mathcal{J})$ y para el lado derecho $\hat{e}_j = e_j(\mathcal{I})$. Las entradas no nulas en m_{ij} pueden calcularse resolviendo el pequeño y no restringido problema de mínimos cuadrados,

$$\text{mín} \left\| \hat{e}_j - \hat{A}\hat{m}_j \right\|_2$$

El problema de mínimos cuadrados puede resolverse, por ejemplo, mediante una factorización QR de $\hat{A}$. Claramente, cada columna m_j puede calcularse, al menos en principio, independientemente de otras columnas de M. Observe que debido a la *sparsidad* de A, la submatriz $\hat{A}$ contendrá solo unas pocas filas y columnas no nulas, así cada problema de minimización tiene un tamaño pequeño y puede resolverse eficientemente por técnicas para matrices densas.

El papel del subespacio restringido $\mathcal{S}$ es preservar la *sparsidad* filtrando aquellas entradas de A^{-1} que contribuyan poco a la calidad del precondicionador. Por ejemplo, puede que queramos ignorar aquellas entradas pequeñas en valor absoluto y retener las más grandes. Desafortunadamente, para una matriz en general *sparse*, normalmente no se conoce la localización de las entradas

con valores grandes en la inversa, por lo que, seleccionar un patrón de *sparsidad* para la inversa aproximada a priori es muy difícil. Una posible excepción es el caso en que A es una matriz en banda simétrica y definida positiva. En este caso, las entradas de A^{-1} están acotadas de forma exponencialmente decreciente en cada fila o columna; ver [24]. Específicamente, existe ρ, $0 < \rho < 1$, y una constante C tal que para todo i, j

$$\left|(A^{-1})_{ij}\right| \leq C\rho^{|i-j|}$$

Los números ρ y C dependen del ancho de banda y del número de condición espectral de A. Para matrices con un ancho de banda grande y/o un número de condición alto, C puede ser muy grande y ρ muy cercano a uno, entonces el decrecimiento puede ser tan lento que es virtualmente imperceptible. Por otra parte, si puede demostrarse que las entradas de A^{-1} decrecen rápidamente, entonces una buena aproximación de A^{-1} puede ser una matriz en banda M. En este caso S debe ser un conjunto de matrices con un ancho de banda prefijado.

Para matrices con un patrón de *sparsidad* general, la situación es mucho más difícil. Lo más común es seleccionar S como el subespacio de las matrices que tienen el mismo patrón de *sparsidad* que la matriz de los coeficientes A. Esta selección está motivada por la observación empírica de que las entradas en la inversa de una matriz *sparse* A en las posiciones (i, j) para las cuales $a_{ij} \neq 0$ tienden a ser relativamente grandes. Sin embargo, este camino simple no es robusto para problemas *sparse* en general puesto que pueden existir entradas grandes de A^{-1} en posiciones fuera del patrón de entradas no nulas de A. Otro camino usado consiste en tomar el patrón de *sparsidad* de la inversa aproximada como el de A^k donde k es un entero positivo, $k \geq 2$. Este camino puede justificarse en términos de la expansión de la serie de Neumann de A^{-1}. Mientras que la inversa aproximada correspondiente a potencias mayores de A son a menudo mejores que las correspondientes a $k = 1$, no existe aún la garantía de que resulten precondicionadores satisfactorios. Más aún, el coste computacional, de almacenamiento y aplicación del precondcionador crece rápidamente con k. Otras estrategias han sido examinadas por Huckle [66], pero necesitamos más evidencias para que dichas técnicas puedan considerarse efectivas.

Debido a que para las matrices *sparse* en general es muy difícil prefijar un patrón de entradas no nulas para M, muchos autores han desarrollado estrategias adaptativas que comienzan con una aproximación inicial, por ejemplo, una matriz diagonal y aumentan sucesivamente este patrón hasta que se satisface el criterio del tipo $\|e_j - Am_j\|_2 < \varepsilon$ para un ε dado (para cada j), o se alcanza el número máximo de términos no nulos en m_j. Este camino fue propuesto primeramente por Cosgrove et al. [38]. Otras estrategias ligeramente diferentes han sido consideradas por Grote y Huckle [60] y Gould y Scott [57].

El objetivo de esta tesis es construir un algoritmo para obtener la inversa aproximada explícita con un patrón de *sparsidad* desconocido, partiendo de el camino seguido por Grote et al.[60] pero cambiando el método de selección de

las entradas en la aproximación inicial de la inversa aproximada, lo cual se trata con profundidad en el Capítulo 5.

Debemos mencionar también la sensibilidad de estos precondicionadores a la reordenación. Es bien conocido que los precondicionadores del tipo factorización incompleta son muy sensibles a la reordenación. Sin embargo, no existen muchos estudios sobre los efectos de ésta en los precondicionadores del tipo inversa aproximada. Otro objetivo de esta tesis es comprobar si la reordenación reduce el número de entradas no nulas en el precondicionador del tipo inversa aproximada y si además mejora la convergencia de los Métodos iterativos cuando se utilizan dichos precondicionadores. En el Capítulo 6 se estudia con profundidad lo expuesto y se muestran los resultados obtenidos.

1.2. Inversas aproximadas *sparse* factorizadas

Existen otros precondicionadores basados en la factorización incompleta de la inversa, esto es, la factorización incompleta de A^{-1}. Si A admite la factorización $A = LDU$ donde L es una matriz triangular inferior con diagonal unidad, D es una matriz diagonal y U es una matriz triangular superior, entonces A^{-1} puede factorizarse $A^{-1} = U^{-1}D^{-1}L^{-1} = ZD^{-1}W^T$ donde $Z = U^{-1}$ y $W = L^{-T}$ son matrices triangulares superiores con diagonal unidad. Obsérvese que, en general, los factores inversos Z y W serán más densos. Por ejemplo, si A es una matriz irreducible en banda, estos factores estarán completamente llenos por encima de la diagonal principal. Los precondicionadores del tipo inversa aproximada *sparse* pueden construirse calculando las aproximaciones *sparse* $\bar{Z} \approx Z$ y $\bar{W} \approx W$. La inversa aproximada factorizada es

$$M = \bar{Z}\bar{D}^{-1}\bar{W}^T \approx A^{-1}$$

donde $\bar{D}$ es una matriz diagonal no singular $\bar{D} \approx D$.

Existen muchos caminos para el cálculo de los factores de la inversa aproximada de una matriz no singular A. Citamos un primer tipo de métodos que no necesita información sobre los factores triangulares de A: el precondicionador de inversa aproximada factorizada se construye directamente de A. Los métodos de este tipo incluyen el precondicionador FSAI presentado por Kolotilina y Yeremin [76], un método relacionado debido a Kaporin [69], los esquemas incompletos de biconjugación [20, 22], y las estrategias de orlado [94]. Otros tipos de métodos calculan primero la factorización triangular incompleta de A usando técnicas clásicas y después obtienen una inversa aproximada *sparse* factorizada calculando las aproximaciones *sparse* a la inversa de los factores triangulares incompletos de A.

1.2.1. Método FSAI

El método FSAI propuesto por Kolitilina y Yeremin [76] puede describirse brevemente como sigue. Asumiendo que A es una matriz simétrica y definida positiva (SPD), sea $\mathcal{S}_L$ un patrón de *sparsidad* triangular inferior definido que incluye la diagonal principal. Entonces la matriz triangular inferior $\hat{G}_L$ se calcula resolviendo la ecuación matricial

$$\left(A\hat{G}_L\right)_{ij} = \delta_{ij} \quad (i,j) \in \mathcal{S}_L$$

Donde $\hat{G}_L$ se calcula por columnas: cada columna necesita la solución de un pequeño sistema SPD "local", cuyo tamaño es igual al número de entradas no nulas permitidos en dicha columna. Todas las entradas de la diagonal de $\hat{G}_L$ son positivas. Definiendo $\hat{D} = diag(\left(\hat{G}_L\right))^{-1}$ y $\hat{G}_L = \hat{D}^{1/2}\hat{G}_L$, entonces la matriz precondicionada $\hat{G}_L A \hat{G}_L^T$ es simétrica y definida positiva y tiene las entradas en la diagonal igual a 1. Una forma común de seleccionar el patrón de *sparsidad* es permitir que las entradas no nulas en $\hat{G}_L$ sean solamente en las posiciones correspondientes a las entradas no nulas en la parte triangular inferior de A. Otra selección más costosa pero más sofisticada es considerar el patrón de *sparsidad* de la parte inferior de A^k donde k es un entero positivo y pequeño, por ejemplo, $k = 2$ o $k = 3$ [69].

El factor de la inversa aproximada calculado por el método FSAI puede mostrarse para minimizar $\|I - XL\|_F$ donde L es el factor de Cholesky de A, sujeto a la restricción de *sparsidad* $X \in \mathcal{S}_L$. Sin embargo, no es necesario conocer L para calcular $\hat{G}_L$. La matriz $\hat{G}_L$ minimiza en $X \in \mathcal{S}_L$ el número de condición de Kaporin de XAX^T (ver [69]),

$$\frac{1}{n}\, tr\left(XAX^T\right) / det\left(XAX^T\right)^{1/n}$$

donde n es el orden del problema y $tr(\cdot)$ y $det(\cdot)$ representan la aplicación traza y determinante de una matriz, respectivamente. La extensión de FSAI al caso no simétrico es directa, sin embargo, no se garantiza la posibilidad de resolver los sistemas lineales locales y la no singularidad de la inversa aproximada aún cuando todos los menores principales angulares de A sean no nulos, por lo que se necesita algún control de seguridad. El precondicionador FSAI puede ser implementado en paralelo de forma eficiente y se ha aplicado con éxito a la solución de problemas difíciles en el análisis de estructuras mediante elementos finitos [48],[77]. Su principal desventaja es la necesidad de fijar previamente el patrón de *sparsidad* de los factores de la inversa aproximada. Una solución simple pero a menudo inefectiva, es usar el patrón de *sparsidad* de la parte triangular inferior de A para resolver problemas *sparse* en general. En [77] se describen distintas selecciones del patrón de *sparsidad* para matrices simétricas y definidas positivas obtenidas en el análisis de elementos finitos. También pueden verse los experimentos numéricos en [107].

1.2.2. Método AINV

Otro método de calcular una inversa aproximada factorizada es el basado en la biconjugación incompleta, propuesto primeramente en [14]. Este camino denominado método AINV se describe detalladamente en [20] y [22]. El método AINV no necesita el conocimiento previo del patrón de *sparsidad* y puede aplicarse a matrices con patrones de *sparsidad* generales. La construcción del precondicionador AINV se basa en un algoritmo que calcula dos conjuntos de vectores $\{z_i\}_{i=1}^n$ y $\{w_i\}_{i=1}^n$ que son $A-$biconjugados tal que $w_i^T A z_j = 0$ si y solo si $i \neq j$. Dada una matriz no singular $A \in \mathbb{R}^n$, existe una relación estrecha entre el problema de invertir A y el problema de calcular los dos conjuntos de vectores $A-$biconjugados $\{z_i\}_{i=1}^n$ y $\{w_i\}_{i=1}^n$. Si

$$Z = \{z_i, z_2, ..., z_n\}$$

es la matriz cuya i-ésima columna es z_i y

$$W = \{w_1, w_2, ..., w_n\}$$

es la matriz cuya i-ésima columna es w_i, entonces

$$W^T A Z = \begin{pmatrix} p_1 & 0 & \cdots & 0 \\ 0 & p_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & p_n \end{pmatrix},$$

donde $p_i = w_i^T A z_i \neq 0$. Se sigue que W y Z son necesariamente no singulares y

$$A^{-1} = Z D^{-1} W^T = \sum_{i=1}^{n} \frac{z_i w_i^T}{p_i}$$

De aquí que la inversa de A se conoce si los dos conjuntos de vectores $A-$biconjugados son conocidos. Existen infinitos conjuntos de estos vectores. Las matrices W y Z cuyas columnas son $A-$biconjugadas pueden calcularse explícitamente por medio de un proceso de biconjugación aplicado a las columnas de cualquier par de matrices no singulares $W^{(0)}, Z^{(0)} \in \mathbb{R}^{n \times n}$. Una selección conveniente desde el punto de vista computacional es hacer $W^{(0)} = Z^{(0)} = I$, aplicándose el proceso de biconjugación a una base canónica de vectores.

1.2.3. Método de orlado.

Un tercer camino que se utiliza para calcular el precondicionador del tipo inversa aproximada factorizado directamente de la matriz de entrada A se basa en el orlado. En la actualidad son posibles muchos esquemas. El método que se describe aquí es una modificación propuesta por Saad en [94], (pag.308). Sea A_k

la submatriz principal de A de orden $k \times k$. Considérese el siguente esquema de orlado:

$$\begin{pmatrix} W_k^T & 0 \\ w_k^T & 1 \end{pmatrix} \begin{pmatrix} A_k & v_k \\ y_k^T & \alpha_{k+1} \end{pmatrix} \begin{pmatrix} Z_k & z_k \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} D_k & 0 \\ 0 & \delta_{k+1} \end{pmatrix}$$

donde $z_k = -Z_k D_k^{-1} W_k^T v_k$, $w_k = -W_k D_k^{-1} Z_k^T y_k$ y $\delta_{k+1} = \alpha_{k+1} + w_k^T A_k z_k + y_k^T z_k + w_k^T v_k$. Aquí W_k y Z_k son matrices trangulares superiores de orden k. Por otra parte, w_k, v_k, y_k y z_k son vectores de orden k, mientras que δ_{k+1} y α_{k+1} son escalares, siendo $\alpha_{k+1} \equiv a_{k+1,k+1}$. Comenzando en $k = 1$, este esquema sugiere un algoritmo obvio para el cálculo de los factores inversos de A (asumiendo que A admite una factorización LU). Cuando este esquema se ejecuta de forma incompleta, se obtiene una factorización aproximada de A^{-1}. Puede preservarse la *sparsidad* eliminando elementos de los vectores w_k y z_k una vez calculados, por ejemplo, usando cierta tolerancia al igual que en el proceso AINV. El precondicionador resultante de la factorización de la inversa aproximada *sparse* se denominará AIB (Approximate Inverse via Bordering), es decir, inversa aproximada *sparse* mediante orlado. Aparte de un producto matriz-vector con A_k, la construcción del precondicionador AIB requiere cuatro productos *sparse* matriz-vector que involucran a W_k, Z_k y sus traspuestas en cada paso k, lo cual constituye la mayor parte del trabajo a realizar. Es importante que estas operaciones se efectúen en modo *sparse-sparse*. Nótese que los cálculos para los factores inversos Z y W están fuertemente acoplados, en contraste con el algoritmo de biconjugación. Como siempre, si A es simétrica, $W = Z$ y el trabajo se reduce a la mitad. Más aún, si A es simétrica y definida positiva, tal como se muestra en [94], en aritmética exacta, $\delta_k > 0$ para todo k. Así, el precondicionador AIB está siempre bien definido en el caso simétrico y definido positivo. En el caso general, se requieren modificaciones en la diagonal para completar el proceso.

Las inversas aproximadas *sparse* factorizadas no tienen otros problemas que limiten la efectividad como ocurre con otros métodos. Al contrario que los métodos explicados en las secciones previas, las inversas aproximadas *sparse* factorizadas pueden utilizarse como precondicionadores para el método del gradiente conjugado para resolver problemas simétricos y definidos postitivos. Así, si A es simétrica y definida positiva, entonces $Z = W$ y el precondicionador $M = \bar{Z}\bar{D}^{-1}\bar{Z}^T$ es simétrico y definido positivo según las entradas de la diagonal $\bar{D}$ sean todas positivas. Está claro que la no singularidad de M es trivialmente controlable cuando M se expresa de foma factorizada. Siguiendo [32] puede decirse que las formas factorizadas nos dan una mejor aproximación de A^{-1} para la misma capacidad de almacenamiento que las no factorizada porque pueden ser matrices más densas comparando con el número total de entradas no nulas de sus factores. Tal y como muestran los experimentos en [24], esta observación intuitiva casi se traduce en una mejora en la convergencia para el mismo número de entradas no nulas. Más aún, las formas factorizadas son más baratas de calcular y en la mayoría de los casos requieren definir menos pará-

metros que las técnicas mencionadas en las secciones previas. Finalmente, las formas factorizadas son sensibles al reordenamiento de la matriz de coeficientes, propiedad que puede explotarse para reducir el efecto de llenado en los factores inversos y/o mejorar la velocidad de convergencia, ver [22], [14], [28].

Por otra parte, las formas factorizadas tiene un problema. Al ser métodos de factorización (inversa) incompleta pueden fallar debido a interrupciones en el proceso de factorización incompleta como el ILU. Mientras que las estrategias de desviación de la diagonal [80] o las de compensación de la diagonal pueden usarse como protección de los cálculos, no existe garantía de que el precondicionador resultante sea efectivo, especialmente cuando se necesitan grandes y/o numerosas desviaciones. El método FSAI requiere prescribir la estructura de los factores no nulos por adelantado, lo que hace difícil su uso en problemas con patron de *sparsidad* generales. Otros métodos como AINV, ofrecen oportunidades limitadas de factorización en la fase de construcción del precondicionador. En su formulación corriente, AINV parece difícil de implementar en máquinas de memoria distribuida. También el paralelismo en la aplicación de inversas aproximadas factorizadas es algo menos que en las formas no factorizadas, puesto que en las primeras es necesario ejecutar dos productos matriz-vector con los factores secuencialmente.

1.3. Técnicas de ILU inversa

Muchos autores han propuesto construir precondicionadores del tipo inversas aproximadas *sparse* factorizadas basados en el siguiente proceso en dos etapas: primero una factorización LU incompleta $A \approx \bar{L}\bar{U}$ que se calcula usando técnicas clásicas y después una aproximaxión de la inversa de los factores $\bar{L}$ y $\bar{U}$, ver [2, 40, 99, 100, 44]. Existen varias posibilidades para calcular inversas aproximadas de $\bar{L}$ y $\bar{U}$, y cada una nos permite obtener diferentes precondicionadores.

Asumimos que los factores incompletos $\bar{L}$ y $\bar{U}$ existen. Entonces los factores inversos aproximados pueden calcularse resolviendo de forma inexacta los $2n$ sistemas lineales triangulares

$$\bar{L}x_i = e_i, \quad \bar{U}y_i = e_i \quad (1 \leq i \leq n)$$

Obsérvese que todos estos sistemas lineales pueden resolverse independientemente, por lo que podemos contar con realizar una buena paralelización al menos en principio. Estos sistemas lineales se resuelven aproximadamente para reducir el tiempo de computación y porque debe conservarse la *sparsidad* en las columnas de los factores inversos aproximados.

Una posibilidad es prefijar los patrones de *sparsidad* $\mathcal{S}_L$ y $\mathcal{S}_U$ para las inversas aproximadas de $\bar{L}$ y $\bar{U}$. Las aproximaciones *sparse* pueden calcularse usando la norma Frobenius. El método adaptativo SPAI puede usarse como alternativa

para invertir aproximadamente $\bar{L}$ y $\bar{U}$, sin necesidad de prefijar los patrones de *sparsidad*. A partir de algunos experimentos realizados en [44] se concluyó que este camino no es recomendable.

Un camino mejor consiste en resolver los $2n$ sistemas triangulares para las columnas de $\bar{L}^{-1}$y $\bar{U}^{-1}$, por descenso y remonte respectivamente. Se conserva la *sparsidad* introduciendo las entradas en los vectores soluciones teniendo en cuenta tanto las posiciones (de forma más general, usando un esquema de nivel de llenado) como la tolerancia. Se han descrito con detalle muchos de estos esquemas [40, 100, 44]. Algunos autores han propuesto introducir en $x_i = \bar{L}^{-1}e_i$ y $y_i = \bar{U}^{-1}e_i$, una vez que se calcula la solución exacta, pero un esquema más práctico es introducir durante el proceso de sustitución mejor que después [44].

Los precondicionadores de esta clase comparten algunas ventajas de los métodos de inversas aproximadas factorizadas pero tienen ciertas desventajas que los precondicionadores descritos en las secciones previas no tienen. Estas desventajas radican en asumir que se ha calculado la factorización ILU. Esto implica que estos métodos no son aplicables si la factorización ILU no existe o si es inestable como es el caso, algunas veces, de problemas altamente no simétricos e indefinidos [35, 46]. Asumir esto también limita la eficiencia de la paralelización de esta clase de métodos puesto que la fase de construcción del precondicionador no es enteramente paralelizable (calcular una factorización ILU es un proceso altamente secuencial).

Otra desventaja es que el cálculo del precondicionador involucra dos niveles de incompletitud, este es el caso de otros métodos de inversas aproximadas considerados anteriormente. Para algunos problemas, esto puede llevarnos a una degradación significativa en la calidad del precondicionador. Quizás , la presencia de dos niveles de incompletitud hace que estos métodos sean difíciles de aplicar en la práctica debido a la necesidad de seleccionar un gran número de parámetros definidos por el usuario. En general, los métodos de ILU inversa son mucho más difíciles de usar que los precondicionadores descritos en las subsecciónes 1.1 y 1.2.

1.3.1. Técnicas de factorización ILU inversa basadas en la expansión de Neumann truncada

Las técnicas de factorización ILU inversa basadas en la expansión de Neumann truncada no son métodos de inversa aproximada estrictamente hablando. Pueden considerarse un híbrido de las técnicas de precondicionamiento ILU y polinomial. Sin embargo son similares a otros métodos ya descritos que se basan en una factorización ILU donde los factores incompletos se invierten de forma inexacta aplicando algún tipo de truncamiento. En particular, la sustitución por descenso y remonte se reemplazan por productos matriz-vector con matrices triangulares *sparse*. Esta idea proviene de van der Vorst [103] y se ha aplicado recientemente a los precondicionadores SSOR que pueden verse

como un tipo de factorización incompleta en Gustafsson y Lindskog [62]. El precondicionador SSOR de Neumann truncado para una matriz simétrica A se define como sigue. Sea $A = E + D + E^T$ la descomposición de A en sus partes estrictamente triangular inferior, diagonal y triangular superior. Considérese el precondicionador SSOR,

$$M = \left(w^{-1}D + E\right)\left(w^{-1}D\right)^{-1}\left(w^{-1}D + E^T\right)$$

donde $0 < w < 2$ es un parámetro de relajación. Sea $L = wED^{-1}$ y $\hat{D} = w^{-1}D$, entonces

$$M = (I + L)\,\hat{D}\,(I + L)^T$$

tal que

$$M^{-1} = (I + L)^{-T}\,\hat{D}^{-1}\,(I + L)^{-1}$$

Note que

$$(I + L)^{-1} = \sum_{k=0}^{n-1} (-1)^k L^k$$

Para algunas matrices, por ejemplo, aquellas con diagonal dominante, $\|L^k\|$ disminuye rápidamente cuando k aumenta y la suma puede aproximarse con un polinomio de grado pequeño en L, esto es, 1 o 2. Por ejemplo, para primer orden,

$$G = \left(I - L^T\right)\hat{D}^{-1}\,(I - L) \approx M^{-1} \approx A^{-1}$$

puede considerarse como una inversa aproximada factorizada de A.

Debido a que el precondicionador SSOR no requiere cálculos, excepto para la estimación de w, posiblemente, este es un precondicionador libre desde el punto de vista virtual. También es muy fácil de implementar. Por otra parte, la efectividad de este método se restringe a problemas para los cuales el precondicionador SSOR estándar trabaja eficientemente, lo cual no ocurre en la generalidad de los problemas. Más aún, la expansión de Neumann truncada a un polinomio de grado pequeño puede provocar una degradación seria de la velocidad de convergencia, particularmente en problemas donde la diagonal no es dominante. La idea de la expansión de Neumann truncada puede también aplicarse a la factorización de Cholesky incompleta y, de forma más general, a los precondicionadores de factorización LU incompleta.

Capítulo 2

Conceptos básicos

2.1. Normas matriciales

2.1.1. Notación

Comenzaremos por revisar algunas ideas del álgebra lineal numérica. Una referencia excelente sobre las ideas básicas tanto del álgebra lineal numérica como de los métodos directos de resolución de sistemas de ecuaciones lineales puede encontrarse en [71, 64, 65].

Escribimos un sistema de ecuaciones lineales como,

$$Ax = b \tag{2.1}$$

donde A es una matriz no singular de orden $n \times n$ y $b \in \mathbb{R}^n$ es conocido y definiremos la solución exacta de (2.1) como

$$x^* = A^{-1}b \in \mathbb{R}^n \tag{2.2}$$

En lo que sigue, denotaremos por x la solución aproximada de la ecuación (2.1) y $\{x_k\}_{k\geq 0}$ será la secuencia de iteraciones. Por otra parte, llamaremos $(x)_i$ a la i-ésima componente de x, y $(x_k)_i$ a la i-ésima componente de x_k, aunque usualmente nos referiremos al vector y no a sus componentes.

La norma en $\mathbb{R}^n$ se simbolizará mediante $\|\cdot\|$ al igual que la norma matricial inducida.

Definición 1 *Sea $\|\cdot\|$ una norma en $\mathbb{R}^n$. La norma matricial inducida de una matriz A de orden $n \times n$ se define como,* $\|A\| = \max_{\|x\|=1} \|Ax\|$.

Las normas inducidas cumplen una importante propiedad,

$$\|Ax\| \leq \|A\| \, \|x\| \tag{2.3}$$

El número de condición de A con respecto a la norma $\|\cdot\|$ es,

$$\kappa(A) = \|A\| \left\|A^{-1}\right\| \tag{2.4}$$

donde $\kappa(A)$ es infinita si A es singular. Si $\|\cdot\|$ es la norma p

$$\|x\|_p = \left(\sum_{j=1}^{N} |(x)_i|^p \right)^{\frac{1}{p}} \tag{2.5}$$

escribiremos el número de condición como κ_p.

La mayoría de los métodos iterativos finalizan cuando el vector residuo,

$$r = b - Ax \tag{2.6}$$

es lo suficientemente pequeño. Otro criterio de parada es,

$$\frac{\|r_k\|}{\|r_0\|} < \tau \tag{2.7}$$

que puede relacionarse con el error mediante,

$$e = x - x^* \tag{2.8}$$

en términos del número de condición.

Lema 2 *Sean $b, x, x_0 \in \mathbb{R}^n$, A una matriz no singular y $x^* = A^{-1}b$, entonces*

$$\frac{\|e\|}{\|e_0\|} \leq \kappa(A) \frac{\|r\|}{\|r_0\|} \tag{2.9}$$

Demostración. Como,

$$r = b - Ax = -Ae$$

tenemos,

$$\|e\| = \|A^{-1}Ae\| \leq \|A^{-1}\| \, \|Ae\| = \|A^{-1}\| \, \|r\|$$

y,

$$\|r_0\| = \|Ae_0\| \leq \|A\| \, \|e_0\|$$

De aquí,

$$\frac{\|e\|}{\|e_0\|} \leq \frac{\|A^{-1}\| \|r\|}{\|A\|^{-1} \|r_0\|} = \kappa(A) \frac{\|r\|}{\|r_0\|}$$

como se afirmó. ■

El criterio de parada (2.7) depende de la iteración inicial y, cuando la iteración inicial es buena, puede resultar un trabajo innecesario, sin embargo si la iteración inicial está muy lejos de la solución inicial el resultado puede ser de calidad insuficiente. Por esta razón es preferible concluir las iteraciones cuando

$$\frac{\|r_k\|}{\|b\|} < \tau \tag{2.10}$$

Las dos condiciones (2.7) y (2.10) son las mismas cuando partimos de $x_0 = 0$, lo cual se efectúa comúnmente, en especial cuando la iteración lineal se usa como parte de un método no lineal.

2.1.2. El Lema de Banach y las inversas aproximadas.

El camino más directo para la solución iterativa de un sistema lineal es reescribir la ecuación (2.1) como una iteración lineal de punto fijo. Una forma es escribir $Ax = b$ como

$$x = (I - A)x + b \tag{2.11}$$

y definir la *iteración de Richardson*

$$x_{k+1} = (I - A)x_k + b \tag{2.12}$$

Mostraremos métodos más generales donde $\{x_k\}$ está dado por

$$x_{k+1} = Mx_k + c \tag{2.13}$$

donde M es una matriz cuadrada de orden n denominada *matriz de iteración*. Los métodos iterativos de esta forma se denominan *métodos iterativos estacionarios* porque la forma de transición de x_k a x_{k+1} no depende del desarrollo de la iteración. Los métodos de Krylov, que se discutirán en la sección 2.2, son métodos iterativos no estacionarios.

Todos estos resultados estan basados en el siguiente Lema,

Lema 3 *Si M es una matriz $n \times n$ con $\|M\| < 1$ entonces $I - M$ es no singular y*

$$\left\|(I - M)^{-1}\right\| \leq \frac{1}{1 - \|M\|} \tag{2.14}$$

Demostración. Probaremos que $I - M$ es no singular y que (2.14) se cumple demostrando que

$$\sum_{l=0}^{\infty} M^l = (I - M)^{-1}$$

Las sumas parciales

$$S_k = \sum_{l=0}^{k} M^l$$

forman una sucesión de Cauchy en $\mathbb{R}^{n\times n}$. Nótese que para todo $m > k$

$$\|S_k - S_m\| \leq \sum_{l=k+1}^{m} \left\|M^l\right\|$$

Ahora, $\left\|M^l\right\| \leq \|M\|^l$ porque $\|\cdot\|$ es una norma matricial inducida por una norma vectorial. De aquí

$$\|S_k - S_m\| \leq \sum_{l=k+1}^{m} \|M\|^l = \|M\|^{k+1} \left(\frac{1-\|M\|^{m-k}}{1-\|M\|}\right) \to 0$$

cuando $m, k \to \infty$. Por tanto, la secuencia S_k converge a S. Como $MS_k + I = S_{k+1}$, tendremos $MS + I = S$ y $(I - M)S = I$. Esto prueba que $I - M$ es no singular y que $S = (I - M)^{-1}$.

Observando que,

$$\|(I - M)^{-1}\| \leq \sum_{l=0}^{\infty} \|M\|^l = (1 - \|M\|)^{-1}$$

se demuestra (2.14) y completa la prueba ■

El siguiente corolario es una consecuencia directa del Lema 3.

Corolario 4 *Si $\|M\| < 1$ entonces la iteración (2.13) converge a $x = (I - M)^{-1}c$ para toda iteración inicial x_0.*

Una consecuencia del corolario 4 es que la iteración de Richardson (2.12) convergerá si $\|I - A\| < 1$. Algunas veces es posible precondicionar el sistema lineal multiplicando ambos miembros de (2.1) por una matriz B

$$BAx = Bb \tag{2.15}$$

de tal forma que se mejore la convergencia del método iterativo. En la sección (2.3) se profundiza en los métodos de precondicionamiento. En el contexto de la iteración de Richardson, las matrices B que permiten aplicar el lema de Banach y su corolario se denominan *inversas aproximadas*..

Definición 5 *B es una inversa aproximada de A si $\|I - BA\| < 1$.*

El siguiente teorema se denomina comúnmente Lema de Banach.

Teorema 6 *Si A y B son dos matrices de orden $n \times n$ y B es una inversa aproximada de A. Entonces, A y B son ambas no singulares y,*

$$\|A^{-1}\| \leq \frac{\|B\|}{1 - \|I - BA\|}, \|B^{-1}\| \leq \frac{\|A\|}{1 - \|I - BA\|}, \tag{2.16}$$

y

$$\|A^{-1} - B\| \leq \frac{\|B\| \, \|I - BA\|}{1 - \|I - BA\|}, \|A - B^{-1}\| \leq \frac{\|A\| \, \|I - BA\|}{1 - \|I - BA\|} \tag{2.17}$$

Demostración. Sea $M = I - BA$. Por el Lema 3, $I - M = I - (I - BA) = BA$ es no singular. Por tanto, ambas matrices A y B son no singulares. Por la ecuación (2.14)

$$\|A^{-1}B^{-1}\| = \|(I - M)^{-1}\| \leq \frac{1}{1 - \|M\|} = \frac{1}{1 - \|I - BA\|} \tag{2.18}$$

como $A^{-1} = (I - M)^{-1} B$, enconces la inecuación (2.18) implica la primera parte de la ecuación (2.16). La segunda parte se sigue de forma similar de $B^{-1} = A(I - M)^{-1}$. Para completar la prueba se debe tener en cuenta que $A^{-1} - B = (I - BA) A^{-1}$, $A - B^{-1} = B^{-1}(I - BA)$ y (2.16). ■

La iteración de Richardson, precondicionada con inversa aproximada tiene la forma,

$$x_{k+1} = (I - BA) x_k + Bb \tag{2.19}$$

Si la norma de $I - BA$ es pequeña, entonces, no solo convergerá la iteración rápidamente sino que, segun indica el Lema 2, el criterio de parada basado en el residuo precondicionado $Bb - BAx$ reflejará mejor el error real. Este método es una técnica muy efectiva para resolver ecuaciones diferenciales, ecuaciones integrales y problemas relacionados. También pueden interpretarse bajo esta luz los métodos multimalla.

2.1.3. El radio espectral

El análisis en la sección antèrior relaciona la convergencia de la ecuación (2.13) con la norma de la matriz A. Sin embargo, la norma de M puede ser pequeña en algunos tipos de normas y muy grande en otros tipos. Aquí la iteración no está completamente descrita por $\|M\|$. El concepto de radio espectral nos permite hacer una descripción completa.

Denotando por $\sigma(A)$ el conjunto de los autovalores de A.

El radio espectral de una matriz A de orden $n \times n$ es

$$\rho(A) = \max_{\lambda \in \sigma(A)} |\lambda| = \lim_{k \to \infty} \left\|A^k\right\|^{\frac{1}{k}} \tag{2.20}$$

El término de la derecha de la segunda igualdad en (2.20) es el límite que resulta al utilizar la caracterización por radicales de convergencia de la serie de potencias $\sum A^k$.

El radio espectral de M es independiente de cualquier norma matricial particular de M, de hecho

$$\rho(A) \leq \|A\| \tag{2.21}$$

para cualquier norma matricial inducida. La inecuación (2.21) concluye que el radio espectral es una cota inferior del conjunto de los valores de las normas matriciales sobre A. Además, se cumple el siguiente teorema.

Teorema 7 *Sea A una matriz de orden $n \times n$. Entonces para cualquier $\epsilon > 0$ existe una norma $\|\cdot\|$ en $\mathbb{R}^n$ tal que*

$$\rho(A) > \|A\| - \epsilon \tag{2.22}$$

En otras palabras, el radio espectral es el ínfimo del conjunto todas las normas matriciales.

Corolario 8 *El menor de los valores singulares de A nunca excede del menor de sus valores propios en módulo.*

Demostración. Denotemos los autovalores y valores singulares de A en orden decreciente,

$$|\lambda_1| \geq |\lambda_2| \geq \ldots \geq |\lambda_n|$$
$$\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_n$$

como

$$\rho(A) = \inf_{\|\cdot\|} \|A\|$$

entonces

$$\rho(A) \leq \|A\|$$

En particular, tomando como norma matricial la norma espectral o de Hilbert,

$$\rho(A) \leq \|A\|_2$$

y

$$|\lambda_1(A)| \leq \left|\lambda_1(AA^T)\right|$$

es decir,

$$|\lambda_1(A)| \leq \sigma_1(A)$$

en particular, cambiando A por A^{-1},

$$|\lambda_1(A^{-1})| \leq \sigma_1(A^{-1})$$

de donde,

$$\sigma_n(A) = \frac{1}{\sigma_n(A^{-1})} \leq \frac{1}{|\lambda_n(A^{-1})|} = |\lambda_n(A)|$$

finalmente,

$$\sigma_n(A) \leq |\lambda_n(A)| \blacksquare$$

Este resultado será de gran utilidad en el Capítulo 4.

2.2. Métodos basados en subespacios de Krylov

La aplicación de diferencias finitas, elementos finitos, elementos de contorno, volúmenes finitos, para la obtención de soluciones aproximadas de problemas de contorno en derivadas parciales, dan como resultado grandes sistemas de ecuaciones lineales (2.1) donde la matriz A es de tipo *sparse*.

Para la resolución de estos sistemas, además de los métodos directos, basados generalmente en la factorización de la matriz A del sistema utilizando la eliminación gaussiana y de los métodos iterativos clásicos (Jacobi, Gauss-Seidel, Relajación,...) que partiendo de una aproximación inicial x_0 generan una secuencia de vectores x_i que converge a la solución buscada, en los últimos años se han desarrollado otros métodos basados en los subespacios de Krylov, que presentan algunas ventajas respecto a los anteriores.

Estos métodos están basados en un proceso de proyección sobre un subespacio de Krylov que es generado por vectores de la forma $p(A)v$. Estas técnicas aproximan $A^{-1}b$ por $p(A)b$ donde $p(A)$ es un polinomio matricial elegido adecuadamente.

2.2.1. Subespacios de Krylov

En los métodos iterativos, las sucesivas aproximaciones del sistema (2.1), vienen dadas por una relación de recurrencia de la forma,

$$x_{k+1} = x_k + B^{-1}(b - Ax_k) \tag{2.23}$$

o bien,

$$Bx_{k+1} = Cx_k + b, \text{ siendo } C = B - A \tag{2.24}$$

La elección de las matrices B y C en función de la matriz A permiten obtener los métodos clásicos de Jacobi, Gauss-Seidel o de Relajación.

Estas expresiones, (2.23) y (2.24), también se pueden escribir en función del vector residuo $r_k = b - Ax_k$ como,

$$x_{k+1} = x_k + B^{-1}r_k \tag{2.25}$$

De esta forma, dada una aproximación inicial x_0, las sucesivas iteraciones se podrían expresar,

$$
\begin{aligned}
x_1 &= x_0 + B^{-1}r_0 \\
x_2 &= x_1 + B^{-1}r_1 = x_0 + B^{-1}r_0 + B^{-1}(b - Ax_1) = \\
&= x_0 + B^{-1}r_0 + B^{-1}(b - Ax_0 - AB^{-1}r_0) = \\
&= x_0 + 2B^{-1}r_0 - B^{-1}A(B^{-1}r_0) \\
x_3 &= x_2 + B^{-1}r_2 \\
x_4 &= x_3 + B^{-1}r_3 \\
&\cdots\cdots\cdots \\
&\cdots\cdots\cdots \\
&\cdots\cdots\cdots \\
x_k &= x_{k-1} + B^{-1}r_{k-1}
\end{aligned}
$$

En el caso en que $B = I$ quedaría,

$$
\begin{aligned}
x_1 &= x_0 + r_0 \\
x_2 &= x_1 + r_1 = x_0 + r_0 + (b - Ax_1) = x_0 + r_0 + (b - Ax_0 - Ar_0) = \\
&= x_0 + 2r_0 - Ar_0 \\
x_3 &= x_2 + r_2 = x_0 + 2r_0 - Ar_0 + (b - Ax_2) = \\
&= x_0 + 2r_0 - Ar_0 + (b - Ax_0 + 2Ar_0 - A^2r_0) = x_0 + 2r_0 + Ar_0 - A^2r_0 \\
x_4 &= x_3 + r_3 \\
&\cdots\cdots\cdots \\
&\cdots\cdots\cdots \\
&\cdots\cdots\cdots \\
x_k &= x_{k-1} + r_{k-1}
\end{aligned}
$$

Es decir que la k-ésima iteración de la solución aproximada se puede expresar como suma de la aproximación inicial y una combinación lineal de k vectores,

$$x_k = x_0 + C.L.\left\{r_0, Ar_0, A^2r_0, ..., A^{k-1}r_0\right\} \tag{2.26}$$

Por tanto,

$$x_k = x_0 + \left[r_0, Ar_0, A^2r_0, ..., A^{k-1}r_0\right] \tag{2.27}$$

El subespacio $\mathcal{K}_k(A; r_0)$ de base $\left[r_0, Ar_0, A^2r_0, ..., A^{k-1}r_0\right]$, es llamado subespacio de Krylov de dimensión k correspondiente a la matriz A y al residuo inicial r_0.

2.2.2. Métodos basados en los subespacios de Krylov

Estos métodos utilizados para la resolución de grandes sistemas lineales, se obtienen para adaptarse, en principio, a dos requerimientos básicos. Por un lado, minimizar una cierta norma del vector residuo sobre un subespacio de Krylov generado por la matriz del sistema y que se traduce en una convergencia

suave sin grandes fluctuaciones, y por otro, ofrecer un bajo coste computacional por iteración y no exigir una capacidad excesiva de almacenamiento.

Sin embargo, esto no es siempre posible. Sólo en sistemas de ecuaciones lineales cuya matriz de coeficientes es simétrica y definida positiva, el algoritmo del Gradiente Conjugado propuesto por Hestenes y Stiefel en 1952 [63] y desarrollado en la práctica a partir de 1970, alcanza teóricamente, salvo errores de redondeo, la solución exacta en un número de iteraciones igual a la dimensión del sistema y verifica los requisitos esenciales de minimalidad y optimalidad anteriores.

Cuando la matriz del sistema no cumple las condiciones de simetría y positividad, el método del Gradiente Conjugado no es aplicable y, en general, no existen métodos que cumplan los dos requisitos anteriores simultáneamente sin añadir inconvenientes y/o desventajas. Por todo ello, los métodos utilizados para estos sistemas, se construyen para adaptarse a uno de ellos, bien al de minimización, o bien al de ofrecer un bajo coste computacional y de almacenamiento.

Los podemos agrupar en tres grandes familias (ver [88]): métodos de ortogonalización, métodos de biortogonalización y los métodos basados en la ecuación normal.

Dentro de estas familias de métodos, hemos considerado para nuestros experimentos numéricos los más utilizados actualmente y que describiremos brevemente: el CGS (*Conjugated Gradient Squared*), el Bi-CGSTAB (*Biconjugate Gradient Stabilized*), el QMRCGSTAB (*Quasi-Minimal Residual Stabilized*) y el GMRES (*Generalized Minimal Residual*).

2.2.2.1. Método CGS (Conjugated Gradient Squared)

Presentamos el método CGS como punto de partida para entender la construcción del algoritmo Bi-CGSTAB. Desarrollado por Sonnelved en 1989 [95], es una modificación del Doble Gradiente Conjugado (Bi-CG). Sonnenveld observó que, en caso de convergencia del Bi-CG, la convergencia de los "residuos auxiliares" no es explotada y sólo se calculan para la valoración de los parámetros que aparecen en el algoritmo. Propone, entonces, la siguiente modificación donde todos los esfuerzos se concentran en la convergencia de los vectores del sistema original r_i.

Consideremos las siguientes expresiones polinómicas para los vectores residuo de la iteración j-ésima r_j, y la correspondiente dirección conjugada p_j,

$$r_j = \phi_j(A)r_0 \tag{2.28}$$

$$p_j = \pi_j(A)r_0 \tag{2.29}$$

donde ϕ_j es un cierto polinomio matricial de grado j que satisface la restricción $\phi_j(0) = I$, y π_j es igualmente otro polinomio matricial de grado j.

De forma similar, y teniendo en cuenta que los vectores r_j^* y p_j^* del sistema auxiliar en el algoritmo Bi-CG se obtuvieron a la vez usando las mismas fórmulas de recurrencia que r_j y p_j, simplemente reemplazando la matriz A por A^T, entonces,

$$r_j^* = \phi_j(A^T)r_0^*,\ p_j^* = \pi_j(A^T)r_0^* \tag{2.30}$$

Por tanto, el escalar α_j del algoritmo Bi-CG se obtendrá,

$$\alpha_j = \frac{\langle \phi_j(A)r_0, \phi_j(A^T)r_0^* \rangle}{\langle A\pi_j(A)r_0, \pi_j(A^T)r_0^* \rangle} = \frac{\langle \phi_j^2(A)r_0, r_0^* \rangle}{\langle A\pi_j^2(A)r_0, r_0^* \rangle} \tag{2.31}$$

Sonneveld, sugiere trabajar con aproximaciones $\widehat{x}_j$, que satisfacen,

$$\widehat{r}_j = b - A\widehat{x}_j = \phi_j^2(A)r_0 \tag{2.32}$$

Como la expresión del vector residuo del sistema original en el algoritmo Bi-CG viene dada por $r_{j+1} = r_j - \alpha_j A p_j$, entonces sustituyendo,

$$\phi_{j+1}(A)r_0 = \phi_j(A)r_0 - \alpha_j A\pi_j(A)r_0 \tag{2.33}$$

y elevando al cuadrado,

$$\phi_{j+1}^2(A) = \phi_j^2(A) - \alpha_j A\left[2\phi_j(A)\pi_j(A) - \alpha_j A\pi_j^2(A)\right] \tag{2.34}$$

Para calcular por recurrencia las direcciones conjugadas, como,

$$\pi_{j+1}(A) = \phi_{j+1}(A) + \beta_j\pi_j(A) \tag{2.35}$$

elevando al cuadrado

$$\pi_{j+1}^2(A) = \phi_{j+1}^2(A) + 2\beta_j\phi_{j+1}(A)\pi_j(A) + \beta_j^2\pi_j(A)^2 \tag{2.36}$$

De (2.35) se tiene que,

$$\phi_j(A)\pi_j(A) = \phi_j^2(A) + \beta_{j-1}\phi_j(A)\pi_{j-1}(A) \tag{2.37}$$

y de forma similar,

$$\phi_{j+1}(A)\pi_j(A) = \phi_j^2(A) + \beta_{j-1}\phi_j(A)\pi_{j-1}(A) - \alpha_j A\pi_j^2(A) \tag{2.38}$$

Si definimos,

$$r_j = \phi_j^2(A)r_0 \tag{2.39}$$

$$p_j = \pi_j^2(A)r_0 \tag{2.40}$$

$$q_j = \phi_{j+1}(A)\pi_j(A)r_0 \tag{2.41}$$

Transformando estas recurrencias de polinomios, obtenemos,

$$r_{j+1} = r_j - \alpha_j A\left[2r_j + 2\beta_{j-1}q_{j-1} - \alpha_j A p_j\right] \tag{2.42}$$

$$q_j = r_j + \beta_{j-1}q_{j-1} - \alpha_j A p_j \tag{2.43}$$

$$p_{j+1} = r_{j+1} + 2\beta_j q_j + \beta_j^2 p_j \tag{2.44}$$

Definiremos los vectores auxiliares,

$$d_j = 2r_j + 2\beta_{j-1}q_{j-1} - \alpha_j A p_j \tag{2.45}$$

$$u_j = r_j + \beta_{j-1}q_{j-1} \tag{2.46}$$

Con estas relaciones resulta el siguiente algoritmo en el que no figuran los vectores residuos ni las direcciones conjugadas del sistema auxiliar, eliminando de esta forma los productos A^T por vector.

ALGORITMO CGS

Aproximación inicial x_0. $r_0 = b - Ax_0$; r_0^* arbitrario tal que $r_0^T r_0^* \neq 0$

Fijar $p_0 = u_0 = r_0$

Desde $j = 1, 2, \ldots$ hasta converger, hacer

$$\alpha_j = \frac{\langle r_j, r_o^* \rangle}{\langle Ap_j, r_o^* \rangle}$$
$$q_j = u_j - \alpha_j A p_j$$
$$x_{j+1} = x_j + \alpha_j (u_j + q_j)$$
$$r_{j+1} = r_j - \alpha_j A (u_j + q_j)$$
$$\beta_j = \frac{\langle r_{j+1}, r_0^* \rangle}{\langle r_j, r_0^* \rangle}$$
$$u_{j+1} = r_{j+1} + \beta_j q_j$$
$$p_{j+1} = u_{j+1} + \beta_j (q_j + \beta_j p_j)$$

Fin

2.2.2.2. Método Bi-CGSTAB (Biconjugated Gradient Stabilized)

En el CGS, los vectores residuos verifican $\widehat{r}_j = \phi_j^2(A) r_0$. Van der Vorst propone obtener un vector residuo en el Bi-CGSTAB [104] mediante la aplicación sucesiva de dos polinomios reductores distintos de la forma,

$$r_j' = \psi_j(A)\phi_j(A) r_0 \tag{2.47}$$

tal que las relaciones de recurrencia del algoritmo donde intervenga este polinomio $\psi_j(A)$ no deben ser excesivamente complicadas y los parámetros que figuren en su definición sean fácilmente optimizables. En este sentido sugiere para $\psi_j(A)$ la expresión,

$$\psi_{j+1}(A) = (I - w_j A)\psi_j(A) \tag{2.48}$$

determinando w_j, por la condición de mínimo para r_j en la iteración j-ésima.

Las relaciones de recurrencia se derivan de forma similar que en el algoritmo CGS. Así,

$$\psi_{j+1}(A)\phi_{j+1}(A) = (I - w_j A)\psi_j(A)\phi_{j+1}(A) = \tag{2.49}$$

$$= (I - w_j A)\left[\psi_j(A)\phi_j(A) - \alpha_j A\psi_j(A)\pi_j(A)\right] \tag{2.50}$$

Necesitamos una relación de recurrencia para la correspondiente dirección conjugada. Esto es,

$$\psi_j(A)\pi_j(A) = \psi_j(A)\left[\phi_j(A) + \beta_{j-1}\pi_{j-1}(A)\right] = \tag{2.51}$$

$$= \psi_j(A)\phi_j(A) + \beta_{j-1}(I - w_{j-1}A)\psi_{j-1}(A)\pi_{j-1}(A) \tag{2.52}$$

Definiendo,

$$r_j = \psi_j(A)\phi_j(A)r_0,$$

$$p_j = \psi_j(A)\pi_j(A)r_0$$

y teniendo en cuenta las anteriores relaciones, podemos encontrar las correspondientes fórmulas de recurrencia en función de los escalares α_j y β_j,

$$r_{j+1} = (I - w_j A)(r_j - \alpha_j A p_j) \tag{2.53}$$

$$p_{j+1} = r_{j+1} + \beta_j(I - w_j A)p_j \tag{2.54}$$

Para obtener el algoritmo se toma como referencia, al igual que en el CGS, el algoritmo Bi-CG, efectuando las transformaciones necesarias para que las relaciones de recurrencia de la solución sean función del nuevo vector residuo. Recordemos que en el algoritmo BCG $\beta_j = \rho_{j+1}/\rho_j$ con,

$$\rho_j = \langle \phi_j(A)r_0, \phi_j(A^T)r_0^* \rangle = \langle \phi_j^2(A)r_0, r_0^* \rangle$$

Pero como ρ_j no lo hallamos porque ninguno de los vectores $\phi_j(A)r_0, \phi_j(A^T)r_0^*$ ó $\phi_j^2(A)r_0$ están disponibles, lo podemos relacionar con el escalar,

$$\widetilde{\rho}_j = \langle \phi_j(A)r_0, \psi_j(A^T)r_0^* \rangle$$

que podemos obtener como,

$$\widetilde{\rho}_j = \langle \psi_j(A)\phi_j(A)r_0, r_0^* \rangle = \langle r_j, r_0^* \rangle$$

Desarrollando $\phi_j(A^T)r_0^*$ explicitamente para relacionar los escalares, ρ_j y $\widetilde{\rho}_j$,

$$\widetilde{\rho}_j = \langle \phi_j(A)r_0, \eta_1^j(A^T)^j r_0^* + \eta_2^j(A^T)^{j-1}r_0^* + \ldots \rangle$$

incorporando la ortogonalidad de $\phi_j(A)r_0$ respecto de todos los vectores $(A^T)^i r_0^*$, con $i < j$ y teniendo en cuenta que sólo los coeficientes principales de los polinomios son relevantes en el desarrollo del anterior producto, si γ_1^j es el principal coeficiente para el polinomio $\phi_j(A)$, entonces,

$$\widetilde{\rho}_j = \left\langle \phi_j(A)r_0, \frac{\eta_1^j}{\gamma_1^j}\phi_j(A^T)r_0^* \right\rangle = \frac{\eta_1^j}{\gamma_1^j}\rho_j$$

Examinando las relaciones de recurrencia para $\phi_{j+1}(A)$ y $\psi_{j+1}(A)$, los coeficientes principales para estos polinomios fueron hallados para satisfacer las relaciones,

$$\eta_1^{j+1} = -w_j\eta_1^j, \quad \gamma_1^{j+1} = -\alpha_j\gamma_1^j$$

Por tanto,

$$\frac{\widetilde{\rho}_{j+1}}{\widetilde{\rho}_j} = \frac{w_j}{\alpha_j}\frac{\rho_{j+1}}{\rho_j}$$

que nos permite encontrar la siguiente relación para β_j,

$$\beta_j = \frac{\widetilde{\rho}_{j+1}}{\widetilde{\rho}_j}\frac{\alpha_j}{w_j}$$

De forma similar, y por una simple fórmula de recurrencia podemos encontrar α_j como,

$$\alpha_j = \frac{\langle \phi_j(A)r_0, \phi_j(A^T)r_0^* \rangle}{\langle A\pi_j(A)r_0, \pi_j(A^T)r_0^* \rangle}$$

De la misma manera que anteriormente, en los productos de polinomios, tanto en el numerador como en el denominador, sólo se consideran los términos correpondientes a sus respectivos coeficientes principales y como éstos son idénticos para $\phi_j(A^T)r_0^*$ y $\pi_j(A^T)r_0^*$, podemos escribir,

$$\begin{aligned}\alpha_j &= \frac{\langle \phi_j(A)r_0, \phi_j(A^T)r_0^* \rangle}{\langle A\pi_j(A)r_0, \phi_j(A^T)r_0^* \rangle} \\ &= \frac{\langle \phi_j(A)r_0, \psi_j(A^T)r_0^* \rangle}{\langle A\pi_j(A)r_0, \psi_j(A^T)r_0^* \rangle} \\ &= \frac{\langle \psi_j(A)\phi_j(A)r_0, r_0^* \rangle}{\langle A\psi_j(A)\pi_j(A)r_0, r_0^* \rangle}\end{aligned}$$

Y como $p_j = \psi_j(A)\pi_j(A)r_0$, entonces,

$$\alpha_j = \frac{\widetilde{\rho}_j}{\langle Ap_j, r_0^* \rangle} \tag{2.55}$$

A partir de la ecuación (2.53), si hacemos,

$$s_j = r_j - \alpha_j Ap_j \tag{2.56}$$

el valor óptimo para el parámetro w_j que interviene en la construcción del polinomio reductor $\psi_j(A)$ y que figura en las relaciones de recurrencia del algoritmo lo obtendremos con la condición de minimizar la norma del vector residuo,

$$r_{j+1} = s_j - w_j A s_j$$

$$\|r_{j+1}\|^2 = \langle s_j - w_j A s_j, s_j - w_j A s_j \rangle = \langle s_j, s_j \rangle - 2w_j \langle s_j, A s_j \rangle + w_j^2 \langle A s_j, A s_j \rangle$$

$$\frac{\partial \|r_{j+1}\|^2}{\partial w_j} = -2 \langle s_j, A s_j \rangle + 2w_j \langle A s_j, A s_j \rangle = 0$$

con lo que,

$$w_j = \frac{\langle A s_j, s_j \rangle}{\langle A s_j, A s_j \rangle} \tag{2.57}$$

La ecuación (2.53) ahora resulta,

$$r_{j+1} = s_j - w_j A s_j = r_j - \alpha_j A p_j - w_j A s_j \tag{2.58}$$

y la relación de recurrencia para el vector solución viene dada por,

$$x_{j+1} = x_j + \alpha_j p_j + w_j s_j \tag{2.59}$$

Una vez expresados todos los vectores en función del nuevo residuo y determinadas sus relaciones de recurrencia, así como los escalares que intervienen en las mismas, se puede escribir el algoritmo.

ALGORITMO BI-CGSTAB

Aproximación inicial x_0. $r_0 = b - A x_0$; r_0^* arbitrario.

Fijar $p_0 = r_0$

Desde $j = 1, 2, \ldots$ hasta converger, hacer

$$\alpha_j = \frac{\langle r_j, r_o^* \rangle}{\langle A p_j, r_o^* \rangle}$$
$$s_j = r_j - \alpha_j A p_j$$
$$w_j = \frac{\langle A s_j, s_j \rangle}{\langle A s_j, A s_j \rangle}$$
$$x_{j+1} = x_j + \alpha_j p_j + w_j s_j$$
$$r_{j+1} = s_j - w_j A s_j$$
$$\beta_j = \frac{\langle r_{j+1}, r_0^* \rangle}{\langle r_j, r_0^* \rangle} \frac{\alpha_j}{w_j}$$
$$p_{j+1} = r_{j+1} + \beta_j (p_j - w_j A p_j)$$

Fin

En el algoritmo figuran dos productos matriz por vector y cuatro productos escalares, mientras que el CGS exige los mismos productos matriz por vector y sólo dos productos escalares. Sin embargo en la mayoría de los casos la convergencia del Bi-CGSTAB es más rápida y uniforme, necesitando menor carga computacional para alcanzar una determinada tolerancia, pues la reducción del número de iteraciones compensa su mayor coste.

2.2.2.3. Método de QMRCGSTAB

Desarrollado por Chan y otros [31] está basado en la aplicación del principio de minimización usado en el algoritmo Bi-CGSTAB al método QMR, de la misma forma que el TFQMR es derivado del CGS.

Sea,

$$Y_k = [y_1, y_2, \ldots y_k] \ ,$$
$$W_{k+1} = [w_0, w_1, \ldots w_k]$$

tal que,

$$\begin{cases} y_{2l-1} = p_l \ \text{ para } \ l = 1, \ldots, [(k+1)/2] \\ y_{2l} = s_l \ \text{ para } l = 1, \ldots, [k/2] \end{cases}$$

y,

$$\begin{cases} w_{2l-1} = s_l \ \text{ para } \ l = 1, \ldots, [(k+1)/2] \\ w_{2l} = r_l \ \text{ para } \ l = 0, 1, \ldots, [k/2] \end{cases}$$

donde $[k/2]$ y $[(k+1)/2]$ son la parte entera de $k/2$ y $(k+1)/2$ respectivamente.

Definiendo $[\delta_1, \delta_2, \ldots \delta_k]$ tal que,

$$\begin{cases} \delta_{2l} = \omega_l \ \text{ para } \ l = 1, \ldots, [(k+1)/2] \\ \delta_{2l-1} = \alpha_l \ \text{ para } \ l = 1, \ldots, [(k+1)/2] \end{cases}$$

Entonces, para cada columna de W_{k+1} y Y_k, las expresiones (2.56) y (2.58) pueden escribirse como,

$$Ay_m = (w_{m-1} - w_m)\, \delta_m^{-1}, \qquad m = 1, \ldots, k \tag{2.60}$$

o usando notación matricial,

$$AY_k = W_{k+1} E_{k+1}$$

donde E_{k+1} es una matriz bidiagonal $(k+1) \times k$ con elementos diagonales δ_m^{-1} y en la diagonal inferior $-\delta_m^{-1}$.

Esto puede ser fácilmente comprobado hasta que el grado de los polinomios correspondientes a los vectores r_j, s_j y p_j sean $2j$, $2j-1$, y $2j-2$, respectivamente. Entonces, Y_k y W_k generarán el mismo subespacio de Krylov generado por r_0 pero de grado $k-1$.

La idea principal del QMRCGSTAB es encontrar una aproximación a la solución del sistema (2.1) usando el subespacio de Krylov $\mathcal{K}_{k-1}$ en la forma,

$$x_k = x_0 + Y_k g_k \qquad \text{con} \qquad g_k \in \mathbb{R}^n$$

La expresión para el vector residuo queda,

$$r_k = r_0 - AY_k g_k = r_0 - W_{k+1} E_{k+1} g_k$$

Teniendo en cuenta el hecho de que el primer vector de W_{k+1} es justamente r_0, entonces,

$$r_k = W_{k+1}(e_1 - E_{k+1}g_k)$$

Como las columnas de W_{k+1} no están normalizadas, se usa una matriz de escalado $\Sigma_{k+1} = diag(\sigma_1, \ldots, \sigma_{k+1})$ con $\sigma_j = \|w_j\|$ para hacer unitarias las columnas de W_{k+1}. Entonces,

$$r_k = W_{k+1}\Sigma_{k+1}^{-1}\Sigma_{k+1}(e_1 - E_{k+1}g_k) = W_{k+1}\Sigma_{k+1}^{-1}(\sigma_1 e_1 - H_{k+1}g_k)$$

con $H_{k+1} = \Sigma_{k+1}E_{k+1}$.

La aproximación QMR consiste en la minimización de $\|\sigma_1 e_1 - H_{k+1}g_k\|$ para algún $g \in \mathbb{R}^k$, donde este problema de mínimos cuadrados es resuelto usando la descomposición QR de la matriz H_{k+1} de forma incremental utilizando las rotaciones de Givens. Como H_{k+1} es bidiagonal inferior, solamente es necesaria la rotación en los pasos previos.

ALGORITMO QMRCGSTAB

Aproximación inicial x_0. ;
$r_0 = b - Ax_0$;
Elegir $\widetilde{r}_0$ tal que $\rho_0 = \widetilde{r}_0^T r_0 \neq 0$
$p_0 = v_0 = d_0 = 0$;
$\rho_0 = \alpha_0 = \omega_0 = 1$; $\tau = \|r_0\|$, $\theta_0 = 0$, $\eta_0 = 0$;
Desde $k = 1, 2, \ldots$, hacer:
$\rho_k = \widetilde{r}_0^T r_{k-1}$; $\beta_k = (\rho_k \alpha_{k-1})/\rho_{k-1}\omega_{k-1}$;
$p_k = r_{k-1} + \beta_k(p_{k-1} - \omega_k v_{k-1})$;
$v_k = Ap_k$;
$\alpha_k = \rho_k / \widetilde{r}_0^T v_k$;
$s_k = r_{k-1} - \alpha_k v_k$;
Primera cuasi-minimización
$\widetilde{\theta}_k = \|s_k\|/\tau$; $c = \dfrac{1}{\sqrt{1+\widetilde{\theta}_k^2}}$; $\widetilde{\tau} = \tau\,\widetilde{\theta}_k c$;
$\widetilde{\eta}_k = c^2\alpha_k$;
$\widetilde{d}_k = p_k + \dfrac{\theta_{k-1}^2 \eta_{k-1}}{\alpha_k} d_{k-1}$;
$\widetilde{x}_k = x_{k-1} + \widetilde{\eta}_k \widetilde{d}_k$;
Hallar:
$t_k = As_k$;
$\omega_k = \dfrac{\langle s_k, t_k\rangle}{\langle t_k, t_k\rangle}$;
$r_k = s_k - \omega_k t_k$;
Segunda cuasi-minimización
$\theta_k = \|r_k\|/\tau$; $c = \dfrac{1}{\sqrt{1+\theta_k^2}}$; $\tau = \widetilde{\tau}\,\theta_k c$;

$\eta_k = c^2\omega_k;$
$d_k = s_k + \frac{\widetilde{\theta}_k^2\widetilde{\eta}_k}{\omega_k}\widetilde{d}_k;$
$x_k = \widetilde{x}_k + \eta_k d_k;$
Si x_k converge parar
Fin

2.2.3. Método GMRES (*Generalized Minimal Residual*)

El GMRES [92, 93, 94] es un método de proyección sobre un subespacio de Krylov $\mathcal{K}_k$ de dimensión k, basado en minimizar la norma del residuo.

Así, el desarrollo del algoritmo consiste en encontrar un vector x de $x_0 + \mathcal{K}_k$ tal que,

$$x = x_0 + V_k y$$

Imponiendo la condición de mínimo para,

$$J(y) = \|b - Ax\| \tag{2.61}$$

como,

$$b - Ax = b - A\left(x_0 + V_k y\right) = r_0 - AV_k y$$

y teniendo en cuenta que

$$AV_k = V_k H_k + w_k e_k^T = V_{k+1}\overline{H}_k \tag{2.62}$$

y que $v_1 = r_0/\parallel r_0 \parallel$, llamando $\beta = \parallel r_0 \parallel$, entonces,

$$b - Ax = \beta v_1 - V_{k+1}\overline{H}_k y$$

Pero, $v_1 = V_{k+1}e_1$, con $e_1 \in \mathbb{R}^{k+1}$, por tanto,

$$b - Ax = V_{k+1}\left(\beta e_1 - \overline{H}_k y\right) \tag{2.63}$$

y la ecuación (2.61) quedará,

$$J(y) = \left\|V_{k+1}\left(\beta e_1 - \overline{H}_k y\right)\right\|$$

Como las columnas de la matriz V_{k+1} son ortonormales por construcción, podemos simplificar la expresión anterior,

$$J(y) = \left\|\left(\beta e_1 - \overline{H}_k y\right)\right\| \tag{2.64}$$

El siguiente algoritmo del GMRES busca el único vector de $x_0 + \mathcal{K}_k$ que minimiza la funcional $J(y)$.

ALGORITMO GMRES
Aproximación inicial x_0. $r_0 = b - Ax_0$;

Definir la $(k+1) \times k$ matriz $\overline{H}_k = \{H\}_{1 \leq i \leq k+1, 1 \leq j \leq k}$. Poner $\overline{H}_k = 0$.
Desde $j = 1, ..., k$ hacer
$w_j = Av_j$
Desde $i = 1, ..., j$ hacer
$\{H\}_{ij} = \langle w_j, v_i \rangle$;
$w_j = w_j - \{H\}_{ij} v_i$;
Fin
$\{H\}_{j+1,j} = \| w_j \|$; Si $\{H\}_{j+1,j} = 0$ poner $k = j$ y parar
$v_{j+1} = \frac{1}{\{H\}_{j+1,j}} w_j$;
Fin
Hallar y_k que minimiza $\| (\beta e_1 - \overline{H}_k y) \|$;
Determinar $x_k = x_0 + V_k y_k$ siendo $V_k = [v_1, v_2, ..., v_k]$;
Calcular $r_k = b - Ax_k$

El algoritmo GMRES, resulta impracticable cuando k es muy grande, ya que conlleva un elevado coste computacional y de almacenamiento. Por esta razón, se suele usar la técnica de *restart* y de truncamiento.

ALGORITMO *Restarted* GMRES
1) Aproximación inicial x_0. $r_0 = b - Ax_0$, $\beta = \| r_0 \|$, y $v_1 = r_0/\beta$;
2) Generar la base de Arnoldi y la matriz $\overline{H}_k$ usando el algoritmo de Arnoldi;
3) Iniciar con v_1;
4) Hallar y_k que minimiza $\| (\beta e_1 - \overline{H}_k y) \|$ y $x_k = x_0 + V_k y_k$;
5) Si se satisface entonces parar. En caso contrario poner $x_0 := x_k$ y volver al paso 1.

2.3. Precondicionamiento

La convergencia de los métodos basados en los subespacios de Krylov mejora con el uso de las técnicas de precondicionamiento. Estas consisten generalmente en cambiar el sistema original (2.1) por otro de idéntica solución, de forma que el número de condicionamiento de la matriz del nuevo sistema sea menor que el de A, o bien que tenga una mejor distribución de autovalores. Para efectuar el precondicionamiento, se introduce una matriz M, llamada matriz de precondicionamiento. Para ello multiplicaremos ambos miembros del sistema (2.1) por la matriz M,

$$MAx = Mb \tag{2.65}$$

tal que,

$$\kappa(MA) < \kappa(A)$$

El menor valor de $\kappa(A)$ corresponde a $M = A^{-1}$, de forma que $\kappa(AA^{-1}) = 1$, que es el caso ideal, para el cual el sistema convergería en una sola iteración, pero el coste computacional del cálculo de A^{-1} equivaldría a resolver el sistema por un método directo. Por ello se sugiere que M sea una matriz lo más próxima a A^{-1} sin que su determinación suponga un coste elevado.

Generalmente, se considera como matriz de precondicionamiento a M^{-1} y obtener M como una aproximación de A, esto es,

$$M^{-1}Ax = M^{-1}b$$

Por tanto, la matriz M debe ser fácilmente invertible para poder efectuar los productos M^{-1} por vector que aparecen en los algoritmos precondicionados sin excesivo coste adicional. Por ejemplo, en el caso en que M es una matriz diagonal o está factorizada adecuadamente, se pueden efectuar dichos productos mediante remonte sin necesidad de calcular M^{-1}.

Dependiendo de la forma de plantear el producto de M^{-1} por la matriz del sistema obtendremos distintas formas de precondicionamiento. Estas son,

$$\begin{array}{ll} M^{-1}Ax = M^{-1}b & \text{(Precondicionamiento por la izquierda)} \\ AM^{-1}Mx = b & \text{(Precondicionamiento por la derecha)} \\ M_1^{-1}AM_2^{-1}M_2x = M^{-1}b & \text{(Precondicionamiento por ambos lados)} \end{array} \tag{2.66}$$

si M puede ser factorizada como $M = M_1M_2$.

Los precondicionadores pues, han de cumplir dos requisitos fundamentales, fácil implementación, evitando un coste computacional excesivo del producto de M^{-1} por cualquier vector y mejorar la convergencia del método. Por tanto una matriz que sea una aproximación más o menos cercana de A, obtenida con estos criterios puede dar lugar a un buen precondicionador. El campo de posibles precondicionadores es, así, muy amplio. Algunos de los más usados son los bien conocidos Diagonal o de Jacobi, SSOR e ILU. Consideraremos estos, además de la Inversa Aproximada de estructura Diagonal que hemos denomimado Diagonal Óptimo, para comparar con los precondicionadores basados en la inversa aproximada de estructura *sparse* general propuestos en esta tesis.

2.3.1. Precondicionador diagonal

Surge comparando la fórmula de recurrencia para la solución que resulta de aplicar el método de Richardson, cuya relación de recurrencia viene dada por $x_{i+1} = x_i + \alpha(b - Ax_i)$, con $\alpha > 0$, al sistema precondicionado con la fórmula correspondiente que se obtiene aplicando el método de Jacobi al sistema sin precondicionar.

De la aplicación del método de Richardson al sistema precondicionado,

$$M^{-1}Ax = M^{-1}b \tag{2.67}$$

se obtiene, para el cálculo de los sucesivos valores de la solución,

$$x_{i+1} = x_i + \alpha \left(M^{-1} b - M^{-1} A x_i \right)$$

Multiplicando por la matriz de precondicionamiento M y haciendo $\alpha = 1$, resulta

$$M x_{i+1} = M x_i + \alpha \left(b - A x_i \right) \tag{2.68}$$

Por otro lado, descomponiendo la matriz del sistema en $A = D - E - F$, (siendo D la matriz diagonal formada por los elementos de la diagonal de A y E y F matrices triangulares inferior y superior, respectivamente), y utilizando el método de Jacobi para la resolución del sistema, se obtiene,

$$D x_{i+1} = D x_i + \left(b - A x_i \right) \tag{2.69}$$

Comparando las expresiones de recurrencia finales de ambos métodos, se observa que el método de Jacobi aplicado al sistema sin precondicionar, equivale al de Richardson, menos robusto y más simple, cuando este se aplica al sistema precondicionado con la matriz diagonal D. Resulta así un precondicionador elemental, fácil de implementar y con matriz inversa que se determina con muy bajo coste computacional, ya que la matriz de precondicionamiento es diagonal y sus entradas son las de la diagonal de A.

2.3.2. Precondicionador SSOR

Si aplicamos el método SSOR al sistema sin precondicionar, considerando la descomposición de la matriz A en $A = D - E - F$ como en el caso anterior siendo ω el parámetro de relajación, se obtiene para la solución,

$$\begin{aligned} \mathbf{x}_{i+1} = & \left(\frac{D}{\omega} - F \right)^{-1} \left(\frac{1-\omega}{\omega} D + E \right) \left(\frac{D}{\omega} - E \right)^{-1} x_i + \\ & + \left(\frac{D}{\omega} - F \right)^{-1} \frac{2-\omega}{\omega} \left(\frac{D}{\omega} - E \right)^{-1} b \end{aligned}$$

operando para expresar esta relación de forma que se pueda comparar con la solución que resulta de aplicar el método de Richardson al sistema precondicionado,

$$\begin{aligned} & \frac{1}{\omega (2-\omega)} (D - \omega E) D^{-1} (D - \omega F) x_{i+1} \\ & = \frac{1}{\omega (2-\omega)} (D - \omega E) D^{-1} (D - \omega F) x_i + (b - A x_i) \end{aligned}$$

con lo que resulta como matriz de precondicionamiento,

$$M = \frac{1}{\omega (2-\omega)} (D - \omega E) D^{-1} (D - \omega F) \tag{2.70}$$

que en el caso de sistemas simétricos, como se cumple que,

$$(D - \omega F) = (D - \omega E)^T$$

podemos expresarla como un producto de dos matrices triangulares traspuestas,

$$\mathrm{M} = \left[\frac{(D - \omega E)\, D^{-1/2}}{\sqrt{\omega\,(2-\omega)}}\right]\left[\frac{(D - \omega E)\, D^{-1/2}}{\sqrt{\omega\,(2-\omega)}}\right]^T \tag{2.71}$$

para el caso de sistemas no simétricos también lo podremos expresar como un producto de matrices triangulares, inferior y superior respectivamente,

$$M = \left(I - \omega E D^{-1}\right)\left(\frac{D - \omega F}{\omega(2-\omega)}\right) \tag{2.72}$$

2.3.3. Precondicionador ILU(0)

Resulta de la aproximación de A por una factorización incompleta LU, guardando las misma entradas nulas en las matrices triangulares L y U [81],

$$A = LU \approx ILU(0) = M \tag{2.73}$$

donde las entradas de M, m_{ij}, son tales que,

$$m_{ij} = 0 \quad \text{if} \quad a_{ij} = 0 \tag{2.74}$$

$$\{A - LU\}_{ij} = 0 \quad \text{if} \quad a_{ij} \neq 0 \tag{2.75}$$

Es decir que lo elementos nulos de la matriz del sistema siguen siendo nulos en las posiciones respectivas de las matrices triangulares para no incrementar el coste computacional.

2.3.4. Precondicionador Diagonal Óptimo

Resulta de resolver un problema de minimización [8],

$$\min_{M \in S} \|MA - I\|_F = \|NA - I\|_F$$

donde S es el subespacio de las matrices diagonales de orden n.

La solución a este problema, que puede verse en [56], es:

$$N = diag\left(\frac{a_{11}}{\|e_1^T A\|_2^2}, \frac{a_{22}}{\|e_2^T A\|_2^2}, \ldots, \frac{a_{nn}}{\|e_n^T A\|_2^2}\right) \tag{2.76}$$

$$\|NA - I\|_F^2 = n - \sum_{i=1}^{n} \frac{a_{ii}}{\|e_i^T A\|_2^2} \tag{2.77}$$

2.3.5. Algunos métodos de Krylov precondicionados

2.3.5.1. Algoritmo Bi-CGSTAB

El método BICGSTAB introduce un nuevo parámetro $\widetilde{\omega}$ en cada iteración para minimizar el residuo (ver [104]). Para cada forma de precondicionamiento, el valor del parámetro $\widetilde{\omega}$ resulta:

$$\begin{aligned}
\widetilde{\omega} &= \frac{(As)^T s}{(As)^T As} && \text{precondicionamiento por la derecha} \\
\widetilde{\omega} &= \frac{(M^{-1}As)^T (M^{-1}s)}{(M^{-1}As)^T (M^{-1}As)} && \text{precondicionamiento por la izquierda} \\
\widetilde{\omega} &= \frac{(L^{-1}As)^T (L^{-1}s)}{(L^{-1}As)^T (L^{-1}As)} && \text{precondicionamiento por ambos lados}
\end{aligned} \tag{2.78}$$

De este modo, el cálculo de $\widetilde{\omega}$ incluye un proceso de sustitución por iteración para el precondicionamiento por la izquierda y dos para el precondicionamiento por ambos lados. No obstante, si obtenemos $\widetilde{\omega}$ a partir de la minimización del residuo del sistema original sin precondicionar, todos los valores dados en la ecuación (2.78) coinciden con el del precondicionamiento por la derecha. En este caso se obtiene también un único algoritmo,

ALGORITMO BICGSTAB PRECONDICIONADO
Aproximación inicial x_0. $r_0 = b - Ax_0$;
Elegir r_0^* arbitrario, $p_0 = r_0$
Desde $j = 1, 2, \ldots$ hasta converger, hacer
Resolver $Mz_j = r_j$
$y_j = Ap_j$
Resolver $Mv_j = y_j$
$\widetilde{\alpha}_j = \dfrac{\langle z_j, r_0^* \rangle}{\langle v_j, r_0^* \rangle}$
$s_j = r_j - \widetilde{\alpha}_j y_j$
$u_j = z_j - \widetilde{\alpha}_j v_j$
$t_j = Au_j$
$\widetilde{\omega}_j = \dfrac{\langle t_j, s_j \rangle}{\langle t_j, t_j \rangle}$
$x_{j+1} = x_j + \widetilde{\alpha}_j p_j + \widetilde{\omega}_j u_j$
$r_{j+1} = s_j - \widetilde{\omega}_j t_j$
$\widetilde{\beta}_j = \dfrac{\langle z_{j+1}, r_0^* \rangle}{\langle z_j, r_0^* \rangle} \times \dfrac{\widetilde{\alpha}_j}{\widetilde{\omega}_j}$
$p_{j+1} = z_{j+1} + \widetilde{\beta}_j (p_j - \widetilde{\omega}_j v_j)$
Fin

Cualquier otra elección del residuo inicial conducirá a una nueva forma de precondicionamiento (ver [98]).

2.3.5.2. Algoritmo QMRCGSTAB

Este método aplica el principio de cuasi-minimización al BICGSTAG. La minimización por mínimos cuadrados incluida en el proceso es resuelta usando la descomposición QR de la matriz de Hessemberg de forma incremental con las rotaciones de Givens. En el algoritmo QMRCGSTAB dado a continuación, las rotaciones de Givens son escritas explicitamente como propone originalmente Chan y otros [31].

ALGORITMO QMRCGSTAB PRECONDICIONADO
Aproximación inicial x_0. $r_0 = b - Ax_0$;
Elegir $\widetilde{r}_0$ tal que $\widehat{\rho}_0 = \widetilde{r}_0^T r_0 \neq 0$
$p_0 = v_0 = d_0 = 0$;
$\widehat{\rho}_0 = \widetilde{\alpha}_0 = \widetilde{\omega}_0 = 1$; $\tau = \|r_0\|$, $\theta_0 = 0$, $\eta_0 = 0$;
Mientras $\| r_{j-1} \| / \| r_0 \| \geq \varepsilon \quad (j = 1, 2, 3, ...)$,
Resolver $Mz = r_{j-1}$;
$\hat{\rho}_j = \widetilde{r}_0^T z$; $\tilde{\beta}_j = (\hat{\rho}_j / \hat{\rho}_{j-1})(\tilde{\alpha}_{j-1} / \tilde{\omega}_{j-1})$;
$p_j = z + \tilde{\beta}_j (p_{j-1} - \tilde{\omega}_{j-1} v_{j-1})$;
$y = Ap_j$;
Resolver $Mv_j = y$;
$\widetilde{\alpha}_j = \widehat{\rho}_j / \widetilde{r}_0^T v_j$;
$s = r_{j-1} - \widetilde{\alpha}_j y$;
Primera cuasi-minimización
$\widetilde{\theta}_j = \|s\| / \tau$; $c = \dfrac{1}{\sqrt{1 + \widetilde{\theta}_j^2}}$; $\widetilde{\tau} = \tau\, \widetilde{\theta}_j c$;
$\widetilde{\eta}_j = c^2 \widetilde{\alpha}_j$;
$\widetilde{d}_j = p_j + \dfrac{\theta_{j-1}^2 \eta_{j-1}}{\widetilde{\alpha}_j} d_{j-1}$;
$\widetilde{x}_j = x_{j-1} + \widetilde{\eta}_j \widetilde{d}_j$;
Hallar:
$u = z - \tilde{\alpha}_j v_j$;
$t = Au$;
$\omega_j = \dfrac{\langle s_j, t_j \rangle}{\langle t_j, t_j \rangle}$;
$r_j = s - \widetilde{\omega}_j t$;
Segunda cuasi-minimización
$\theta_j = \|r_j\| / \widetilde{\tau}$; $c = \dfrac{1}{\sqrt{1 + \theta_j^2}}$; $\tau = \widetilde{\tau}\, \theta_j c$;
$\eta_j = c^2 \widetilde{\omega}_j$;
$d_j = s_j + \dfrac{\widetilde{\theta}_j^2 \widetilde{\eta}_j}{\widetilde{\omega}_j} \widetilde{d}_j$;
$x_j = \widetilde{x}_j + \eta_j d_j$;
Fin

2.3.5.3. Algoritmo VGMRES

El algoritmo Variable GMRES es una variación del original GMRES [94] o si se prefiere del FGMRES [93]. El primer cambio introducido es la resolución directa del problema de mínimos cuadrados [52] que aparece en cada iteración. Por otro lado, se fija una subtolerancia δ que puede ser función de ε, la tolerancia exigida a la solución, y se incrementa la dimensión del subespacio de Krylov k una unidad en cada paso mientras que la norma del vector residuo sea mayor o igual a δ y k sea menor que la dimensión máxima permitida (por ejemplo, por exigencias de memoria) del subespacio de Krylov k_{top}. A partir de este punto, el valor de k es constante en las siguientes iteraciones (ver Galán y otros [51]).

ALGORITMO VGMRES PRECONDICIONADO
Aproximación inicial x_0. $r_0 = b - Ax_0$;
Elegir k_{init}, k_{top}, $\delta \in [0,1]$, $k = k_{init}$
Mientras $\| r_{i-1} \| / \| r_0 \| \geq \varepsilon$ $(i = 1, 2, 3, ...)$,
$\beta_{i-1} = \| r_{i-1} \|$; $v_i = r_{i-1}/\beta_{i-1}$;
Si $\| r_{i-1} \| / \| r_0 \| \geq \delta$ y $k < k_{top}$ Hacer $k = k + 1$;
Para $j = 1, ..., k$ Hacer
Resolver $Mz_j = v_j$;
$w = Az_j$;
Para $n = 1, ..., j$ Hacer
$\{H\}_{nj} = w^t v_n$;
$w = w - \{H\}_{nj} v_n$;
Fin
$\{H\}_{j+1j} = \| w \|$;
$v_{j+1} = w / \{H\}_{j+1j}$;
Fin
Resolver $U_k^t \bar{p} = d_k$ y $U_k p = \bar{p}$; con $\begin{matrix} \{d_k\}_m = \{H\}_{1m} \\ \{U_k\}_{lm} = \{H\}_{l+1m} \end{matrix}$ $l, m = 1, ..., k$;
$\lambda_i = \dfrac{\beta_{i-1}}{1 + d_k^t p}$;
$u_k = \lambda_i p$;
$x_i = x_{i-1} + Z_k u_k$; siendo $Z_k = [z_1, z_2, ..., z_k]$;
$r_i = Z_{k+1} \hat{r}_i$; con $\begin{matrix} \{\hat{r}_i\}_1 = \lambda_i \\ \{\hat{r}_i\}_{l+1} = -\lambda_i \{\bar{p}\}_l \end{matrix}$ $l = 1, ..., k$;
Fin

Este algoritmo, al igual que el FGMRES, permite cambiar el precondicionador en cada iteración.

2.4. Esquemas de almacenamiento

Las matrices de los sistemas lineales que estamos resolviendo son *sparse* y el número de elementos no nulos es mucho menor que el de los nulos. El esquema de almacenamiento de estas matrices, debe fundamentalmente reducir el coste computacional y el requerimiento de memoria en el ordenador.

Existen distintas formas de almacenamiento para estas matrices *sparse*, que datan de las primeras experiencias numéricas en el campo de la Ingeniería y que han ido mejorándose a medida que lo ha permitido el desarrollo de la tecnología informática. Una técnica más reciente, que tuvo su auge con la aparición de ordenadores paralelos es la de almacenamiento elemento a elemento (ver por ejemplo Montero y otros [86]). La efectividad de estos métodos está directamente relacionada con la utilización del paralelismo masivo en el computación.

2.4.1. Almacenamiento de la matriz del sistema

El esquema de almacenamiento que utilizamos en este trabajo, para una matriz *sparse* A de dimensión $n \times n$ es una versión del formato Ellpack-Itpack [94]. Se requieren dos matrices rectangulares de dimensión $n \times n_d$, una de reales y otra de enteros, donde n_d es el máximo número de términos no nulo por fila y $n_d << n$. La primera columna de la matriz de reales contiene a los términos de la diagonal de A, y a continuación, de forma ordenada se coloca el resto de términos no nulos de la fila. En la primera columna de la matriz de enteros se coloca el número de términos no nulos de la fila. A continuación, en cada fila se almacena la posición de columna de cada elemento no nulo de A. En ambas matrices las filas son completadas con tantos ceros como sea necesario.

Por ejemplo, para la matriz,

$$A = \begin{bmatrix} a_{11} & 0 & a_{13} & 0 & 0 & 0 & a_{17} \\ 0 & a_{22} & 0 & a_{24} & a_{25} & a_{26} & 0 \\ a_{31} & 0 & a_{33} & 0 & 0 & 0 & 0 \\ 0 & a_{42} & 0 & a_{44} & a_{45} & 0 & 0 \\ 0 & a_{52} & 0 & 0 & a_{55} & 0 & a_{57} \\ 0 & a_{62} & 0 & 0 & 0 & a_{66} & 0 \\ a_{71} & 0 & 0 & a_{74} & a_{75} & 0 & a_{77} \end{bmatrix}$$

las matrices de elementos no nulos y de posiciones serían,

$$V_{\mathbf{A}} = \begin{bmatrix} a_{11} & a_{13} & a_{17} & 0 \\ a_{22} & a_{24} & a_{25} & a_{26} \\ a_{33} & a_{31} & 0 & 0 \\ a_{44} & a_{42} & a_{45} & 0 \\ a_{55} & a_{52} & a_{57} & 0 \\ a_{66} & a_{62} & 0 & 0 \\ a_{77} & a_{71} & a_{74} & a_{75} \end{bmatrix} \qquad P_{\mathbf{A}} = \begin{bmatrix} 3 & 3 & 7 & 0 \\ 4 & 4 & 5 & 6 \\ 2 & 1 & 0 & 0 \\ 3 & 2 & 5 & 0 \\ 3 & 2 & 7 & 0 \\ 2 & 2 & 0 & 0 \\ 4 & 1 & 4 & 5 \end{bmatrix}$$

2.5. Reordenación

Las técnicas de reordenación, que se han aplicado fundamentalmente en la resolución de sistemas de ecuaciones lineales por métodos directos, están basadas en la teoría de grafos y proporcionan una nueva matriz con un ancho de banda o perfil menor, tal que el efecto *fill-in* debido al los procesos de factorización completa se reduce y, por tanto, también el espacio necesario para almacenar la matriz factorizada.

La reordenación no afecta al almacenamiento de la matriz, ya que el número de elementos no nulos, que son los almacenados en la forma compacta que usamos, se sigue conservando aunque ocupen posiciones distintas.

En el capítulo 6 de esta tesis, hemos considerado algunas técnicas de reordenamiento para mostrar el efecto de la renumeración en la construcción de precondicionadores del tipo inversa aproximada *sparse* y en la resolución iterativa de sistemas de ecuaciones lineales.

A continuación resumimos los algoritmos de reordenamiento (ver [1, 45]). Sea A una matriz no simétrica de dimensión n con patrón simétrico y sea $G = \langle V, E\rangle$ el grafo dirigido de la matriz A, donde $V = \{1, 2, ..., n\}$ es el conjunto de vértices y E es el conjunto de aristas $\langle i, j\rangle$ tal que $a_{i,j} \neq 0$. El conjunto de nodos adjacentes a v en G se denomina $Adj_G(v)$. El grado del nodo v es $|Adj_G(v)|$. $L(G)$ es una partición del conjunto de nodos V conocida como estructura de nivel enraizada en el nodo v, donde $L_1 = \{v\}$ y L_i es el conjunto de los nodos adyacentes al nivel L_{i-1} que aún no están en el nivel anterior. El ancho de la estructura de nivel i, $w_i(L)$, está definido por el cardinal de L_i, y el ancho de la estructura de nivel, $L(G)$, es $w(L) = \max w_i(L)$. El algoritmo del Grado Mínimo (MDG) se ha utilizado para reducir el efecto *fill-in* o de llenado en la factorización de matrices con patron de *sparsidad* simétrico. La implementación del algoritmo puede encontrarse en [54]:

ALGORITMO MDG

1 - Construir el grafo asociado a la matriz A, $g(x) = \langle V, E\rangle$, donde V es el conjunto de nodos y $E = \{\{a, b\} : a \neq b \ / \ a, b \in V\}$.

2 - Mientras $V \neq \emptyset$:

2.1- Elegir un nodo v de grado mínimo en $g(x) = \langle V, E\rangle$ y reordenar como nodo siguiente.

2.2 - Definir:

$V_v = V - \{v\}$,

$E_v = \{\{a, b\} \in E \ / \ a, b \in V_v\} \cup \{\{a, b\}\, a \neq b \ / \ a, b \in Adj_g(v)\}$.

Siendo, $Adj_g(v)$el conjunto de nodos conectados a v en el grafo $g(x)$.

y hacer

$V = V_v$, $E = E_v$ y $g(x) = \langle V, E\rangle$.

El algoritmo de Cuthill-McKee Inverso (RCM) [53] es una modificación del algoritmo de Cuthill-McKee [39] que toma simplemente el reordenamiento inverso del obtenido en último lugar. Estos algoritmos se caracterizan por reducir

el ancho de banda y el perfil de la matriz.

ALGORITMO RCM

1 - Construir el grafo asociado a la matriz A, $g(x) = \langle V, E\rangle$, siendo V el conjunto de nodos y $E = \{\{a,b\} : a \neq b \ / \ a,b \in V\}$.

2 - Determinar un nodo inicial (pseudo-periférico) y renumerarlo como x_1.

3 - Renumerar los nodos conectados a x_i en orden ascendente de grado.

4 - Efectuar el ordenamiento inverso.

El algoritmo de Mínimo Vecino (MN) [81] es una variante del algoritmo de Grado Mínimo que elimina los nodos seleccionados en la estructura del grafo asociado a la matriz A, tal que no se inserte ni se defina ninguna nueva arista en el grafo. Este selecciona el nodo que tiene el mínimo número de nodos adyacentes o vecinos. Este algoritmo es muy útil en la factorización incompleta con el mismo patrón de *sparsidad* de la matriz A, por ejemplo el precondicionador ILU(0) que será utilizado en los experimentos numéricos de esta tesis.

ALGORITMO MN

1 - Construir el grafo asociado a la matriz A, $g(x) = \langle V, E\rangle$, donde V es el conjunto de nodos y $E = \{\{a,b\} : a \neq b \ / \ a,b \in V\}$.

2 - Mientras $V \neq \emptyset$:

2.1- Elegir un nodo v de grado mínimo en $g(x) = \langle V, E\rangle$ y reordenar como nodo siguiente.

2.2 - Definir:

$V_v = V - \{v\}$, $E_v = \{\{a,b\} \in E \ / \ a,b \in V_v\}$.

y hacer

$V = V_v$, $E = E_v$ y $g(x) = \langle V, E\rangle$.

La elección del nodo inicial en el segundo paso de los algoritmos anteriores se ha determinado usando el algoritmo de George [54], lo que nos permite comenzar desde un nodo pseudo-periférico.

Si definimos la distancia $d(x,y)$ entre dos nodos x e y en un grafo $g(x) = \langle V, E\rangle$, como la longitud de la trayectoria más corta que une ambos nodos, y la excentricidad de un nodo x por $\varepsilon(x) = Max\,\{d(x,y)/x,y \in V\}$, el algoritmo se escribe de la siguiente forma,

ALGORITMO DE GEORGE. BÚSQUEDA DE NODOS PSEUDO-PERIFÉRICOS

1- Elegir un nodo arbitrario r de V.

2 . Generar una estructura con niveles enraizada en r,

$$\{L_0(r), L_1(r), \ldots, L_{\varepsilon(r)}(r)\}.$$

siendo $L_i(r) = \{x/d(x,r) = i\}$.

3 - Elegir un nodo x de grado mínimo en $L_{\varepsilon(r)}(r)$.

4 . Generar una estructura con niveles enraizada en x,

$$\{L_0(x), L_1(x), \ldots, L_{\varepsilon(x)}(x)\}$$

5 - Si $\varepsilon(x) > \varepsilon(r)$, establecer $x \rightarrow r$ y volver al paso 3.
6 - Caso contrario tomamos x como nodo inicial.

2.6. Precondicionadores explícitos e implícitos

Existen dos clases de métodos para la construcción de precondicionadores para una matriz A: (1) los métodos explícitos y (2) los métodos implícitos. Con los métodos explícitos podemos calcular una aproximación G de forma explícita para la inversa A^{-1} de una matriz no singular A dada. Esto es, para resolver $Ax = b$ consideramos el sistema,

$$GAx = \widetilde{b} \tag{2.79}$$

donde $\widetilde{b} = Gb$, resolviéndose iterativamente (2.79). Como podemos disponer explícitamente tanto de G como de A, cada iteración requiere sólo de productos matriz-vector. Alternativamente, podemos usar una aproximación explícita por la derecha y resolver iterativamente

$$AGy = b \tag{2.80}$$

para obtener el vector y. finalmente se calcula

$$x = Gy$$

En lo que sigue nos centraremos en el caso de precondicionadores por la izquierda pero los métodos presentados son igualmente aplicables para precondicionadores por la derecha.

En los métodos implícitos se calcula comúnmente una factorización aproximada de A como, por ejemplo, la factorización LU incompleta y se utiliza ésta como un camino implícito. Dado un vector x cada aplicación de este precondicionador requiere la solución del sistema lineal

$$LUd = r \tag{2.81}$$

donde $r = Ax - b$ para algún vector d que aparece en el método iterativo. Como sabemos, (2.81) requiere una solución por remonte y por descenso. Aquí la matriz de precondicionamiento es $C^{-1}A$ donde $C = LU$.

La matriz G en (2.79) o (2.80) puede verse como una inversa aproximada. Un método explícito da una inversa aproximada directamente mientras que el método implícito requiere primero de la factorización de la matriz que se usa para calcular las aproximaciones $d = (LU)^{-1}r$ de la acción de $A^{-1}r$ de la inversa en un método iterativo, por ejemplo. De aquí podemos decir que en un método implícito no construimos la inversa aproximada explícitamente. Sin embargo, el método implícito también puede ser usado para calcular inversas aproximadas explícitas.

Podemos también construir un método híbrido, por ejemplo, combinando los dos métodos en un camino con dos etapas donde se use un método de factorización incompleta (implícito) para la matriz particionada en bloques y, en la segunda etapa se use un método directo para aproximar la inversa del bloque pivote que aparece en cada etapa de la factorización.

En esta sección presentamos varios métodos para calcular aproximadas inversas de matrices. También discutimos cómo pueden aproximarse matrices que no se conocen explícitamente pero de las que se dispone de su producto matriz por vector.

Una estructura común de las matrices inversas aproximadas es la estructura en banda. Los rangos para las entradas de una matriz dada disminuyen según la distancia a la diagonal aumenta, la exactitud de dichas aproximaciones es de primordial importancia y discutiremos este tópico al final de la sección.

2.6.1. Dos métodos para el cálculo de aproximadas inversas de matrices en banda por bloques

Para ilustrar los métodos de aproximación explícita e implícita consideramos primeramente el cálculo de una aproximada inversa de una matriz en banda. Considérese una matriz $B = [b_{ij}]$, donde $b_{ij} = 0$ si $j < i - p$ o $j > i + q$. Por tanto, B es una matriz en banda con un ancho de semibanda izquierdo p y un ancho de semibanda derecho q. De hecho, podemos considerar el caso más general donde las entradas de B son matrices en bloque. Podemos asumir que los bloques de la diagonal B_{ij} son regulares y posteriomente hacer las suposiciones que se requieran. Se dice que B tiene un ancho de semibanda izquierdo del bloque igual a p y un ancho de semibanda derecho del bloque igual a q si $B_{ij} = 0$, si $j < i - p$ o $j > i + q$.

2.6.1.1. Un método explícito de aproximación

Consideraremos un método de cálculo de una inversa aproximada G de una matriz en banda B particionada por bloques. Sea $G = [G_{ij}]$ la partición correspondiente a la inversa aproximada de B a calcular, y asumimos que G tiene un ancho de semibanda del bloque p_i y q_i respectivamente, donde $p_i \geq p$ y $q_i \geq q$. Para calcular $G_{ij}\ i - p_1 \leq j \leq i + q_1$ hacemos,

$$(GB)_{ij} = \sum_{k=i-p_1}^{i+q_1} G_{i,k}B_{k,j} = \Delta_{ij}, \qquad i - p_1 \leq j \leq i + q_1 \tag{2.82}$$

donde,

$$\Delta_{ij} = \left\{ \begin{array}{l} I_i \quad \text{para } i = j \\ 0 \text{ para el resto} \end{array} \right\}$$

siendo I_i la matriz identidad del mismo orden que B_{ij} . La ecuación (2.82) da $p_1 + q_1 + 1$ ecuaciones con matrices en bloque que pueden usarse para calcular

el mismo número de bloques de G_{ij}. Sin embargo, como $B_{k,j} = 0$ si $k < j - q$ o $k > j + p$ tenemos,

$$\sum_{\substack{\text{máx } k=\{i-p_1, j-q\}}}^{\text{mín}\{i+q_1, j+p\}} G_{i,k}B_{k,j} = \Delta_{ij}, \qquad j = i - p_1, i - p_1 + 1, ..., i + q_1 \tag{2.83}$$

de forma similar, pero a menor escala las ecuaciones son válidas para $i < 2p_1+1$ y $i > m - q_1$, donde m es el orden de B tenemos,

$$\begin{aligned}
&\text{para } j = i - p_1 : && \sum_{k=i-p_1}^{i-p_1+p} G_{i,k}B_{k,i-p_1} = 0 \\
&\text{para } j = i - p_1 + 1 : && \sum_{k=i-p_1+1}^{i-p_1+p+1} G_{i,k}B_{k,i-p_1+1} = 0 \\
&\vdots && \vdots \\
&\text{para } j = i : && \sum_{k=i-p_1}^{i-p_1+p} G_{i,k}B_{k,i} = 0 \\
&\vdots && \vdots \\
&\text{para } j = i + q_1 : && \sum_{k=i+q_1}^{i+q_1} G_{i,k}B_{k,i+q_1} = 0
\end{aligned}$$

Estas ecuaciones tienen solución si hacemos algunas suposiciones adicionales sobre B. Ahora mostramos el método para el caso especial, pero importante, donde $p_1 = p = q_1 = q = 1$, esto es, el caso donde B y G son bloques triangulares donde es suficiente asumir para poder obtener solución, que B_{ij} y la matriz de Schur complementaria en (2.84) son no singulares. Entonces (2.83) muestra que

$$\begin{aligned}
G_{i,i-1}B_{i-1,i-1} + G_{i,j}B_{i,i-1} &= 0 \\
G_{i,i-1}B_{i-1,i} + G_{i,i}B_{i,i} + G_{i,i+1}B_{i+1,i} &= I, \\
G_{i,i}B_{i,i+1} + G_{i,i+1}B_{i+1,i+1} &= 0
\end{aligned}$$

así,

$$\begin{aligned}
G_{i,i-1} &= -G_{i,i}B_{i,i-1}(B_{i-1,i-1})^{-1} \\
G_{i,i+1} &= -G_{i,i}B_{i,i}(B_{i+1,i+1})^{-1}
\end{aligned}$$

y,

$$G_{i,i-1} = \left[B_{i,i} - B_{i,i-1}(B_{i-1,i-1})^{-1}B_{i-1,i} - B_{i,i+1}(B_{i+1,i+1})^{-1}B_{i+1,i}\right]^{-1} \tag{2.84}$$

donde debemos asumir que B_{ii} es no singular y que la inversa de la última matriz existe (complemento de Schur). Por ejemplo, si B es una H-matriz en bloques, mostraremos que las condiciones de solubilidad se cumplen. Para $i =$

1 y para $i = n$ ninguno de los términos entre paréntesis en (2.84) está ausente. Se puede ver que $G_{i,i-1}, G_{i,i}, G_{i,i+1}$ es la i-ésima fila de la inversa exacta de

$$\widehat{B} = \begin{pmatrix} I & & & & & & 0 \\ & \ddots & & & & & \\ & & B_{i-1,i-1} & B_{i-1,i} & & & \\ & & B_{i,i-1} & B_{i,i} & B_{i,i+1} & & \\ & & & B_{i+1,i} & B_{i+1,i+1} & & \\ & & & & & \ddots & \\ 0 & & & & & & I \end{pmatrix}$$

Es interesante resaltar que la matriz entre paréntesis en (2.84) es la misma que la matriz obtenida (en la i-ésima fila) cuando hacemos un reordenamiento par-impar de la matriz tridiagonal por bloques dada respectivamente. Nótese también que G es no simétrica en general, porque las matrices $G_{i,i-1}$ y $G_{i-1,i}^T$ no tienen que ser iguales necesariamente, aún cuando B sea simétrica. Las entradas de los bloques en cada fila de G pueden calcularse consecuentemente.

Finalmente puede verse que GB contiene una diagonal principal unitaria y una sub y super diagonal a la distancia dos de la diagonal principal. El resto de las entradas son nulas, De aquí que sea *sparse*.

Para una matriz tridiagonal tenemos

$$g_{i,i} = \left(b_{i,i} - \frac{b_{i,i-1}b_{i-1,i}}{b_{i-1,i-1}} - \frac{b_{i,i+1}b_{i+1,i}}{b_{i+1,i+1}} \right)^{-1},$$

$$g_{i,i-1} = -\frac{g_{i,i}b_{i,i-1}}{b_{i-1,i-1}}, \quad g_{i,i+1} = -\frac{g_{i,i}b_{i,i+1}}{b_{i+1,i+1}}, \quad i = 1, 2, ..., n$$

2.6.1.2. Método implícito de Aproximación

Considérese otra vez una matriz con bloques en banda B. Existe un algoritmo para calcular los bloques de la inversa exacta de B dentro de una banda consistente en $p_1 \geq p$ bloques a la izquierda y $q_1 \geq q$ bloques a la derecha de la diagonal, en cada fila del bloque. El algoritmo fue presentado en [9] y en [5] y se basa en una idea presentada en [101] (ver también [47]).

El algoritmo requiere la factorización previa de B. Por tanto, este método es prácticamente aplicable solo cuando los anchos de banda p, q de B sean pequeños y las entradas B_{ij} sean escalares o, en caso de ser matrices en bloque, tener órdenes pequeños. Si B se factoriza en la forma

$$B = (I - \widetilde{L})D^{-1}(I - \widetilde{U}) \tag{2.85}$$

donde D es una matriz en bloque diagonal y $\tilde{L}, \tilde{U}$ son bloques estrictamente triangulares inferiores y superiores respectivamente, entonces no es necesario calcular mas inversas de las matrices en bloques. Debe resaltarse que el cálculo de la matriz D en (2.85) requiere el cálculo de las inversas de las matrices en bloque pivotes. Una segunda ventaja importante del algoritmo es que no necesitamos calcular ninguna entrada de la inversa fuera de la parte de la banda requerida por B^{-1}. El algoritmo se basa en las siguientes identidades:

Lema 9 *Sea $B = LDU^{-1}$, donde $L = I - \tilde{L}$, $U = I - \tilde{U}$ y $\tilde{L}$ y $\tilde{U}$ son estrictamente triangular inferior y superior respectivamente. Entonces;*

$$(a)\ B^{-1} = DL^{-1} + \tilde{U}B^{-1}$$

$$(b)\ B^{-1} = U^{-1}D + B^{-1}\tilde{L}$$

Demostración. Tenemos $B^{-1} = U^{-1}DL$ así, $(I - \tilde{U})B^{-1} = DL^{-1}$ y $B^{-1}(I - \tilde{L}) = U^{-1}D$, los cual es *(a)* respectivamente *(b)* ■

Las relaciones *(a)* y *(b)* pueden usarse para el cálculo de los triángulos superiores e inferiores de B^{-1}respectivamente. Como L^{-1} es triangular inferior con matriz en bloque diagonal igual a la identidad, L^{-1} no se considerará en el cálculo del triángulo superior de B^{-1}, de igual forma que U^{-1} no se considerará en el cálculo del triángulo inferior.

El algoritmo para el cálculo de la banda de B^{-1} con bloques inferiores y superiores con ancho de semibanda p_1 y q_1 tienen la forma siguiente. Asumimos que $B = [B_{ij}]$ tiene un bloque de orden n, un bloque con ancho de semibanda inferior y superior p y q respectivamente y ha sido factorizada en la forma (2.85)

ALGORITMO BBI (INVERSA EN BLOQUES POR BANDA)

Desde $r = n, n-1, ..., 1$, hacer

$$(B^{-1})_{r,r} = D_{r,r} + \sum_{s=1}^{\min(q,n-r)} \tilde{U}_{r,r+s}(B^{-1})_{r,r+s} \tag{2.86}$$

Desde $k = 1, 2, ..., q_1$, hacer

$$(B^{-1})_{r-k,r} = \sum_{s=1}^{\min(q,n-r+k)} \tilde{U}_{r-k,r-k+s}(B^{-1})_{r-k+s,r} \tag{2.87}$$

Desde $k = 1, 2, ..., q_1$, hacer

$$(B^{-1})_{r,r-k} = \sum_{t=1}^{\min(p,n-r+k)} (B^{-1})_{r,r-k+t}\tilde{L}_{r-k+t,r-k} \tag{2.88}$$

Cabe destacar que sólo se realizan los productos matriz-matriz. Además si B es simétrica, solamente se necesita una de las ecuaciones (2.87) o (2.88).

Comentario 10 *La ecuación (2.86) muestra en particular que,*

$$(B^{-1})_{n,n} = D_{n,n}$$

donde $D_{n,n}$ es el último bloque en D. Esto se convertirá en una propiedad muy útil para los métodos de estimación de valores singulares y números de condición de matrices precondicionadas.

Comentario 11 *Patrón de sparsidad de la envoltura: El algoritmo BBI puede extenderse al caso donde B tiene un patrón de sparsidad en la envoltura, esto es, cuando existen p, q tales que $B_{ij} = 0$ para $i - j > p_i$ y para $j - i > q_i$. En particular, el método puede utilizarse para calcular una parte de la envoltura de la inversa de la matriz semibanda, para lo cual $B_{ij} = 0$, $|i - j| > p$, pero $B_{1,n} \neq 0$. Tales matrices aparecen, por ejemplo, para las ecuaciones en diferencias elípticas con condiciones de contorno periódicas. En este caso, el algoritmo toma la forma de la ecuación (2.86), pero debe añadirse el término $\tilde{U}_{r-k,n}(B^{-1})_{n,r}$ a la ecuación (2.87) y el término $(B^{-1})_{r,n}\tilde{L}_{n,r-k}$ a la ecuación (2.88) porque la última fila de $\tilde{L}$ y la última columna de $\tilde{U}$ están llenas generalmente. Más aún, para $r = n$ debe calcularse $(B^{-1})_{n-k,n}$ y $(B^{-1})_{n,n-k}$ para todo $k, k = 1, 2, ..., n-1$ en las ecuaciones (2.87) y (2.88)*

2.6.1.3. Matrices definidas positivas

El método anterior tiene una desventaja: aún cuando B sea definida positiva, la banda $[B^{-1}]^{p_1 q_1}$ de B^{-1} puede no ser definida postiva como se muestra en el siguiente ejemplo,

Ejemplo 12 *Sea* $G = \begin{bmatrix} 1 & -2 & 1 \\ -2 & 5 & -3 \\ 1 & -3 & 4 \end{bmatrix}$. *Entonces G es definida positiva, pero la banda $[G]^{1,1}$ de G, donde*

$$[G]^{1,1} = \begin{bmatrix} 1 & -2 & 0 \\ -2 & 5 & -3 \\ 0 & -3 & 4 \end{bmatrix}$$

es indefinida. Más generalmente, esto puede observarse de $G^{[p]} = G - R$. Aquí, $G^{[p]}$ es la banda simétrica, esto es, $p = q$, de G. R es indefinida puesto que tiene la diagonal nula. Por tanto, puede ocurrir que el menor autovalor de $G^{[p]}$ sea negativo y entonces $G^{[p]}$ será indefinida.

Sin embargo, para una clase importante de matrices B, denominadas matrices monótonas, podemos modificar la matriz $G^{[p]}$ con una compensación diagonal para las entradas que se eliminan en R, tal que la matriz modificada se convierta en definida positiva.

Definición 13 *Una matriz real A se dice que es monótona si $Ax \geq 0$ implica $x \geq 0$.*

Lema 14 *A es monótona si y sólo si A es no singular con* $A^{-1} \geq 0$.

Teorema 15 *(Compensación diagonal). Sea B monótona, simétrica y definida positiva. Sea* $G = B^{-1}$ *y* $R = G - G^{[p]}$, *donde* $G^{[p]}$ *es la banda simétrica de G con ancho de semibanda p. Entonces* $\tilde{G} = G^{[p]} + D$ *es definida positiva, donde D es una matriz diagonal tal que* $Du = Ru$, *para algún vector positivo u.*

Demostración. Como $G = B^{-1} \geq 0$, tenemos $R \geq 0$. Por tanto, $D \geq 0$. Sea

$$V = \text{diag}(u_1, u_2, ..., u_n)$$

Entonces la matriz $(D - R)V$ tiene diagonal dominante, y, por el teorema de los círculos de Gershgorin, los autovalores son no negativos. Lo mismo ocurre para $V^{-1}(D - R)V$ y también para $D - R$, porque $D - R$ es equivalente a esta última matriz. Por tanto, $D - R$ es semidefinida positiva y la relación

$$\tilde{G} - G = G^{[p]} + D - G = D - R$$

muestra que $\tilde{G} - G$ es semidefinida positiva. Entonces, $\tilde{G}$ es definida positiva debido a que G también lo es, al ser la inversa de una matriz definida positiva. ■

Dentro de todos los posibles vectores u, debe seleccionarse aquel que mejore el número de condición de $G^{[p]}B$ lo más posible. Esto es similar al uso de un vector para la modificación en los métodos de factorización incompleta.

Tanto los métodos explícitos como los implícitos tratados anteriormente fueron usados en Axelsson, Brinkkemper e Il'in [9] para cacular aproximaciones tridiagonales de las inversas de matrices tridiagonales que se obtienen durante la factorización en bloque de matrices en las ecuaciones en diferencias para ecuaciones diferenciales elípticas de segundo orden en dos dimensiones. Concus, Golub y Meurant [37] utilizaron también un método implícito para calcular la parte de la banda tridiagonal de la inversa de una matriz tridiagonal, basados en un método que usa dos vectores para generar la matriz, el cual fue sugerido primeramente por Asplund [4]. Sin embargo ocurre que este último método no puede extenderse de forma estable para calcular la banda con anchos de semibanda mayores o iguales que 2, por lo que en la práctica está limitado al caso $p = q = 1$.

2.6.2. Una Clase de Métodos para calcular las inversas aproximadas de matrices

Consideraremos ahora una rama general de la clase de métodos para construir inversas aproximadas, basados en los trabajos de [75] y [73]. Veremos que los métodos explícitos e implícitos presentados anteriormente se relacionan y además, para matrices simétricas son equivalentes a las dos versiones de esta clase de métodos.

La idea básica es la siguiente: Dado un patrón de *sparsidad*, se calcula la inversa aproximada G con dicho patrón de *sparsidad* para una matriz no singular A dada de orden n, esta será la mejor aproximación en alguna norma. Las normas basadas en las trazas de la matriz $(I - GA)W(I - GA)^T$, esto es, para el cuadrado de la norma de Frobenius con pesos de la matriz de error $(I - GA)$, proporcionan métodos prácticos y eficientes para ciertas selecciones de la matriz de pesos W. Considérese la funcional,

$$F_W(G) = \|I - GA\|_W^2 \equiv tr\left\{(I - GA)W(I - GA)^T\right\}$$

donde W es una matriz simétrica y definida positiva y se asume que G depende de algunos parámetros libres, $\alpha_1, ..., \alpha_p$. Estos parámetros pueden ser las entradas de G en posiciones definidas por un conjunto de p pares de índices $(i, j) \in S$, que es un subconjunto del conjunto total de pares, $\{(i, j); 1 \leq i \leq n; 1 \leq j \leq n\}$, siendo n el orden de A. Más aún, $g_{ij} = 0$ para todo par de índices fuera del patrón de *sparsidad* S.

Entonces, S define el patrón de *sparsidad* de $G = \{g_{ij}\}$ y $g_{ij} = 0$ para todo $(i, j) \in S^C$ que es el complementario de S definido por

$$S^C = \{(i, j) \notin S; 1 \leq i \leq n; 1 \leq j \leq n\}$$

Al igual que en el caso anterior, S contiene al menos los pares de índices $\{(i, i); 1 \leq i \leq n\}$. Así, el menor patrón de *sparsidad* de G es el de la matriz diagonal. Obsérvese que $F_W(G) \geq 0$ y $F_W(G) = 0$ si $G = A^{-1}$. Es más, el lema 16, nos muestra que $\|\cdot\|_W$ es una norma tal que $\|I - GA\|_W$ nos da una medida del error cuando G se aproxima a A^{-1}.

Queremos calcular los parámetros $\{\alpha_i\}$ para minimizar

$$F_W(G) = F_W(\alpha_1, ..., \alpha_p)$$

La solución a este problema debe satisfacer las relaciones estacionarias

$$\frac{\partial F_W}{\partial \alpha_i} = 0, i = 1, ..., p$$

Sea $\|B\|_W = \left\{tr(BWB^T)\right\}^{\frac{1}{2}}$. Para demostrar que $\|\cdot\|_W$ es una norma, definimos el siguiente lema del cual se también se derivan otras propiedades útiles.

Lema 16 *Sean A, B dos matrices cuadradas de orden n. Sea W una matriz definida*

positiva. Entonces,

(a) $tr(A) = tr(A^T), tr(A+B) = tr(A) + tr(B)$

(b) $tr(A) = \sum_{i=1}^{n} \lambda_i(A)$

(c) $tr(AA^T) = \sum_{i,j=1}^{n} a_{ij}$ *y* $tr(AA^T) \geq tr(A^2)$

(d) $\|AB\|_W \leq \|A\|_I \|B\|_W$, *donde* $\|\cdot\|_I$ *es la norma Frobenius,*
$\|A\|_I \leq \left\{ {}_{ij}a \sum_{i,j}^{2} \right\}^{\frac{1}{2}} = \left\{ tr(AA^T) \right\}^{\frac{1}{2}}$

(e) $\|A\|_I \leq \|A\|_W$ *y* $\|AB\|_W \leq \|A\|_W \|B\|_W$, *si y sólo si* $W - I$ *es definida positiva*

(f) $\|B\|_W = \left\{ tr(BWB^T) \right\}^{\frac{1}{2}}$ *es una norma aditiva*

Demostración. El apartado *(a)* se sigue de la definición del operador traza, $tr(A) = \sum_{i=1}^{n} a_{ij}$ y *(b)* se muestra en [8]. La primera propiedad de *(c)* se obtiene por el cálculo directo. Para demostrar la desigualdad, nótese que,

$$tr(A^2) = \sum_i \sum_j a_{ij} a_{ji} \leq \frac{1}{2} \sum_i \sum_j (a_{ij}^2 + a_{,i}^2) = \sum_i \sum_j a_{ij}^2 = tr(AA^T)$$

Para demostrar *(d)* partimos de $\|AB\|_F \leq \|A\|_F \|B\|_F$ (Apéndice A de [8]). De esta forma,

$$\begin{aligned} \|AB\|_W &= \left\{ tr(ABWB^TA^T) \right\}^{\frac{1}{2}} = \left\{ tr(A\tilde{B}\tilde{B}^TA^T) \right\}^{\frac{1}{2}} \\ &= \left\| A\tilde{B} \right\|_I \leq \|A\|_I \left\| \tilde{B} \right\|_I = \|A\|_I \|B\|_W \end{aligned}$$

donde $A\tilde{B} = BW^{\frac{1}{2}}$. Ahora *(e)* se sigue de *(d)*, y

$$\begin{aligned} \|A\|_I &= \left\{ tr(AA^T) \right\}^{\frac{1}{2}} = \left\{ \sum \lambda_i(AA^T) \right\}^{\frac{1}{2}} \leq \left\{ \sum \lambda_i(AWA^T) \right\}^{\frac{1}{2}} \\ &= \left\{ tr(AWA^T) \right\}^{\frac{1}{2}} = \|A\|_W \end{aligned} \tag{2.89}$$

donde la desigualdad se cumple si y sólo si $W - I$ es semidefinida positiva. Finalmente,

$$\begin{aligned} \|A+B\|_W &= \left\{ tr\left[(AW^{\frac{1}{2}} + BW^{\frac{1}{2}})(AW^{\frac{1}{2}} + BW^{\frac{1}{2}})^T \right] \right\}^{\frac{1}{2}} \\ &= \left\| AW^{\frac{1}{2}} + BW^{\frac{1}{2}} \right\|_I \\ &\leq \left\| AW^{\frac{1}{2}} \right\|_I + \left\| BW^{\frac{1}{2}} \right\|_I \leq \|A\|_W + \|B\|_W \end{aligned}$$

■

Considérese ahora el problema: hallar $G \in S$ tal que

$$\|I - GA\|_W \leq \left\|I - \tilde{G}A\right\|_W \text{ para todo } \tilde{G} \in S$$

donde $\left\|I - \tilde{G}A\right\|_W$ es la menor para $\tilde{G} = G$ dentro de todas las matrices con patrón de *sparsidad* S. Para simplificar la notación usamos la misma que para el conjunto de matrices que tienen este patrón de *sparsidad* como el correspondiente conjunto de índices. Nótese que,

$$\begin{aligned} F_W(G) &= tr\left\{(I - GA)W(I - GA)^T\right\} \\ &= trW - tr(GAW) - tr(GAW)^T + tr(GAWA^TG^T) \\ &= trW - \sum_{i,j} g_{ij}\left[(AW)_{ji} + (AW^T)_{ji}\right] + tr(GAWA^TG^T) \end{aligned}$$

Por lo tanto, las relaciones estacionarias,

$$\frac{\partial F_W(G)}{\partial g_{ij}} = 0, \quad (i,j) \in S$$

muestran que,

$$-(AW)_{ji} - (AW^T)_{ji} + (AWA^TG^T)_{ji} + (GAWA^T)_{ij} = 0$$

o, como $W = W^T$,

$$(GAWA^T)_{ij} = (WA^T)_{ij}, \quad (i,j) \in S \tag{2.90}$$

La ecuación (2.90) define el conjunto de ecuaciones que deben satisfacer las entradas de $G \in S$. En dependencia de la selección de S y/o A, estas ecuaciones pueden o no tener solución. Por otro lado, la ecuación (2.90) es un caso particular de

$$(GAV)_{ij} = V_{ij}, \quad (i,j) \in S \tag{2.91}$$

donde $V = WA^T$ en la ecuación (2.90). Puede verse que (2.91) tiene solución única si todos los menores de AV, restringidos al conjunto S, son no singulares. Entonces, el problema es encontrar las matrices apropiadas W (o V) y los conjuntos S.

Comentario 17 *(Cálculos en paralelo) Si asumimos que V es conocida de forma explícita, las ecuaciones para el cálculo de las entradas de G pueden paralelizarse, porque las entradas en cualquier fila de G se calculan independientemente de las entradas de otras filas, es decir, pueden ser calculadas en paralelo entre las filas. Una matriz G definida de esta forma será en general no simétrica aún si A es simétrica.*

Establecemos los resultados anteriores en un teorema y consideramos algunas selecciones particulares e importantes en la práctica de la matriz de pesos W.

Teorema 18 *Sea A no singular y sea $G \in S$ tal que,*

$$\|I - GA\|_W \leq \left\|I - \tilde{G}A\right\|_W \text{ para cualquier } \tilde{G} \in S$$

Entonces G debe ser solución del sistema lineal

$$(GAWA^T)_{ij} = (WA^T)_{ij}, \quad (i,j) \in S$$

Para la siguiente selección de W, este sistema y $F_W(G)$ son como sigue,
(a) Para $W = A^{-1}$, donde A es simétrica y definida positiva, entonces,

$$F_W(G) = tr(A^{-1}) - tr(G)$$

y G satisface,

$$(GA)_{ij} = \delta_{ij}, \qquad (i,j) \in S$$

(b) Para $W = I_n$, entonces,

$$F_W(g) = n - tr(GA)$$

y G satisface,

$$(GAA^T)_{ij} = (A^T)_{ij}, \qquad (i,j) \in S$$

(c) Para $W = A^k$, donde k es un entero positivo y A es simétrica y definida positiva, entonces,

$$F_W(G) = tr(A^k - GA^{k+1})$$

y G satisface,

$$(GA^{k+2})_{ij} = (A^{k+1})_{ij}, \qquad (i,j) \in S$$

(d) Para $W = (A^TA)^{-1}$ entonces,

$$F_W(G) = tr\left[(A^{-1} - G)A^{-T}\right]$$

y G satisface,

$$G_{ij} = (A^{-1})_{ij}, \qquad (i,j) \in S$$

Demostración. La parte general establecida es (2.90) la cual se ha demostrado anteriormente. Para la matriz G óptima, el lema 16 *(a)* muestra que,

$$\begin{aligned} F_W(G) &= trW - tr(2GAW - GAWA^TG^T) \\ &= trW - 2tr(GAW) + tr(WA^TG^T) \\ &= trW - tr(GAW) = tr(W - GAW) \\ &= tr((I - GA)W) \end{aligned}$$

Con lo cual se obtienen las diferentes selecciones de W. ■

Note que para WA^{-1} tenemos el método explícito (2.82) discutido anteriormente para A simétrica y para $W(A^TA)^{-1}$ usamos el método implícito.

Para $W = I_n$, entonces $\|I - GA\|_W = \|I - GA\|_I$, esto es, este es el error en la norma de Frobenius sin peso.

Debido a la relativa simplicidad del cálculo de G, el método con $W = A^{-1}$ puede ser el más importante desde el punto de vista práctico cuando A es simétrica y definida positiva. Sin embargo, G no será simétrica generalmente.

La funcional F_W nos da cierta información de cómo seleccionar el conjunto de *sparsidad* de forma óptima. Obsérvese que $F_W(G) \geq 0$ y, por ejemplo, para $W = A^{-1}$ (si A es simétrica y definida positiva) y para $W = I_n$, donde $F_W(G) = tr(A^{-1}) - tr(g)$ y $F_W(G) = n - tr(GA)$, respectivamente, vemos que debemos tratar de seleccionar el conjunto S (de algún cardinal dado), para maximizar las trazas de G y de GA, respectivamente. En la práctica puede calcularse $F_W(G)$ sólo para $W = A^k$, con k un entero no negativo.

Comentario 19 *Como se vió en el Comentario 17, si la matriz V en la ecuación (2.91) está disponible, entonces las ecuaciones para calcular G se desacoplan, de tal forma que las entradas en cualquier fila de G pueden calcularse independientemente de las entradas de la otra fila. En el teorema 18 esto se cumple para los casos (a), (b) y (c) (para (c) si A^{k+1} está disponible), pero no en el caso (d). Se ve claramente por el método que, en (a),*

$$(GA)_{ij} = \delta_{ij}, \qquad (i,j) \in S$$

no todas las entradas de A se utilizarán para calcular $G \in S$. Como

$$(GA)_{ij} = \sum_{k,(i,k)\in S} g_{i,k}a_{k,j} = \delta_{ij}, \qquad (i,j) \in S,$$

vemos que para cada i se utilizarán sólo las entradas $a_{k,j}$, donde $(i,k) \in S \wedge (i,j) \in S$. En el grafo de la matriz para G, esto significa que para cada i sólo las entradas $a_{k,j}$ de A serán utilizadas, donde los vértices k y j están directamente conectados al vértice i. Por ejemplo,

$$S = \left\{ \begin{array}{c} (1,1),(1,2),(i,i-1),(i,i),(i,i+1) \\ (\text{para } i = 2,3,...,n-1),(n,n-1),(n,n) \end{array} \right\}$$

si entonces sólo se requieren las entradas $a_{k,j}$ donde,

$$k = i-1, i, i+1, \quad j = i-1, i, i+1$$

Esto es, para calcular las entradas en la i-ésima fila de G, $g_{i,i-1}$, $g_{i,i}$, $g_{i,i+1}$, sólo se usarán las entradas de A,

$$a_{i-1,i-1}, a_{i-1,i}, a_{i-1,i+1}, a_{i,i-1}, a_{i,i}, a_{i,i+1}, a_{i+1,i-1}, a_{i+1,i}, a_{i+1,i+1}$$

Esto significa que, en este caso, solo la parte pentadiagonal de A se usa y las otras entradas de A no tendrán influencia en la inversa aproximada de G. De forma más general, calcular g_{ik}, $(i,k) \in S$ para $k = k_1^{(i)}, k_2^{(i)}, ..., k_{S_{(j)}}^{(i)}$ (si existen pares de índices

s_i en S para la i-ésima fila) para cada i, entonces las únicas entradas usadas serán $a_{k,j}$ con $k = k_1^{(i)}, k_2^{(i)}, ..., k_{S_{(j)}}^{(i)}$, $j = k = k_1^{(i)}, k_2^{(i)}, ..., k_{S_{(j)}}^{(i)}$. Naturalmente, podemos esperar en general que este método sea exacto sólo si todas las otras entradas de A (fuera del correspondiente patrón de sparsidad) sean cero o relativamente pequeña. El uso de la compensación diagonal de las entradas elminadas algunas veces puede mejorar la aproximación significativamente. Para el caso (b) tenemos,

$$\|GA - I\|_W = \|GA - I\|_F^2 = \sum_{i=1}^{n} \left\|g_i A - e_i^T\right\|_2^2$$

Esto muestra que el cálculo de G puede realizarse independientemente como una colección de n subproblemas de mínimos cuadrados para cada fila específica g_i . Por el desacoplamiento descrito en el Comentario 17, se sigue que, sólo en el caso donde no ocurra desacoplamiento, la aproximación G de A preservará la simetría en general. Por tanto, aún cuando A sea simétrica, las matrices G calculadas por uno de los métodos en (a), (b), o (c) serán generalmente no simétricas. Cuando se utiliza la matriz G como precondicionador en un método iterativo, sabemos que el número de condición espectral puede ser importante para la convergencia de dicho método iterativo. A continuación se muestra una estimación del número de condición, que es una extensión al caso de las normas Frobenius con pesos, de un resultado aparecido en [38]. Además, nos da una condición suficiente para la positividad de GA.

Teorema 20 *Sea A una matriz no singular de orden n y sea G la inversa aproximada obtenida minimizando $\|I_n - GA\|_W$ sujeta a un patrón de sparsidad dado S, donde $W - I_n$ es semidefinida positiva. Sea G la solución óptima y sea $\varepsilon = \|I_n - GA\|_W$ donde asumimos que $\varepsilon < 1$. Entonces (a) GA es no singular y*

$$\kappa(GA) = \left\|(GA)^{-1}\right\|_2 \|GA\|_2 \leq \frac{1+\varepsilon}{1-\varepsilon}$$

(b) GA es definida positiva.(Observe que en los casos (a), (c) y (d) en el Teorema 18 podemos escalar A usando un factor para convertir a $W - I_n$ en semidefinida positiva)

Demostración. Como se sabe, $\|B\|_2 \leq \|B\|_I$ para cualquier matriz B. Por tanto, usando el Lema 16 (e), encontramos

$$\|I_n - GA\|_2 \leq \|I_n - GA\|_I \leq \|I_n - GA\|_W = \varepsilon$$

por tanto,

$$|\|GA\|_2 - \|I_n\|_2| \leq \varepsilon$$

y

$$1 - \varepsilon \leq \|GA\|_2 \leq 1 + \varepsilon$$

Además, como $GA = I_n - (I_n - GA)$, para cualquier $x \in \mathbb{R}^n$, tenemos que,

$$\begin{aligned} x^T GAx &= x^T x - x^T (I_n - GA)x \\ &\geq \|x\|_2^2 - \|x\|_2^2 \|(I_n - GA)\|_2 \\ &\geq (1 - \varepsilon) \|x\|_2^2 \end{aligned}$$

de lo cual concluimos que GA es definida positiva cuando $\varepsilon < 1$. También, tenemos

$$\left\|(GA)^{-1}\right\|_2 = \left\|(I_n - (I_n - GA))^{-1}\right\|_2 \leq \frac{1}{1-\varepsilon}$$

así,

$$\kappa(GA) = \left\|(GA)^{-1}\right\|_2 \|GA\|_2 \leq \frac{1+\varepsilon}{1-\varepsilon}$$

■

Observe que,

$$\|I_n\|_W^2 = \begin{pmatrix} tr(A^{-1}) & \text{en el caso (a)} \\ n & \text{en el caso (b)} \\ tr(A^k), k \geq 1 & \text{en el caso (c)} \\ tr((AA^T)^{-1} & \text{en el caso (d)} \end{pmatrix}$$

del teorema 18 y $\|I_n - GA\|_W$ está disponible en la práctica sólo en los casos (b) y (c). Si se calcula una inversa aproximada G (por cualquier método), entonces la demostración del teorema 20 muestra que si $\varepsilon' = \|I_n - GA\|_F < 1$, entonces $\kappa(GA) \leq \frac{1+\varepsilon'}{1-\varepsilon'}$.

El patrón de *sparsidad* S de G no tiene que ser el mismo que el de la matriz de coeficientes A, pues puede ser aún más *sparse*. Esto es importante para ciertas ecuaciones integrales donde frecuentemente A es una matriz llena pero sus inversas pueden aproximarse con exactitud usando una matriz más *sparse*.

2.6.3. Comparación de resultados

Ahora mostramos algunos resultados. El primero es obvio, según lo que se plantea en el teorema 18 (b): Si $W = I_n$ y $S_2 \supseteq S_1$ entonces,

$$n \geq tr(G_2 A) \geq tr(G_1 A)$$

así, G_2, que corresponde a S_2, es una mejor aproximación de A^{-1} en el sentido de la norma Frobenius que G_1, correspondiente a S_1. Note que si $S = S_0 = \{(i,i); 1 \leq i \leq n\}$ entonces el teorema 18 (b) muestra que,

$$(GAA^T)_{i,i} = (A^T)_{i,j}$$

donde $G = diag(g_{i,i})$, así,

$$g_{i,i} = a_{i,i} / \sum_{j=1}^{n} a_{ij}^2$$

Por tanto, $g_{i,i} \leq a_{i,i}^{-1}$ y $G \neq D_A^{-1}$ (menos para $A = D_A$), donde D_A es la diagonal de A. Considérese a continuación el caso $W = A^{-1}$, donde A es semidefinida positiva. Entonces $G = D_A$ si $S = S_0$. El siguiente teorema de comparación es válido.

Teorema 21 *Sea $W = A^{-1}$, donde $A[a_{ij}]$ es semidefinida positiva. Entonces, si $S_2 \supseteq S_1$, tenemos*

$$a_{i,i}^{-1} \leq (G_1)_{i,i} \leq (G_2)_{i,i} \leq (A^{-1})_{i,i} \tag{2.92}$$

Demostración. Primero note que el teorema 18 (a) muestra que $G = D_A^{-1}$ si $S = S_0 = \{(i,i); 1 \leq i \leq n\}$. Además, para cualquier S,

$$F_W(G) = tr(A^{-1} - G) = \sum_{i=1}^{n}((A^{-1})_{i,j} - g_{i,j})$$

Como las entradas de cada fila de G se calculan independientemente de las entradas de las otras filas, tenemos, de hecho,

$$F_W(G) = \sum_{i=1}^{n} F_W(G)_i$$

donde $F_W(G)_i = (A^{-1})_{i,j} - g_{i,j}$ y la solución de G es tal que minimiza cada término de $F_W(G)_i$ independientemente de los otros términos, bajo las restricciones $G \in S$. Esto muestra que cada término es no negativo, esto es, $F_W(G)_i \geq 0$ y, que $F_W(G_2)_i \geq F_W(G_1)_i$, lo cual demuestra la ecuación (2.92). Los valores extremos de la izquierda y la derecha se toman para $S = S_0$ y S en el total del conjunto respectivamente. ■

Ahora la ecuación (2.92) no se cumple para matrices no simétricas definidas positivas A para las cuales G satisface,

$$(GA)_{i,j} = \delta_{i,j}, \quad (i,j) \in S$$

Lema 22 *Para $i = 1, 2, ..., n$ sea $B^{(i)} = \alpha^{(i)} \odot A$, el producto de Hadamard (entradas pareadas), donde*

$$\alpha_{kj}^{(i)} = \begin{cases} 1, \textit{ para } (k,j) \textit{ tal que } (i,k) \in S \wedge (i,j) \in S \\ \textit{(todas las entradas en este bloque son nulas)} \\ 1, \textit{ para } k = j \\ 0, \textit{ en otro caso} \end{cases}$$

Asumimos que $B^{(i)}$ es no singular y consideramos el método del teorema 18 (a). Entonces las entradas no nulas de la i-ésima fila de G son iguales a las correspondientes entradas de la inversa exacta de $B^{(i)}$.

Demostración. Sea $G^{(i)}$ la inversa exacta de $B^{(i)}$. La ecuación para la i-ésima fila de $G^{(i)}$ es,

$$\sum_{k=1}^{n} G_{i,k}^{(i)} B_{k,j}^{(i)} = \delta_{i,j}, \quad j = 1, 2, ..., n$$

Asumimos que todas las entradas $B^{(i)}_{k,j} = 0$ para $(i,k) \in S^C$ o $(i,j) \in S^C$. Así, para un i fijo y $(i,j) \in S$ tenemos

$$(G^{(i)}B^{(i)})_{i,j} = \sum_{k,(i,k)\in S} g^{(i)}_{ik} a_{kj} = \delta_{ij}$$

y la demostración se sigue del comentario 19 ■

Como una consecuencia inmediata de la definición de una H-matriz en bloques [8], tenemos que si A es una H-matriz en bloques, entonces $\alpha^{(i)} \odot A$ es también una H-matriz en bloques. Por tanto, en particular, $B^{(i)}$ es no singular si A es una H-matriz en bloques, y de esta forma $G^{(i)}$ queda determinada de forma única. Este hecho importante se demostrará de nuevo en el teorema 26.

El siguiente teorema muestra que, para M-matrices podemos afinar la comparación de resultados hecha en el teorema 21.

Teorema 23 *Sea A una matriz tipo M y G determinada por $(GA)_{i-,j} = \delta_{i,j}$, $(i,j) \in S$ (a) Entonces G es no singular y $A = G^{-1} - R$ es una partición débilmente regular. (b) Sea $S_2 \supseteq S_1$ y sea G_i correspondiente a S_i, $i = 1,2$. Entonces*

$$D_A^{-1} \leq G_1 \leq G_2 \leq A^{-1}$$

esto es, la monotonía se cumple para todas las entradas, no sólo las de la diagonal, como en el teorema 21

Demostración. El lema 22 muestra que las entradas no nulas de la i-ésima fila de G son iguales a las correspondientes entradas de la i-ésima fila de $G^{(i)}$, donde $G^{(i)}$ es la inversa exacta de $B^{(i)} = \alpha^{(i)} \odot A$ y $\alpha^{(i)}$ está definida en el lema 22. Como A es una M-matriz, se ve claramente que $B^{(i)}$ es una M-matriz también y en particular es no singular. Por tanto, sus inversas $G^{(i)}$ son no negativas, y el lema 22 muestra que $G \geq 0$; en adición, G no tiene filas nulas. (Por ejemplo, el teorema 21 muestra que $g_{i,i} \geq a_{i,i}^{-1}$).

Considere ahora

$$GA = I - GR$$

Primero mostramos que GA es una Z-matriz: para $(i,j) \in S$ tenemos, $(GA)_{i,j} = \delta_{i,j}$. Así, en particular, $(GA)_{i,j} = 0$ si $i \neq j$ y para $(i,j) \notin S$ encontramos que

$$(GA)_{i,j} = \sum_{k,(i,k)\in S} g_{i,k}a_{k,j} = \sum_{k,k\neq j,k,(i,k)\in S} g_{i,k}a_{k,j}$$

porque $g_{i,j} = 0$ cuando $(i,j) \notin S$. Por tanto $(i,j) \notin S$ los términos en el sumatorio anterior son todos no positivos cuando $a_{k,j} \leq 0, k \neq j$ y $g_{i,k} \geq 0$ (como hemos mostrado arriba).

A continuación mostramos que GA es una M-matriz. Como A es una M-matriz, existe $x > 0$ tal que $AX > 0$. Por tanto, como G es no negativa y no tiene filas nulas, $GAx > 0$. Esto, unido al resultado anterior, nos muestra que GA es

una M-matriz. Como A es una M-matriz, ésta es, en particular, no singular, por lo que G también es no singular. Esto muestra que la partición $A = G^{-1} - R$ es una partición débilmente regular lo que demuestra el apartado (a).

Para demostrar el apartado (b), está que para $S_2 \supseteq S_1$, las entradas no nulas de $(G_2)_i$ (la i-ésima fila) son iguales a las correspondientes entradas no nulas de la inversa exacta de $\alpha_2^{(i)} \odot A$, y las entradas no nulas de $(G_1)_i$ son iguales a las correspondientes entradas no nulas de la inversa exacta de $\alpha_1^{(i)} \odot A$, donde $\alpha_2^{(i)} \geq \alpha_1^{(i)}$ (en el sentido de las entradas). Esto demuestra que

$$I - G_2 A \leq I - G_1 A \quad \text{o} \quad G_1 A \leq G_2 A \text{ (en el sentido de las entradas)}$$

Por tanto, cuando $A^{-1} \geq 0, G_1 \leq G_2$. La parte inferior de (b) se sigue del teorema 20 y la parte superior se sigue siendo S_2 el conjunto completo para el cual $G_2 = A^{-1}$. ■

Corolario 24 *Sea A una M-matriz y G, una matriz determinada por $(GA)_{i,j} = \delta_{i,j} \in S$. Entonces $\rho(I - GA) < 1$.*

Demostración. Como A es una M-matriz, podemos escribir $A = D_A - B$ donde D_A es la diagonal de A y $B \geq 0$. Esto es una partición regular. En particular,

$$\rho(I - D_A^{-1}A) = \rho(D_A^{-1}B) < 1$$

Pero la demostración del teorema 23 muestra que $GA \geq \rho(I - D_A^{-1}A)$, porque $I - GA$ es no negativa ($(GA)_{i,j} = 1$). ■

Ahora extendemos este resultado a la clase de H-matrices. Primero mostramos una comparación de resultados clásica.

Lema 25 *(Teorema de Ostrovsky). Sea A una H-matriz y sea $\mathcal{M}(A)$ su matriz de comparación. Entonces $|A^{-1}| \leq \mathcal{M}(A^{-1})$, donde $|A|$ es la matriz cuyas entradas son $|a_{i,j}|$.*

Demostración. Cosidérese la partición $A = D - B$ donde $D = diag(A)$. Entonces $\mathcal{M}(A) = |D| - |B|$, y, asumiendo que $\mathcal{M}(A)$ es una M-matriz, $\rho(|D^{-1}B| < 1$. Más aún, $A = D(I - D^{-1}B)$, lo que muestra que,

$$\begin{aligned} A^{-1} &= (I - D^{-1}B)^{-1}D^{-1} = [I + D^{-1}B + (D^{-1}B)^2 + ...]D^{-1} \\ &= D^{-1} + D^{-1}BD^{-1} + D^{-1}BD^{-1} + ... \end{aligned}$$

Por lo tanto

$$\begin{aligned} |A^{-1}| &\leq |D|^{-1} + |D|^{-1}|B||D|^{-1} + ... \\ &= (I - |D|^{-1}|B|)^{-1}|D|^{-1} = (|D| - |B|)^{-1} = \mathcal{M}(A)^{-1} \end{aligned}$$

■

Teorema 26 *Sea A una H-matriz y G, definida por $(GA)_{i,j} = \delta_{i,j}, (i,j) \in S$. Entonces G está únicamente determinada y $\rho(I - GA) < 1$*

Demostración. De forma similar a la demostración de la parte (b) del teorema 23, notamos primeramente que la i-ésima fila de G es la inversa exacta de $\alpha^{(i)} \odot A$ para un producto matricial de Hadamard. Claramente, $\alpha^{(i)} \odot A$ es una H-matriz. Por tanto,, su inversa existe. Como esto es cierto para cualquier i, G está determinada de forma única.

Sea $\mathcal{M}(A)$ la matriz de comparación y sea $\tilde{G} = [\tilde{g}_{i,j}]$ la inversa aproximada de $\mathcal{M}(A)$ para el conjunto S, esto es,

$$\{\tilde{G}\mathcal{M}(A)\}_{i,j} = \delta_{i,j}, \quad (i,j) \in S$$

Primero, el teorema de Ostrovsky muestra que, para inversas exactas G_i y $\tilde{G}_i$ de $\alpha^{(i)} \odot A$ y $\alpha^{(i)} \odot \mathcal{M}(A)$, respectivamente, tenemos $|G_i| \leq \tilde{G}_i$. Esto es,

$$|G| \leq \tilde{G}$$

Demostraremos a continuación que

$$|GA| \leq \left|\tilde{G}\mathcal{M}(A)\right| \tag{2.93}$$

Esta inecuación es trivial para $(i,j) \in S$, por lo que necesitamos considerar solamente $(i,j) \notin S$.

Para tales posiciones tenemos

$$\begin{aligned} |(GA)_{i,j}| &= \left| \sum_{k;k\neq j(i,k)\in S} g_{i,k}a_{k,j} \right| \leq \sum_{k;k\neq j(i,k)\in S} |g_{i,k}|\,|\{\mathcal{M}(A)\}_{k,j}| \\ &\leq \sum_{k;k\neq j(i,k)\in S} \tilde{g}_{i,k}\,[-\{\mathcal{M}(A)\}_{k,j}] = -\{\tilde{G}\mathcal{M}(A)\}_{i,j} \end{aligned}$$

Por tanto,

$$|I - GA| \leq \left|I - \tilde{G}\mathcal{M}(A)\right|$$

y, usando el teorema de Perron-Frobenius,

$$\begin{aligned} \rho(I - GA) \leq \rho(|I - GA|) &\leq \rho(\left|I - \tilde{G}\mathcal{M}(A)\right|) \\ &= \rho(I - \tilde{G}\mathcal{M}(A)) < 1 \end{aligned}$$

donde la última desigualdad se sigue del corolario 24 ■

Comentario 27 *(H-Matrices en bloques) Tomando normas en vez de valores absolutos encontramos que el teorema 26 también se cumple para H-matrices en bloques y que*

$$\|G_{i,j}\| \leq \tilde{g}_{i,j} \quad y \quad \|(I - GA)_{i,j}\| \leq \left|(I - \tilde{G}\mathcal{M}_b(A))_{i,j}\right| \tag{2.94}$$

donde $\tilde{g}_{i,j}$ son las entradas de la matriz $\tilde{G}$ que satisfacen $\{\tilde{G}\mathcal{M}_b(A)\}_{i,j} = \delta_{i,j}, (i,j) \in S$, y $\mathcal{M}_b(A)$ es la matriz de comparación en bloques

Corolario 28 *Sea A una H-matriz y $\mathcal{M}(A)$ su matriz de comparación, y sea $x > 0$ tal que $\mathcal{M}(A)x > 0$. Entonces $\mathcal{M}(GA)x > 0$, donde G se define como en el teroema 26, esto es, la dominancia diagonal generalizada se mantiene.*

Demostración. Por la ecuación (2.93) tenemos $|GA| \leq \left|\tilde{G}\mathcal{M}(A)\right|$. Como $\mathcal{M}(A)x > 0$ para algún x positivo y $\tilde{G} \geq 0$, tenemos $\tilde{G}\mathcal{M}(A)x > 0$. Las entradas de la diagonal de GA son unos y $(GA)_{i,j} \leq 0, i \neq j$ (ver la demostración del teorema 23). Entonces,

$$\begin{aligned}\mathcal{M}(GA)x &= x - (I - GA)x \geq x - |I - GA|\, x \\ &\geq x - \left|I - \tilde{G}\mathcal{M}(A)\right| x = x - \left[I - \tilde{G}\mathcal{M}(A)\right] x \\ &= \tilde{G}\mathcal{M}(A)x > 0\end{aligned}$$

donde hemos usado la propiedad de que $\tilde{G}\mathcal{M}(A)$ es una Z-matriz. ■

Los métodos explícitos de precondicionamento fueron propuestos primeramente por Benson [12], Frederickson [50], Benson y Fredericson [13] y Ong [89], siendo las mejores aproximaciones en el sentido de la norma de Frobenius. Fueron extendidos poteriormente con un análisis cuidadoso a las mejores aproximaciones en normas de Frobenius generalizadas por [75, 73].

Capítulo 3

Inversa aproximada basada en el producto escalar de Frobenius

En este Capítulo se presentan unos precondicionadores de construcción en paralelo para la resolución de sistemas de ecuaciones lineales. El cálculo de estos precondicionadores se realiza mediante proyecciones ortogonales usando el producto escalar de Frobenius. Así, el problema $\min_{M \in \mathcal{S}} ||AM - I||_F$ y la matriz $M_0 \in \mathcal{S}$ correspondiente a este mínimo, (siendo $\mathcal{S}$ cualquier subespacio de $\mathcal{M}_n(\mathbb{R})$), se calcula explícitamente usando una fórmula acumulativa para reducir los costes computacionales cuando el subespacio $\mathcal{S}$ se extiende a otro que lo contenga. En cada paso los cálculos se realizan aprovechando los resultados anteriores, lo que reduce considerablemente la cantidad de trabajo. En el Capítulo se muestran estos resultados generales para el subespacio de las matrices M tal que AM es simétrica. La principal aplicación se desarrolla para el subespacio de las matrices con un patrón de sparsidad dado el cual puede construirse iterativamente aumentando progresivamente el conjunto de entradas no nulas en cada columna.

3.1. Resultados teóricos

Partiendo del sistema de ecuaciones lineales (2.1) donde A es una matriz de orden elevado, *sparse* y no singular. Como se ha expuesto en el Capítulo anterior (2), en general, la convergencia de los métodos iterativos basados en subespacios de Krylov no está asegurada o puede ser muy lenta. Para mejorar su comportamiento, consideramos una matriz de precondicionamiento M y transformamos el sistema (2.1) en cualquiera de los siguientes problemas equivalentes,

$$MAx = Mb \tag{3.1}$$

$$AMy = b \; ; \; x = My \tag{3.2}$$

es decir, sistemas precondicionados por la izquierda o por la derecha, respectivamente (ver [87, 97]). Usaremos aquí el precondicionamiento por la derecha y M debe seleccionarse tal que AM esté cercana a la identidad en cierto sentido. Esta cercanía puede medirse usando normas matriciales, por ejemplo, aquellas normas inducidas por las normas p $(1 \leq p \leq \infty)$ de $\mathbb{R}^n$ o la norma Frobenius.

Hemos preferido la norma Frobenius por dos razones. La primera es teórica. La norma Frobenius es prehilbertiana, lo que no ocurre con otras normas usuales como $||\cdot||_1$, $||\cdot||_2$ o $||\cdot||_\infty$. Esto nos permite usar la Teoría de Aproximación en espacios prehilbertianos. La segunda razón es de naturaleza computacional. Las columnas de M pueden calcularse y usarse independientemente, es decir, en paralelo [38, 41, 61], ya que

$$||AM - I||_F^2 = \sum_{j=1}^{n} ||(AM - I)\, e_j||_2^2; \quad e_j = (0, ..., 0, \overset{j}{\check{1}}, 0, ..., 0)^T \tag{3.3}$$

Así, para un subespacio vectorial $\mathcal{S}$ de $\mathcal{M}_n(\mathbb{R})$, el problema a resolver es

$$\min_{M \in \mathcal{S}} ||AM - I||_F = ||AN - I||_F \tag{3.4}$$

y se considera N un buen precondicionador del sistema (2.1) si es una inversa aproximada de A (con respecto a la norma Frobenius) en el sentido estricto, es decir, $||AN - I||_F < 1$. En general, esta condición no se satisface. De hecho, para una matriz dada A, siempre es posible encontrar un subespacio $\mathcal{S}$ que no contenga ninguna inversa aproximada de A estrictamente.

En el siguiente teorema se resume la conveniencia de usar una aproximada inversa como precondicionador [8, 71].

Teorema 29 *Sea $A, M \in M_n(R)$ tal que $||AM - I||_F < 1$. Entonces,*

(i) Mes no singular y $M^{-1} = \left[\sum_{m=0}^{\infty} (I - AM)^m\right] A$

(ii) AMes definida positiva

(iii) $\kappa_2(AM) \leq \dfrac{1 + ||AM - I||_F}{1 - ||AM - I||_F}$

donde $\kappa_2(AM)$ representa el número de condición de la matriz AM relativo a la norma 2.

Como punto de partida, consideremos la norma Frobenius con pesos,

$$\forall A \in \mathcal{M}_n(\mathbb{R}) : ||A||_W^2 = tr(AWA^t)\,, (W \text{ simétrica y definida positiva})$$

Sea $K \subset \{1, 2, ..., n\} \times \{1, 2, ..., n\}$ el patrón de *sparsidad* de M. Definimos el espacio de matrices con esa estructura *sparse* como, y

$$\mathcal{S} = \{M \in \mathcal{M}_n(\mathbb{R}) / m_{i,j} = 0; \forall (i,j) \notin K\}$$

Axelsson [8] estudia el problema precondicionado por la izquierda

$$\min_{M \in \mathcal{S}} ||MA - I||_W^2 = ||N^*A - I||_W^2 \tag{3.5}$$

y lo convierte en,

$$\left(N^*AWA^t\right)_{i,j} = \left(WA^t\right)_{i,j}; \forall (i,j) \in K \tag{3.6}$$

$$||N^*A - I||_W^2 = tr\left[(I - N^*A)W\right] \tag{3.7}$$

De forma similar, se resuelve el problema precondicionado por la derecha

$$\min_{M \in \mathcal{S}} ||AM - I||_W^2 = ||AN - I||_W^2 \tag{3.8}$$

que da lugar a,

$$\left(A^tANW\right)_{i,j} = \left(A^tW\right)_{i,j}; \forall (i,j) \in K \tag{3.9}$$

$$||AN - I||_W^2 = tr\left[(I - AN)W\right] \tag{3.10}$$

En particular, para $W = I$,

$$(A^tAN)_{i,j} = a_{ji}\ ;\ \forall (i,j) \in K \tag{3.11}$$

$$||AN - I||_F^2 = n - tr(AN) \tag{3.12}$$

Nuestro próximo objetivo es generalizar las ecuaciones (3.11) y (3.12) para un subespacio arbitrario $\mathcal{S}$ de $\mathcal{M}_n(\mathbb{R})$ (sección 3.1.1). A continuación, en la sección 3.2 obtenemos las expresiones explícitas para la matriz N y $||AN - I||_F^2$ de una base dada del subespacio $\mathcal{S}$, caracterizando aquellas que nos dan inversas aproximadas. Finalmente, en la sección 3.3 aplicamos estos resultados a algunos precondicionadores usuales (diagonal, *sparse*, simétricos y otros).

3.1.1. Mejor aproximación y producto escalar de Frobenius

En general, una norma matricial inducida no es prehilbertiana, ni aún cuando las normas vectoriales relacionadas lo sean. En efecto, esto puede demostrarse con el siguiente contraejemplo para la norma 2,

$$A = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}; B = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}$$

ya que

$$\|A\|_2 = \|B\|_2 = \|A + B\|_2 = \|A - B\|_2 = 1$$

Sin embargo, la norma de Frobenius con pesos trivialmente proviene del producto escalar

$$\langle A, B\rangle_W = tr(AWB^t), \forall A, B \in \mathcal{M}_n(\mathbb{R}) \tag{3.13}$$

y, en particular, para $W = I$, $\|\cdot\|_F$ es prehilbertiana y,

$$\langle A, B\rangle_F = tr(AB^t) = \sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij}b_{ij}, \forall A, B \in \mathcal{M}_n(\mathbb{R}) \tag{3.14}$$

En lo que sigue, el término ortogonalidad se referirá al producto escalar de Frobenius $\langle\cdot,\cdot\rangle_F$. Entonces, el problema (3.4) definido en el espacio Prehilbertiano $(\mathcal{M}_n(\mathbb{R}), \langle\cdot,\cdot\rangle_F)$ se reduce a encontrar la proyección ortogonal de la matriz I sobre el subespacio $A\mathcal{S}$.

Teorema 30 *Sea $A \in M_n(R)$ una matriz no singular y S un subespacio vectorial de $M_n(R)$. Entonces, la solución al problema (3.4) será,*

$$tr\left(A^tANM^t\right) = tr\left(AM\right) \ ; \ \forall M \in \mathcal{S} \tag{3.15}$$

y,

$$\|AN - I\|_F^2 = n - tr(AN) \tag{3.16}$$

Demostración. Como $\dim(A\mathcal{S}) = \dim(\mathcal{S}) < \infty$, el teorema de la mejor aproximacion en subconjuntos cerrados y convexos asegura la existencia y unicidad de la solución al problema (3.4) y el teorema de la proyección ortogonal caracteriza la mejor aproximación AN de I en el subespacio $A\mathcal{S}$ por la condición

$$\langle AN - I, AM\rangle_F = 0 \ ; \ \forall M \in \mathcal{S} \tag{3.17}$$

que es equivalente a,

$$tr(A^tANM^t) = tr(AM) \ ; \ \forall M \in \mathcal{S} \tag{3.18}$$

Además,

$$\|AN - I\|_F^2 = \langle I - AN, I\rangle_F + \langle AN - I, AN\rangle_F = n - tr(AN)$$

■

El teorema anterior generaliza los resultados (3.11) y (3.12), obtenidos para el subespacio de las matrices con un patrón de sparsidad dado $K \subset \{1, 2, ..., n\}\times\{1, 2, ..., n\}$, en cualquier subespacio vectorial de $\mathcal{M}_n(\mathbb{R})$.

3.2. $\mathcal{S}$-Inversa generalizada y complemento ortogonal de Frobenius

3.2.1. $\mathcal{S}$-Inversa generalizada

En esta sección generalizamos la obtención del mejor precondicionador (en norma Frobenius) en un subespacio arbitrario $\mathcal{S}$ de $\mathcal{M}_n(\mathbb{R})$, es decir, la $\mathcal{S}$-inversa generalizada para la matriz A del sistema lineal (2.1), donde A es una matriz no singular, *sparse* y no necesariamente simétrica.

Sea $\mathcal{S}$ un subespacio vectorial de $\mathcal{M}_n(\mathbb{R})$. Sea

$$A = U\Sigma V^T \; ; \; \Sigma = \begin{pmatrix} \sigma_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma_n \end{pmatrix}$$

una descomposición en valores singulares de la matriz A. Entonces, la solución al problema de optimización (3.4) viene dada por:

$$N = VQU^T \; ; \; \min_{P \in V^T \mathcal{S} U} ||\Sigma P - I||_F = ||\Sigma Q - I||_F \tag{3.19}$$

$$||AN - I||_F = ||\Sigma Q - I||_F \tag{3.20}$$

Procedamos así:

$$A = U\Sigma V^T; \tag{3.21}$$
$$A^T = V\Sigma U^T \tag{3.22}$$
$$AA^T = U\Sigma^2 U^T \tag{3.23}$$
$$A^T A = V\Sigma^2 V^T \tag{3.24}$$

Entonces, el problema de minimización (3.4) resulta,

$$\begin{aligned} \min_{M \in \mathcal{S}} ||AM - I||_F &= \left\{ \begin{array}{l} A = U\Sigma V^T \\ M = VXU^T \end{array} \right\} \\ &= \min_{X \in V^T \mathcal{S} U} \left\| U\Sigma V^T VXU^T - I \right\|_F \\ &= \min_{X \in V^T \mathcal{S} U} \left\| U\Sigma XU^T - I \right\|_F \\ &= \min_{X \in V^T \mathcal{S} U} \left\| U\Sigma V^T VXU^T - UU^T \right\|_F \\ &= \min_{X \in V^T \mathcal{S} U} \left\| \Sigma X - I \right\|_F \end{aligned} \tag{3.25}$$

Así,

$$\min_{M \in \mathcal{S}} ||AM - I||_F = \min_{X \in V^T \mathcal{S} U} \left\| \Sigma X - I \right\|_F \tag{3.26}$$
$$||AN - I||_F = \left\| \Sigma Y - I \right\|_F \tag{3.27}$$

Luego,

$$A = U\Sigma V^T \Rightarrow N = VYU^T \text{ con } \min_{X \in V^T SU} \|\Sigma X - I\|_F = \|\Sigma Y - I\|_F \tag{3.28}$$

3.2.2. Complemento ortogonal de Frobenius

Resolver el problema de minimización (3.4) consiste en encontrar una matriz $AN - I$ que sea ortogonal a la matriz AM en el sentido de la norma de Frobenius, es decir $< AN - I, AM >_F = 0$, $M \in S$, esto es, $AN - I \in (AS)^{\perp}$.

Proposición 31

$$\forall \mathcal{S} \subseteq \mathcal{M}_n(\mathbb{R}) : (AS)^{\perp} = \left(A^{-1}\right)^T S^{\perp} \tag{3.29}$$

En consecuencia,

$$AN - I \in (AS)^{\perp} = \left(A^{-1}\right)^T S^{\perp} \Rightarrow AN - I = \left(A^{-1}\right)^T Q; Q \in S^{\perp} \Rightarrow \tag{3.30}$$

$$A^T AN = A^T + Q; Q \in S^{\perp} \tag{3.31}$$

lo que constituye la Ecuación Normal Generalizada.

Sea $\dim(S) = p$ y $\{E_1, ..., E_p\}$ una base de S, entonces,

$$\langle A^T AN, E_i \rangle_F = \langle A^T, E_i \rangle_F + \langle Q, E_i \rangle_F ; \forall i = 1, 2, ..., p \tag{3.32}$$

Por tanto,

$$\langle A^T AN, E_i \rangle_F = \langle A^T, E_i \rangle_F ; \forall i = 1, 2, ..., p \tag{3.33}$$

constituyen las nuevas Ecuaciones Normales Generalizadas.

En conclusión, N y $\|AN - I\|_F$ pueden determinarse por la ecuación normal a partir de $S^{\perp}$

A continuación presentamos el complemento ortogonal de algunos subespacios particulares.

3.2.2.1. Subespacio de las matrices cuadradas

$S = \mathcal{M}_n(\mathbb{R}) \Longrightarrow S^{\perp} = \{0\}$

$$A^T AN = A^T \Longrightarrow N = A^{-1} \text{ y } A^{-1} \in S \tag{3.34}$$

3.2.2.2. Subespacio de las matrices *sparse*

$S = \{M \in \mathcal{M}_n(\mathbb{R})/m_{ij} = 0; \forall (i,j) \notin K\}$, $K \subset \{1,2,...,n\} \times \{1,2,...,n\}$
$\Longrightarrow S^{\perp} = \{Q \in \mathcal{M}_n(\mathbb{R})/q_{ij} = 0; \forall (i,j) \in K\}$

$$A^T AN = A^T + Q \tag{3.35}$$

$$\left\langle A^T AN, E_{i,j} \right\rangle_F = \left\langle A^T, E_{i,j} \right\rangle_F ; \quad \forall (i,j) \in K \tag{3.36}$$

$$\left((A^T A)N\right)_{ij} = a_{ij} \qquad \forall (i,j) \in K \tag{3.37}$$

3.2.2.3. Subespacio de las matrices simétricas

$S = \mathcal{S}_n(\mathbb{R}) \Longrightarrow S^{\perp} = \mathcal{H}_n(\mathbb{R})$

$$A^T AN = A^T + H \tag{3.38}$$

$$NA^T A = A - H \tag{3.39}$$

Sumando ambas ecuaciones tenemos,

$$A^T AN + NA^T A = A + A^T \tag{3.40}$$

3.2.2.4. Subespacio de las matrices hemisimétricas

$S = \mathcal{H}_n(\mathbb{R}) \Longrightarrow S^{\perp} = \mathcal{S}_n(\mathbb{R})$

$$A^T AN = A^T + S \tag{3.41}$$

$$-NA^T A = A + S \tag{3.42}$$

Sumando ambas ecuaciones tenemos,

$$A^T AN + NA^T A = A^T - A \tag{3.43}$$

3.2.2.5. Subespacio de las matrices que simetrizan a A

$S = \left\{M \in \mathcal{M}_n(\mathbb{R})/\,(AM)^T = AM\right\} \Longrightarrow AS = \mathcal{S}_n \Longrightarrow (AS)^{\perp} = \mathcal{H}_n$

$$AN - I \in (AS)^{\perp} \tag{3.44}$$

$$AN - I = H \tag{3.45}$$

$$AN - I = -H \tag{3.46}$$

Sumando ambas ecuaciones tenemos,

$$2\,(AN - I) = 0 \Longrightarrow AN - I = 0 \Longrightarrow N = A^{-1} \text{ y } A^{-1} \in S \tag{3.47}$$

3.2.2.6. Subespacio de las matrices que antisimetrizan a A

$$S = \left\{ M \in \mathcal{M}_n(\mathbb{R}) / \, (AM)^T = -AM \right\} \Longrightarrow AS = \mathcal{H}_n \Longrightarrow (AS)^{\perp} = \mathcal{S}_n$$

$$AN - I \in (AS)^{\perp} \tag{3.48}$$

$$AN - I = S \tag{3.49}$$

$$-AN - I = S \tag{3.50}$$

Restando ambas ecuaciones tenemos,

$$2AN = 0 \Longrightarrow N = 0 \tag{3.51}$$

3.2.2.7. Subespacio de las matrices simétricas que simetrizan a A

$$S = \left\{ M \in \mathcal{M}_n(\mathbb{R}) / M^T = M \text{ y } (AM)^T = AM \right\} \Longrightarrow$$

$$\Longrightarrow AS = \left\{ AM \in \mathcal{M}_n(\mathbb{R}) / M^T = M \text{ y } (AM)^T = AM \right\} = A\mathcal{S}_n \cap \mathcal{S}_n \tag{3.52}$$

$$AN - I \in (AS)^{\perp} = (A\mathcal{S}_n \cap \mathcal{S}_n)^{\perp} = (A\mathcal{S}_n)^{\perp} + (\mathcal{S}_n)^{\perp} = \mathcal{H}_n A + \mathcal{H}_n \tag{3.53}$$

$$AN - I = H_1 A + H_2 \tag{3.54}$$

De otro modo,

$$AN - I \in (AS)^{\perp} = (A\mathcal{S}_n \cap \mathcal{S}_n)^{\perp} = \left(A^{-1}\right)^T \mathcal{H}_n + \mathcal{H}_n \tag{3.55}$$

$$AN - I = \left(A^{-1}\right)^T H_1 + H_2 \tag{3.56}$$

$$A^T AN = A^T + H_1 + A^T H_2 \tag{3.57}$$

3.3. Expresiones explícitas para los mejores precondicionadores

Otra forma de determinar N y $\|AN - I\|_F$ es a partir de las expresiones explícitas usando una base adecuada del subespacio $\mathcal{S}$ de precondicionadores.

Teorema 32 *Sea $A \in \mathcal{M}_n(\mathbb{R})$ una matriz no singular, sea $\mathcal{S}$ un subespacio vectorial de $\mathcal{M}_n(\mathbb{R})$ de dimensión p, y $\{M_1, ..., M_p\}$ una base de $\mathcal{S}$ tal que $\{AM_1, ..., AM_p\}$ es una base ortogonal de $A\mathcal{S}$. Entonces, la solución al problema (3.4) es*

$$N = \sum_{i=1}^{p} \frac{tr(AM_i)}{\|AM_i\|_F^2} M_i \tag{3.58}$$

$$\|AN - I\|_F^2 = n - \sum_{i=1}^{p} \frac{[tr(AM_i)]^2}{\|AM_i\|_F^2} \tag{3.59}$$

Demostración. AN es la proyección ortogonal de I sobre el subespacio $A\mathcal{S}$. En efecto, si AN se representa por la suma de Fourier de la identidad relacionada con el sistema ortogonal $\left\{\frac{AM_i}{\|AM_i\|_F}\right\}_{i=1}^{p}$, obtenemos

$$AN = \sum_{i=1}^{p} \frac{\langle I, AM_i\rangle_F}{\|AM_i\|_F^2} AM_i = \sum_{i=1}^{p} \frac{tr(AM_i)}{\|AM_i\|_F^2} AM_i \Longrightarrow N = \sum_{i=1}^{p} \frac{tr(AM_i)}{\|AM_i\|_F^2} M_i$$

y también, de la ecuación (3.16)

$$\|AN - I\|_F^2 = n - tr\left[\sum_{i=1}^{p} \frac{tr(AM_i)}{\|AM_i\|_F^2} AM_i\right] = n - \sum_{i=1}^{p} \frac{[tr(AM_i)]^2}{\|AM_i\|_F^2}$$

■

Este teorema puede generalizarse para cualquier base del subespacio $\mathcal{S}$.

Teorema 33 *Sea $A \in \mathcal{M}_n(\mathbb{R})$ una matriz no singular, sea $\mathcal{S}$ un subespacio vectorial de $\mathcal{M}_n(\mathbb{R})$ de dimensión p y $\{M_1, ..., M_p\}$ una base de $\mathcal{S}$. Entonces, la solución al problema (3.4) es*

$$N = \sum_{i=1}^{p} \frac{\det(T_i)}{\det(G_{i-1})\det(G_i)} \widetilde{M}_i \tag{3.60}$$

$$\|AN - I\|_F^2 = n - \sum_{i=1}^{p} \frac{[\det(T_i)]^2}{\det(G_{i-1})\det(G_i)} \tag{3.61}$$

donde, $\det(G_0) = 1$, *G_i es la matriz de Gram del sistema $\{AM_1, ..., AM_i\}$ con respecto al producto escalar de Frobenius, T_i es la matriz que resulta de reemplazar la última fila de G_i por $tr(AM_1), ..., tr(AM_i)$, y $\widetilde{M}_i$ es la matriz obtenida evaluando el determinante simbólico que resulta de remplazar la última fila de* $\det(G_i)$ *por $M_1, ..., M_i$, con $1 \leq i \leq p$.*

Demostración. El algoritmo de Gram-Schmidt aplicado a la base $\{AM_r\}_{r=1}^{p}$ de $A\mathcal{S}$ nos permite obtener la base ortogonal $\left\{A\widetilde{M}_r\right\}_{r=1}^{p}$. Además, denotando

$$\alpha_{ij} = \langle AM_i, AM_j\rangle_F \ ; \ \widetilde{M}_i = \begin{vmatrix} \alpha_{11} & \alpha_{12} & \cdots & \alpha_{1i} \\ \vdots & \vdots & & \vdots \\ \alpha_{(i-1)1} & \alpha_{(i-1)2} & \cdots & \alpha_{(i-1)i} \\ M_1 & M_2 & \cdots & M_i \end{vmatrix}$$

tenemos

$$\left\|A\widetilde{M}_1\right\|_F^2 = \alpha_{11} = \det(G_0)\det(G_1) \tag{3.62}$$

y como,

$$\forall\ i = 2,...,p,\ \ \forall j < i \left\langle AM_j, A\widetilde{M}_i \right\rangle_F = \begin{vmatrix} \alpha_{11} & \alpha_{12} & \cdots & \alpha_{1i} \\ \vdots & \vdots & & \vdots \\ \alpha_{(i-1)1} & \alpha_{(i-1)2} & \cdots & \alpha_{(i-1)i} \\ \alpha_{j1} & \alpha_{j2} & \cdots & \alpha_{ji} \end{vmatrix} = 0$$

entonces, $\forall\ i = 2,...,p$

$$\left\|A\widetilde{M}_i\right\|_F^2 = \left\langle \det(G_{i-1})\, AM_i, A\widetilde{M}_i \right\rangle_F = \det(G_{i-1})\det(G_i) \tag{3.63}$$

Por otra parte, $\forall\ i = 1,...,p$

$$tr\left(A\widetilde{M}_i\right) = tr\left(\begin{vmatrix} \alpha_{11} & \alpha_{12} & \cdots & \alpha_{1i} \\ \vdots & \vdots & & \vdots \\ \alpha_{(i-1)1} & \alpha_{(i-1)2} & \cdots & \alpha_{(i-1)i} \\ AM_1 & AM_2 & \cdots & AM_i \end{vmatrix}\right) = \det(T_i) \tag{3.64}$$

Finalmente, aplicando el teorema 32 a la base $\left\{\widetilde{M}_i\right\}_{i=1}^{p}$ de $\mathcal{S}$, este se transforma en las ecuaciones (3.60) y (3.61). ■

Atendiendo al coste computacional, el teorema 33 presenta el problema del cálculo de los determinantes $\det(G_i)$, $\det(T_i)$ y $\widetilde{M}_i$. El mayor orden de estos puede reducirse considerablemente con una adecuada descomposición del subespacio $\mathcal{S}$ como suma directa de subespacios.

Teorema 34 *Sea $A \in M_n(\mathbb{R})$ una matriz no singular, sea $\mathcal{S}$ un subespacio vectorial de $M_n(\mathbb{R})$ de dimensión p, $\mathcal{S} = \mathcal{S}_1 \oplus ... \oplus \mathcal{S}_q$ con $\dim \mathcal{S}_j = p_j$ y $\left\{M_1^j, ..., M_{p_j}^j\right\}$ una base de $\mathcal{S}_j$; $\forall j = 1,...,q$. Supongamos que los subespacios $A\mathcal{S}_j$ $(1 \le j \le q)$ son mutuamente ortogonales. Entonces, la solución al problema (3.4) es*

$$N = \sum_{j=1}^{q}\sum_{k=1}^{p_j} \frac{\det\left(T_k^j\right)}{\det\left(G_{k-1}^j\right)\det\left(G_k^j\right)} \widetilde{M}_k^j \tag{3.65}$$

$$\|AN - I\|_F^2 = n - \sum_{j=1}^{q}\sum_{k=1}^{p_j} \frac{\left[\det\left(T_k^j\right)\right]^2}{\det\left(G_{k-1}^j\right)\det\left(G_k^j\right)} \tag{3.66}$$

donde, $\forall j = 1,...,q$: G_k^j, T_k^j y $\widetilde{M}_k^j$ son los mismos del teorema 33 para la base $\left\{M_i^j\right\}_{i=1}^{p_j}$ de $\mathcal{S}_j$.

Demostración. Del teorema 33, la solución al problema (3.4) para cada subespacio $\mathcal{S}_j$ $(1 \leq j \leq q)$ es

$$N_j = \sum_{k=1}^{p_j} \frac{\det\left(T_k^j\right)}{\det\left(G_{k-1}^j\right)\det\left(G_k^j\right)} \widetilde{M}_k^j \tag{3.67}$$

$$\|AN_j - I\|_F^2 = n - \sum_{k=1}^{p_j} \frac{\left[\det\left(T_k^j\right)\right]^2}{\det\left(G_{k-1}^j\right)\det\left(G_k^j\right)} \tag{3.68}$$

Por otra parte, de la Teoría de Aproximación en Espacios Prehilbertianos, asumiendo la ortogonalidad mutua de los subespacios $A\mathcal{S}_j$ $(1 \leq j \leq q)$, tenemos,

$$AN = \sum_{j=1}^{q} AN_j \tag{3.69}$$

$$\|AN - I\|_F^2 = \sum_{j=1}^{q} \|AN_j - I\|_F^2 - (q-1)\|I\|_F^2 \tag{3.70}$$

Insertando las ecuaciones (3.67) y (3.68) en las ecuaciones (3.69) y (3.70) respectivamente, se concluye la demostración. ■

La aplicación del teorema 34, en lugar del teorema 33, reduce el máximo orden de los determinantes a calcular de $p = \sum_{j=1}^{q} p_j$ a $\max(p_1, ..., p_q)$.

Los cálculos que se efectúan en el subespacio $\mathcal{S}$ atendiendo a los teoremas 32, 33, 34, son útiles cuando la base se aumenta con una nueva matriz, añadiendo los términos correspondientes de la suma.

Como consecuencia directa de los teoremas 32, 33 y 34, caracterizamos aquellas matrices no singulares cuyos precondicionadores, construidos en un subespacio dado $\mathcal{S}$ nos dan aproximadas inversas. En efecto, basta con tener en cuenta que $\mathcal{S}$ contiene al menos una inversa aproximada de A si, y sólo si, $\|AN - I\|_F < 1$.

3.4. Aplicación a algunos precondicionadores usuales

Los resultados anteriores pueden aplicarse a los precondicionadores que se utilizan frecuentemente para resolver sistemas de ecuaciones lineales.

3.4.1. Precondicionador con patrón de sparsidad dado

Se ha estudiado por muchos autores una inversa aproximada con un patrón de *sparsidad* fijo (ver [73]). Aunque nuestro camino está definido por un

patrón de *sparsidad* fijo, esto también nos permite capturar el mejor patrón de sparsidad para un número dado de entradas no nulas. Sea K un subconjunto de $\{1,2,...,n\} \times \{1,2,...,n\}$ y

$$\mathcal{S} = \{M \in \mathcal{M}_n(\mathbb{R}) / m_{i,j} = 0; \forall (i,j) \notin K\}$$

En lo que sigue, denotaremos por J el conjunto de índices de columna j para el cual existe al menos un índice de fila i tal que $(i,j) \in K$, y

$$K = \underset{j \in J}{\cup} \left\{ \left(i_1^j, j\right), \left(i_2^j, j\right), ..., \left(i_{p_j}^j, j\right) \right\}, 1 \leq i_1^j < i_2^j < ... < i_{p_j}^j \leq n$$

y sea $M_{i,j}$ una matriz de orden $n \times n$, cuya única entrada no nula es $m_{ij} = 1$.

Con esta notación, se resume a continuación la solución al problema (3.4) para un patrón de *sparsidad* K.

Teorema 35 *Sea $A \in M_n(\mathbb{R})$ una matriz no singular y sea S el subespacio de las matrices con patrón de sparsidad K. Entonces, la solución al problema (3.4) es*

$$N = \sum_{j \in J} \sum_{k=1}^{p_j} \frac{\det\left(D_k^j\right)}{\det\left(G_{k-1}^j\right) \det\left(G_k^j\right)} \widetilde{M}_{k,j} \tag{3.71}$$

$$||AN - I||_F^2 = n - \sum_{j \in J} \sum_{k=1}^{p_j} \frac{\left[\det\left(D_k^j\right)\right]^2}{\det\left(G_{k-1}^j\right) \det\left(G_k^j\right)} \tag{3.72}$$

donde, $\forall j \in J$, $\det\left(G_0^j\right) = 1$ y G_k^j es la matriz de Gram de las columnas $i_1^j, i_2^j, ..., i_k^j$ de la matriz A con respecto al producto escalar euclídeo, D_k^j es la matriz que resulta de reemplazar la última fila de G_k^j por $a_{j\,i_1^j}, a_{j\,i_2^j}, ..., a_{j\,i_k^j}$, y $\widetilde{M}_{k,j}$ es la matriz que se obtiene evaluando el determinante simbólico que resulta de reemplazar la última fila de $\det\left(G_k^j\right)$ por $M_{i_1^j,j}, M_{i_2^j,j}, ..., M_{i_k^j,j}$, con $1 \leq k \leq p_j$.

Demostración. Obviamente, $p = \dim(\mathcal{S}) = \sum_{j \in J} p_j$ y una base de $\mathcal{S}$ es

$$\underset{j \in J}{\cup} \left\{ M_{i_1^j,j}, M_{i_2^j,j}, ..., M_{i_{p_j}^j,j} \right\}$$

Ahora, si denotamos

$$\mathcal{S}_j = span\left\{ M_{i_1^j,j}, M_{i_2^j,j}, ..., M_{i_{p_j}^j,j} \right\}; \forall j \in J$$

entonces $\mathcal{S} = \underset{j \in J}{\oplus} \mathcal{S}_j$ y los subespacios $A\mathcal{S}_j$ son mutuamente ortogonales, esto es

$$\forall (i,j), (i',j') \in K,\ j \neq j' : \langle AM_{i,j}, AM_{i',j'} \rangle_F = 0$$

ya que la única columna no nula de la matriz $AM_{i,j}$ es la j-ésima que coincide con la i-ésima columna de la matriz A. En efecto, aplicando el teorema 34 al subespacio $\mathcal{S}$ y teniendo en cuenta

$$\forall (i,j),(i',j) \in K : \langle AM_{i,j}, AM_{i',j} \rangle_F = \langle Ae_i, Ae_{i'} \rangle_2 \tag{3.73}$$

$$\forall (i,j) \in K : tr\,(AM_{i,j}) = a_{ji} \tag{3.74}$$

obtenemos las ecuaciones (3.71) y (3.72). ■

La condición $J = \{1, 2, ..., n\}$ es necesaria para que $\mathcal{S}$ contenga una inversa aproximada de la matriz A puesto que, de otra manera, para toda $M \in \mathcal{S}$, M es singular y, en efecto, $\|AM - I\| \geq 1$ debido al teorema 29. Por la misma razón, si K no contiene ningún par (i_0, i) para un $i_0 \in \{1, 2, ..., n\}$ dado, entonces $\mathcal{S}$ no contiene ninguna inversa aproximada de la matriz A.

Como caso particular de precondicionador con patrón de sparsidad dado, podemos considerar la matriz en banda con ancho de semibanda izquierda q_1 y ancho de semibanda derecha q_2 definida por,

$$K = \{(i,j) \in \{1, 2, ..., n\} \times \in \{1, 2, ..., n\} / j \geq i - q_1 \text{ y } j \leq i + q_2\} \tag{3.75}$$

Para más detalles sobre precondicionadores tipo inversa aproximada en banda, ver [60]. El caso diagonal ($q_1 = q_2 = 0$) se detalla a continuación.

3.4.2. Precondicionador diagonal

La solución al problema (3.4) cuando $\mathcal{S}$ es el subespacio de las matrices diagonales de orden n es bien conocido [8], y puede obtenerse de las ecuaciones (3.71) y (3.72) para

$$p_j = 1 \text{ y } i_1^j = j, \forall j \in J = \{1, 2, ..., n\}$$

Sin embargo, la solución puede obtenerse por un camino más simple.

Teorema 36 *Sea $A \in M_n(\mathbb{R})$ una matriz no singular y sea $\mathcal{S}$ el subespacio de las matrices diagonales de orden n. Entonces, la solución al problema (3.4) es*

$$N = diag\left(\frac{a_{11}}{\|Ae_1\|_2^2}, \frac{a_{22}}{\|Ae_2\|_2^2}, ..., \frac{a_{nn}}{\|Ae_n\|_2^2}\right) \tag{3.76}$$

$$\|AN - I\|_F^2 = n - \sum_{i=1}^{n} \frac{a_{ii}^2}{\|Ae_i\|_2^2} \tag{3.77}$$

Demostración. $\{M_{1,1},...,M_{n,n}\}$ es una base de $\mathcal{S}$ tal que $\{AM_{1,1},...,AM_{n,n}\}$ es ortogonal. En efecto, del teorema 32 y las ecuaciones (3.73), (3.74), se concluye la demostración. ■

3.4.3. Precondicionador simétrico y hemisimétrico

Se estudia aquí el mejor precondicionador simétrico del complemento ortogonal de AS, como una alternativa de los teoremas establecidos en la sección 3.3 usando una base del subespacio S. Denotemos por $\mathcal{S}_n$ y $\mathcal{H}_n$ los subespacios de las matrices simétricas y hemisimétricas de orden n, respectivamente. Entonces, $\forall M \in \mathcal{S}_n, \forall H \in \mathcal{H}_n$, tenemos

$$\langle HA, AM\rangle_F = tr(HAMA^t) = -\langle AM, HA\rangle_F \Rightarrow \langle HA, AM\rangle_F = 0$$

y también

$$\dim(\mathcal{H}_n A) = \dim((A\mathcal{S}_n)^{\perp})$$

Por consiguiente,

$$(A\mathcal{S}_n)^{\perp} = \mathcal{H}_n A$$

Así, usando la ecuación (3.17) el mejor precondicionador simétrico N del sistema (2.1) está dado por

$$AN - I = HA,\ H \in \mathcal{H}_n$$

lo que nos lleva a

$$A^tAN + NA^tA = A + A^t \tag{3.78}$$

Esta ecuación matricial tiene una, y sólo una, solución N, debido a la unicidad de la solución del problema (3.4) para $\mathcal{S} = \mathcal{S}_n$.

Sean ahora $\sigma_1 \geq \sigma_2 \geq ... \geq \sigma_n > 0$ los valores singulares de A, es decir

$$A^tA = V\Sigma^2V^t$$

con $\Sigma = diag(\sigma_1, \sigma_2, ..., \sigma_n)$ y V ortogonal. Entonces, la ecuación (3.78) se transforma en

$$V^tNV = \left(V^t(A + A^t)V\right) \odot P$$

donde $p_{ij} = \frac{1}{\sigma_i^2 + \sigma_j^2}$, $\forall\, i, j = 1, ..., n$, y $\odot$ denota el producto matricial de Hadamard (entradas pareadas).

Finalmente, el mejor precondicionador simétrico del sistema (2.1) es

$$N = V\left[\left(V^t(A + A^t)V\right) \odot P\right]V^t \tag{3.79}$$

y el siguiente teorema nos da una estimación de $||AN - I||_F$.

Teorema 37 *Sea $A \in M_n(\mathbb{R})$ una matriz no singular y sea N el mejor precondicionador simétrico del sistema (2.1), entonces*

$$\frac{1}{2}\left\|A - A^t\right\|_F \left\|A\right\|_2^{-1} \leq ||AN - I||_F \leq \frac{1}{2}\left\|A - A^t\right\|_F \left\|A^{-1}\right\|_2 \tag{3.80}$$

Demostración. Sea $A = U\Sigma V^t$ la descomposición en valores singulares de A. Usando la ecuación (3.79),

$$\begin{aligned}
||AN - I||_F^2 &= \left\|U\Sigma\left[\left(V^t(A+A^t)V\right)\odot P\right]V^t - I\right\|_F^2 \\
&= \left\|U\left[\Sigma\left(V^t(A+A^t)V\right)\odot P - U^tV\right]V^t\right\|_F^2 \\
&= \left\|\left(\Sigma V^tU\Sigma + \Sigma^2U^tV\right)\odot P - \left(\Sigma^2U^tV + U^tV\Sigma^2\right)\odot P\right\|_F^2 \\
&= \left\|\left[\left(\Sigma V^tU - U^tV\Sigma\right)\Sigma\right]\odot P\right\|_F^2 \\
&= \left\|\left[\left(U^t\left(A-A^t\right)U\right)\Sigma\right]\odot P\right\|_F^2 \\
&= \sum_{i,j}\left[\frac{\left(U^t\left(A-A^t\right)U\right)_{ij}\sigma_j}{\sigma_i^2+\sigma_j^2}\right]^2 = \sum_{i<j}\frac{\left(U^t\left(A-A^t\right)U\right)_{ij}^2}{\sigma_i^2+\sigma_j^2}
\end{aligned}$$

Por lo tanto,

$$\frac{1}{2\sigma_1^2}\sum_{i<j}\left(U^t\left(A-A^t\right)U\right)_{ij}^2 \le ||AN-I||_F^2 \le \frac{1}{2\sigma_n^2}\sum_{i<j}\left(U^t\left(A-A^t\right)U\right)_{ij}^2$$

y como

$$\sum_{i<j}\left(U^t\left(A-A^t\right)U\right)_{ij}^2 = \frac{1}{2}||A-A^t||_F^2$$

obtenemos

$$\frac{1}{4\sigma_1^2}||A-A^t||_F^2 \le ||AN-I||_F^2 \le \frac{1}{4\sigma_n^2}||A-A^t||_F^2$$

■

En el siguiente corolario se resumen algunas consecuencias directas de este teorema

Corolario 38 *(i) Si* $2\|A\|_2 \le \left\|A-A^T\right\|_F$, *entonces no hay ninguna inversa aproximada simétrica de* A,
(ii) si $k_2(A) \simeq 1$, *entonces* $\frac{1}{2}\left\|A-A^T\right\|_F\|A\|_2^{-1} \simeq \|AN-I\|_F \simeq \frac{1}{2}\left\|A-A^T\right\|_F\|A^{-1}\|_2$,
(iii) $\frac{1}{2}\left\|A-A^T\right\|_F\|A\|_F^{-1} \simeq \|AN-I\|_F \simeq \frac{1}{2}\left\|A-A^T\right\|_F\|A^{-1}\|_F$.

El corolario 38 afirma que las cotas en (3.80) están más cercanas cuanto más se acerca el número de condición $k_2(A)$ a la unidad. Como un caso particular, si A es ortogonal, entonces

$$\|AN-I\|_F = \frac{1}{2}\left\|A-A^T\right\|_F \text{ y } N = \frac{1}{2}\left(A+A^T\right) \tag{3.81}$$

Estos resultados teóricos pueden obtenerse también aplicando el teorema 32.

Siguiendo el mismo procedimiento que para el caso simétrico, el mejor precondicionador hemisimétrico del sistema (2.1) es

$$N = V\left[\left(V^T\left(A^T-A\right)V\right)\odot P\right]V^T \tag{3.82}$$

3.4.4. Precondicionador M tal que AM sea simétrica

Sea $\mathcal{S} = \{M \in \mathcal{M}_n(\mathbb{R})/(AM)^T = AM\}$. Si la matriz M se factoriza como

$$M = XA^T (X \in \mathcal{M}_n(\mathbb{R})) \tag{3.83}$$

directamente se sigue que $(AM)^T = AM$ si y sólo si $X^T = X$. Así, $\mathcal{S} = \{XA^T/X \in \mathcal{M}_n(\mathbb{R}), X^T = X\}$ y, para aplicar el teorema 33 se puede obtener inmediatamente una base de $\mathcal{S}$ de la base canónica del subespacio de las matrices simétricas de orden n

$$S = span\left(\left\{E_{i,i}A^T\right\}_{i=1}^n \cup \left\{\left(E_{i,j} + E_{j,i}\right)A^T\right\}_{i<j}\right) \tag{3.84}$$

donde $E_{i,j}$ es la matriz cuadrada de orden n cuya única entrada no nula es $e_{ij} = 1$. Note que las matrices de la base anterior de $\mathcal{S}$ sólo contienen una fila de A^T o dos filas en posiciones permutadas.

Además, partiendo de $S' = span\left(\left\{E_{i,i}A^T\right\}_{i=1}^n\right)$ y aumentando la base iterativamente, siempre podemos obtener precondicionadores $M_0 = X_0A^T$ tan cercanos a $A^{-1} \in S$ como se requiera, preservando la simetría de AN tal que $\|AN - I\|_F \leq \left\|AA^T - I\right\|_F$ pues $A^T \in S'$.

El mayor interés de este ejemplo radica en la posbilidad de usar el método CG para resolver las ecuaciones normales generalizadas del sistema (2.1), si X_0 es simétrica definida positiva,

$$AX_0A^ty = b; \quad x = X_0A^Ty \tag{3.85}$$

Capítulo 4

Valores propios y valores singulares en la proyección ortogonal de Frobenius

Los resultados de la teoría de operadores completamente continuos en espacios de Hilbert determinan relaciones particulares para los valores singulares y autovalores de la mejor aproximación en $A\mathcal{S}$, a la identidad. En particular se establece que el menor valor singular y el módulo del menor valor propio de AN no exceden nunca de la unidad cualquiera sea el subespacio $\mathcal{S}$. A continuación se realiza un análisis de convergencia (número de condición, distribución de autovalores y de valores singulares y grado de normalidad) para el precondicionador óptimo en un subespacio arbitrario $\mathcal{S}$ de $\mathcal{M}_n(\mathbb{R})$. Se propone un índice alternativo de la descomposición de Schur para el grado de normalidad

4.1. Menor valor propio y menor valor singular en la proyección ortogonal de Frobenius

Recordemos que la mejor aproximación AN a la identidad en el subespacio $A\mathcal{S}$ se caracteriza por la condición

$$\langle AN - I, AM \rangle_F = 0, \quad \forall M \in \mathcal{S} \tag{4.1}$$

de donde

$$\|AN - I\|_F^2 = n - tr(AN) \tag{4.2}$$

$$\|AN\|_F^2 = tr(AN) \tag{4.3}$$

Denotando por $\{\lambda_k\}_{k=1}^n$ y $\{\sigma_k\}_{k=1}^n$ a los autovalores y los valores singulares de la matriz AN (ordenados en orden decreciente de sus módulos), respectiva-

mente, la ecuación (4.3) se escribe como

$$\sum_{k=1}^{n} \sigma_k^2 = \sum_{k=1}^{n} \lambda_k \tag{4.4}$$

Es decir, para la mejor aproximación AN a la matriz identidad en el sentido de Frobenius, la suma de los cuadrados de los valores singulares coincide con la suma de los autovalores.

Los siguientes resultados establecen con más precisión la relación entre los autovalores y los valores singulares de AN (A y N reales)

Lema 39 *Sea $\mathcal{S}$ un subespacio vectorial cualquiera de $\mathcal{M}_n(\mathbb{R})$ y*

$$\min_{M \in \mathcal{S}} \|AM - I\|_F = \|AN - I\|_F \ ; \ \mathcal{S} \subseteq \mathcal{M}_n(\mathbb{R}). \tag{4.5}$$

Entonces:

$$\sum_{k=1}^{n} \lambda_k^2 \leq \sum_{k=1}^{n} |\lambda_k|^2 \leq \sum_{k=1}^{n} \sigma_k^2 = \sum_{k=1}^{n} \lambda_k \leq \sum_{k=1}^{n} |\lambda_k| \leq \sum_{k=1}^{n} \sigma_k \tag{4.6}$$

Demostración. Aplicando el Teorema de la mayorante de Weyl [106]

$$\sum_{k=1}^{m} |\lambda_k|^p \leq \sum_{k=1}^{m} \sigma_k^p \quad (p > 0 \ ; m = 1, 2, ..., n) \tag{4.7}$$

para $m = n$ y $p = 1$, $p = 2$ y la ecuación (4.4), se concluye la demostración ■

En particular, si AN es simétrica, entonces la igualdad sustituye a todas las desigualdades en (4.6) con la única excepción de $\sum_{k=1}^{n} \lambda_k \leq \sum_{k=1}^{n} |\lambda_k|$. Más aún, si AN simétrica y definida positiva, entonces las seis sumas en (4.6) coinciden.

Teorema 40 *El menor valor singular y el módulo del menor valor propio de AN no son nunca mayores que 1*

Demostración. La ecuación $\|AN - I\|_F^2 = n - tr(AN)$ y el lema 39 nos llevan a,

$$0 \leq \|AN - I\|_F^2 = n - \sum_{k=1}^{n} |\lambda_n|^2 \leq n(1 - |\lambda_n|^2) \leq n(1 - \sigma_n^2) \tag{4.8}$$

de donde,

$0 \leq 1 - |\lambda_n|^2 \Longrightarrow |\lambda_n| \leq 1$

$0 \leq 1 - \sigma_n^2 \Longrightarrow \sigma_n \leq 1$

■

El resultado anterior puede ser utilizado para al interpretar la proximidad a la unidad del módulo del menor valor propio (menor valor singular) en términos de la calidad del precondicionador. De paso, proporcionamos una demostración alternativa para el teorema anterior.

Teorema 41 *$|\lambda_n|$ (o σ_n) determinan por si mismas la calidad del precondicionador N. Con más precisión,*

$$\lim_{|\lambda_n|\to 1} \|AN - I\|_F = \lim_{\sigma_n \to 1} \|AN - I\|_F = 0 \tag{4.9}$$

Demostración. De la ecuación (4.8), se obtiene directamente
$|\lambda_n| \to 1 \Longrightarrow \|AN - I\|_F \to 0$
$\sigma_n \to 1 \Longrightarrow \|AN - I\|_F \to 0$ ■

Una consecuencia inmediata del teorema 40 en relación con el precondicionador N definido por:
$\mathcal{S} = \{M \in \mathcal{M}_n(\mathbb{R})/AM = (AM)^T\}$; $\min_{M\in S} \|AM - I\|_F = \|AN - I\|_F$
nos indica que la generalización de la ecuación normal $A^TAy = b$; $x = A^Ty$ definida en el Capítulo 3 (apartado 3.4.4) constituye una mejora real de dicha ecuación. Con más precisión,

Corolario 42 *(i) Si el menor valor singular de A es mayor que la unidad, entonces, en cualquier subespacio $\mathcal{S}_0$ de $\mathcal{M}_n(\mathbb{R})$ que contenga a A^T, es posible (y sencillo) encontrar una matriz N_0 tal que:*

$$\|AN_0 - I\|_F \lneqq \left\|AA^T - I\right\|_F \tag{4.10}$$

(ii) Si el menor valor singular de A es mayor que la unidad, entonces, en cualquier subespacio $\mathcal{S}_0$ de $\mathcal{S}_2$ que contenga a A^T, es posible (y sencillo) encontrar una matriz N_0 tal que: AN_0 simétrica y

$$\|AN_0 - I\|_F \lneqq \left\|AA^T - I\right\|_F \tag{4.11}$$

Demostración. (i) Sea $\mathcal{S}_0 \subset \mathcal{M}_n(\mathbb{R})$ tal que $\mathcal{S}_0 \ni A^T$ y consideremos el problema:

$$\min_{M\in S_0} \|AM - I\|_F = \|AN_0 - I\|_F \tag{4.12}$$

Procedamos por reducción al absurdo. Si $\nexists N_0 \in S_0$ tal que $\|AN_0 - I\|_F \lneqq \left\|AA^T - I\right\|_F$, entonces, el mínimo (4.10) se alcanza en A^T, es decir, $N_0 = A^T$. En virtud del teorema 40
$|\lambda_n(AN_0)| \leq 1 \Longrightarrow |\lambda_n(AA^T)| \leq 1 \Longrightarrow \sigma_n(A) \leq 1$
en contra de la hipótesis. ■

Con esto hemos probado que en todos aquellos casos en que $\sigma_n(A) > 1$, podemos precondicionar el sistema $Ax = b$ con una matriz $N_0 \neq A^T$ que simetrice a A, es decir, $AN_0 = (AN_0)^T$, y que, a la vez, mejore a la propia A^T como inversa aproximada: $\|AN_0 - I\|_F \lneqq \left\|AA^T - I\right\|_F$.

Finalmente, establecemos una condición necesaria para que la matriz A tenga una inversa aproximada en el subespacio $\mathcal{S}$. Para este propósito se usan los siguientes resultados (ver [64]).

Lema 43 *(Horn) Sean $A, B \in M_n(R)$, y sea $f(x)$ una función no decreciente de variable real en $[0, \infty)$ y convexa mediante la sustitución $x = e^t$ $(-\infty < t < \infty)$. Denotamos los valores singulares ordenados de A, B, y AB por $\sigma_1(A) \geq \cdots \geq \sigma_n(A) \geq 0$, $\sigma_1(B) \geq \cdots \geq \sigma_n(B) \geq 0$, y $\sigma_1(AB) \geq \cdots \geq \sigma_n(AB) \geq 0$, entonces*

$$\sum_{k=1}^{m} f\left(\sigma_k\left(AB\right)\right) \leq \sum_{k=1}^{m} f\left(\sigma_k\left(A\right)\sigma_k\left(B\right)\right), \quad \forall m = 1, 2, ..., n \tag{4.13}$$

Esto nos lleva al siguiente teorema.

Teorema 44 *Sea $A \in M_n(R)$ una matriz no singular y sea $\mathcal{S}$ cualquier subespacio vectorial de $M_n(R)$. Si $\mathcal{S}$ contiene una inversa aproximada de A entonces,*

$$\sum_{k=1}^{n} \left[1 - \sigma_k^2\left(A\right)\sigma_k^2\left(M_0\right)\right] < 1 \tag{4.14}$$

Demostración. Usando el Lema de Horn para $f(x) = x^2$, tenemos

$$\|AN\|_F^2 = \sum_{k=1}^{n} \sigma_k^2\left(AN\right) \leq \sum_{k=1}^{n} \sigma_k^2\left(A\right)\, \sigma_k^2\left(N\right) \tag{4.15}$$

y de las ecuaciones (4.2) y (4.3),

$$1 > \|AM_0 - I\|_F^2 = n - \|AM_0\|_F^2 \geq \sum_{k=1}^{n} \left[1 - \sigma_k^2\left(A\right)\sigma_k^2\left(M_0\right)\right] \tag{4.16}$$

■

4.2. Análisis de convergencia

Cuando M es una inversa aproximada de A, la desigualdad: (ver [8])

$$\kappa_2(AM) \leq \frac{1 + \|AM - I\|_F}{1 - \|AM - I\|_F} \tag{4.17}$$

basada en el lema de Banach, muestra que el número de condición de AM puede hacerse arbitrariamente próximo a 1 con tal que $\|AM - I\|_F$ sea suficientemente pequeña. La proximidad de $\kappa_2(AM)$ a la unidad, así como el grado de normalidad de AM y la concentración de sus autovalores y valores singulares en torno a 1, son condiciones esenciales para la convergencia de la mayoría de los métodos iterativos [60]. A continuación, estudiamos el comportamiento de la matriz AN con respecto a estos cuatro parámetros de convergencia.

La distribución de los valores singulares de AN se resume a continuación.

Teorema 45 *Sea S un subespacio vectorial cualquiera de $M_n(R)$ y*

$$\min_{M \in S} \|AM - I\|_F = \|AN - I\|_F \tag{4.18}$$

Entonces, los valores singulares de AN se encuentran en el intervalo

$$[1 - \|AN - I\|_2 , 1 + \|AN - I\|_2] \tag{4.19}$$

y satisfacen la relación

$$\sum_{k=1}^{n} (1 - \sigma_k)^2 \leq \|AN - I\|_F^2 \tag{4.20}$$

Además, si $\|AN - I\|_2 < 1$ entonces

$$\kappa_2(AN) \leq \frac{1 + \|AN - I\|_2}{1 - \|AN - I\|_2} \tag{4.21}$$

Demostración. Como bien es sabido [90], los valores singulares dependen continuamente de sus argumentos y

$$|\sigma_k - 1| \leq \|AM_0 - I\|_2 \ , \ \ \forall k = 1, 2, ..., n \tag{4.22}$$

Además, del lema 39 y de la ecuación (4.2) obtenemos,

$$\sum_{k=1}^{n} (1 - \sigma_k)^2 = \sum_{k=1}^{n} (1 + \lambda_k - 2\sigma_k) \leq \sum_{k=1}^{n} (1 - \lambda_k) = \|AN - I\|_F^2 \tag{4.23}$$

■

Por otra parte, la distribución de los autovalores de AN se presenta en el siguiente teorema, en el que también se propone y evalúa un estimador del grado de normalidad de AN.

Teorema 46 *Sea S un subespacio vectorial cualquiera de $M_n(\mathbb{R})$ y*

$$\min_{M \in S} \|AM - I\|_F = \|AN - I\|_F \tag{4.24}$$

Entonces los autovalores de AN se encuentran dentro de un círculo de radio $\|AN - I\|_F$ y centro 1 y satisfacen

$$\sum_{k=1}^{n} |1 - \lambda_k|^2 \leq \|AM_0 - I\|_F^2 \tag{4.25}$$

Además,

$$\frac{1}{n} \sum_{k=1}^{n} (|\lambda_k| - \sigma_k)^2 \leq \frac{2}{n} \|AM_0\|_F^2 \ (1 - \sigma_n) \tag{4.26}$$

Demostración. De la inecuación de Weyl, el lema 39 y la ecuación (4.2) se sigue

$$\sum_{k=1}^{n} |1-\lambda_k|^2 = \sum_{k=1}^{n} \left(1+|\lambda_k|^2-2\lambda_k\right) \leq \sum_{k=1}^{n} (1-\lambda_k) = \|AM_0 - I\|_F^2$$

estimación obtenida por Grote y otros [60] usando una descomposición de Schur de la matriz $AN - I$.

Por otro lado, de la inecuación de Weyl y las ecuaciones (4.3) y (4.4), obtenemos

$$\sum_{k=1}^{n} \left(|\lambda_k| - \sigma_k\right)^2 = \sum_{k=1}^{n} \left(|\lambda_k|^2 + \lambda_k - 2|\lambda_k|\,\sigma_k\right) \leq 2\sum_{k=1}^{n} \left(\lambda_k - |\lambda_k|\,\sigma_k\right)$$

$$\leq 2\left(\sum_{k=1}^{n} \lambda_k\right)(1-\sigma_n) = 2\|AM_0\|_F^2\,(1-\sigma_n) \quad \blacksquare$$

La expresión $\frac{1}{n}\sum_{k=1}^{n} \left(|\lambda_k| - \sigma_k\right)^2$ se propone aquí como un estimador del grado de normalidad de AN basándonos en la conocida caracterización de los operadores lineales normales en función de sus valores propios y singulares [55].

Capítulo 5

Inversa aproximada *sparse*

Actualmente, los métodos basados en subespacios de Krylov son las herramientas más eficientes para resolver sistemas de ecuaciones lineales (2.1). Sin embargo, estos métodos deben usarse con precondicionadores muy a menudo. Recientemente, el uso de inversas aproximadas se ha convertido en una buena alternativa para los precondicionadores implícitos debido a su naturaleza paralelizable; para más detalles ver, [20, 38, 41, 61], y también un estudio comparativo completo en [24]. En este Capítulo, se construye una matriz inversa aproximada usando el producto escalar de Frobenius (sección 5.1). Aunque el estudio se ha desarrollado para inversas aproximadas precondicionando por la derecha, los resultados por la izquierda son directos. De hecho, la mayoría de los experimentos en este capítulo se han ejecutado precondicionando por la izquierda. En la sección 5.2, se propone un Algoritmo de la Inversa Aproximada mejorada (IAI) no sólo para obtener un mejor precondicionador a partir de una inversa aproximada dada y usarlo combinado con un método iterativo basado en los subespacios de Krylov, sino incluso para resolver (2.1). El apartado 5.2.1 es un resumen de algunas propiedades teóricas de la inversa aproximada *sparse* y la mejorada. Finalmente, en la sección 5.3 se ilustra la eficiencia de estos precondicionadores y del algoritmo IAI con algunos experimentos numéricos.

5.1. Cálculo de inversas aproximadas *sparse*

Sea $\mathcal{S} \subset \mathcal{M}_n$, el subespacio de las matrices M donde se busca una inversa aproximada explícita con un patrón de *sparsidad* desconocido. La formulación del problema es: encontrar $M_0 \in \mathcal{S}$ tal que

$$M_0 = \arg\min_{M \in \mathcal{S}} \|AM - I\|_F \tag{5.1}$$

Además, esta matriz inicial M_0 pudiera ser posiblemente una aproximada inversa de A en un sentido estricto, es decir, según la definición dada por [71],

$$\|AM_0 - I\|_F = \varepsilon < 1 \tag{5.2}$$

Existen dos razones para esto. Primera, la ecuación (5.2) permite asegurar que M_0 es no singular (lema de Banach), y segunda, esta será la base para construir un algoritmo explícito para mejorar M_0 y resolver la ecuación (2.1).

Un trabajo reciente de Grote y otros [61] nos da un algoritmo eficiente para obtener una inversa aproximada tan cercana a una matriz no singular A como se requiera. Hemos seguido dicha técnica pero variando el método de selección de las entradas en M_0 y el algoritmo usado para resolver el problema (5.1). La construcción de M_0 se realiza en paralelo, independizando el cálculo de cada columna. Aunque nuestro algoritmo nos permite comenzar desde cualquier entrada de la columna k, se acepta comúnmente el uso de la diagonal como primera aproximación. Además, la expresión del precondicionador diagonal óptimo es bien conocida. La siguiente entrada a considerar se selecciona dentro del conjunto de entradas candidatas, el cual se define siguiendo el criterio propuesto por Grote y otros [61]. Sea r_k el residuo correspondiente a la columna k-ésima,

$$r_k = Am_k - e_k \tag{5.3}$$

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

$$||Am_k - e_k||_2^2 = 1 - \sum_{l=1}^{p_k} \frac{\left[\det\left(D_l^k\right)\right]^2}{\det\left(G_{l-1}^k\right) \det\left(G_l^k\right)} \tag{5.4}$$

donde, para todo k, $\det\left(G_0^k\right) = 1$ y G_l^k es la matriz de Gram de las columnas $i_1^k, i_2^k, ..., i_l^k$ de la matriz A con respecto al producto escalar euclídeo, D_l^k es la matriz que resulta de reemplazar la última fila de la matriz G_l^k por $a_{k\,i_1^k}, a_{k\,i_2^k}, ..., a_{k\,i_l^k}$, con $1 \leq l \leq p_k$. Se selecciona el índice j_k que minimiza el valor de $\|Am_k - e_k\|_2$. Esta estrategia (ver [56]) define el nuevo índice seleccionado j_k atendiendo solamente al conjunto $\mathcal{L}_k$, lo que nos lleva a un nuevo óptimo donde se actualizan todas las entradas correspondientes a los índices de $\mathcal{L}_k$. Esto mejora el criterio de [61] donde el nuevo índice se selecciona manteniendo las entradas correspondientes a los índices de $\mathcal{L}_k$. Así m_k se busca en el conjunto

$$\mathcal{S}_k = \{m_k \in \mathbb{R}^n / m_{ik} = 0; \forall i \notin \mathcal{L}_k \cup \{j_k\}\}$$

y

$$m_k = \sum_{l=1}^{p_k} \frac{\det\left(D_l^k\right)}{\det\left(G_{l-1}^k\right) \det\left(G_l^k\right)} \widetilde{m}_l \tag{5.5}$$

donde $\widetilde{m}_l$ es el vector con entradas no nulas i_h^k $(1 \leq h \leq l)$. Cada una de ellas se obtiene evaluando el determinante correspondiente que resulta de reemplazar la última fila de $\det\left(G_l^k\right)$ por e_h^t, con $1 \leq l \leq p_k$.

Evidentemente, los cálculos de $||Am_k - e_k||_2^2$ y m_k pueden actualizarse añadiendo la contribución de la última entrada $j \in \mathcal{J}_k$ a la suma previa de 1 a $p_k - 1$. En la práctica, $\det\left(G_l^k\right)$ se calcula usando la descomposición de Cholesky puesto que G_l^k es una matriz simétrica y definida positiva. Esto sólo involucra la factorización de la última fila y columna si aprovechamos la descomposición de G_{l-1}^k. Por otra parte, $\det\left(D_l^k\right) / \det\left(G_l^k\right)$ es el valor de la última incógnita del sistema $G_l^k\, d_l = \left(a_{k\, i_1^k}, a_{k\, i_2^k}, ..., a_{k\, i_l^k}\right)^t$ necesitándose solamente una sustitución por descenso . Finalmente, para obtener $\widetilde{m}_l$ debe resolverse el sistema $G_l^k\, v_l = e_l$, con $\widetilde{m}_{i_h^k l} = v_{hl}$, $(1 \leq h \leq l)$.

5.2. Método de la Inversa Aproximada mejorada

Aquí se muestra cómo calcular la solución aproximada de un sistema lineal y *sparse* para una inversa aproximada dada. El algoritmo se basa en el siguiente teorema.

Teorema 47 *Si M es una inversa aproximada de A en el sentido estricto, entonces $2M - MAM$ es una inversa aproximada de A mejor.*

Demostración. Sea M una matriz tal que $||AM - I|| = \varepsilon < 1$, para $2M - MAM$ tenemos,

$||A\,[2M - MAM] - I|| = ||[AM - I]\,[I - AM]|| \leq ||AM - I||^2 = \varepsilon^2 < \varepsilon < 1$

Esto se cumple tanto para la norma 2 como para la norma Frobenius. Puede demostrarse lo mismo precondicionando por la izquierda ■

Teniendo en cuenta este resultado, se propone un algoritmo simple pero eficiente para mejorar la inversa aproximada así como resolver la ecuación (2.1). Sea R_i la matriz residuo,

$$R_i = AM_i - I \tag{5.6}$$

Se supone que comenzamos por una inversa aproximada inicial M_0 y la condición (5.2) se cumple. Entonces, usando el teorema 47 sucesivamente, la i-ésima aproximación puede escribirse,

$$M_i = 2M_{i-1} - M_{i-1}AM_{i-1} = M_{i-1}\,(I - R_{i-1}) \tag{5.7}$$

y la correspondiente matriz residuo,

$$R_i = -R_{i-1}^2 \tag{5.8}$$

En este paso, la actualización de la correspondiente solución aproximada de la ecuación (2.1) es,

$$x_i = M_i b \tag{5.9}$$

y el vector residuo,

$$r_i = Ax_i - b = R_i b \tag{5.10}$$

Por lo tanto si calculamos inicialmente,

$$x_0 = M_0 b, \qquad r_0 = R_0 b, \qquad \text{con} \quad R_0 = AM_0 - I,$$

la matriz M_i puede ser actualizada por,

$$M_i = M_0 (I - R_0) \prod_{j=1}^{i-1} \left(I + R_0^{2^j} \right) \tag{5.11}$$

y la matriz residuo,

$$R_i = -R_0^{2^i} \tag{5.12}$$

Entonces podemos afirmar que M_i está, en cierto sentido, al menos 2^i veces más cerca a la inversa de A que M_0:

$$\frac{\|r_i\|_2}{\|b\|_2} = \frac{\left\| -R_0^{2^i} b \right\|_2}{\|b\|_2} \leq \left\| -R_0^{2^i} \right\|_2 \leq \varepsilon^{2^i} \tag{5.13}$$

pues $\left\| -R_0^{2^i} \right\|_2 \leq \left\| -R_0^{2^i} \right\|_F \leq \|R_0\|_F^{2^i}$. Ahora, si usamos $\left\| -R_0^{2^i} \right\|_2$ como criterio de parada relativo, el número de pasos i, que son necesarios para satisfacer una tolerancia dada δ, depende de la calidad de la inversa aproximada inicial pero no del orden de la matriz, y puede conocerse a priori

$$i \geq \frac{\log \frac{\log \delta}{\log \varepsilon}}{\log 2} \tag{5.14}$$

Sin embargo, esta estimación es muy pesimista en la práctica y usar $\frac{\left\| -R_0^{2^i} b \right\|_2}{\|b\|_2}$ en vez de $\left\| -R_0^{2^i} \right\|_2$ nos da una estimación más realista. El valor del cociente de las normas vectoriales en (5.13) puede calcularse eficientemente manteniendo el vector $R_0^{2^{i-1}} b$, que se obtiene a lo largo de los cálculos de x_i en (5.9).

Evidentemente, M_i no se construye explícitamente puesto que no se efectúa el producto matriz-matriz de las ecuaciones (5.11). Sólo se ejecutan producto matriz-vector y suma de vectores en la ecuación (5.9) para obtener x_i. El número de productos matriz-vector m para i pasos es

$$m = 3 + 2 \sum_{j=1}^{i-1} 2^j = 2^{i+1} - 1 \geq 2 \frac{\log \delta}{\log \varepsilon} - 1 \tag{5.15}$$

Finalmente, el algoritmo IAI puede usarse como precondicionador en cualquier método iterativo basado en subespacios de Krylov, sustituyendo la ecuación (5.11) en los productos matriz-vector donde intervenga el precondicionador.

5.2.1. Efectividad teórica de la inversa aproximada mejorada

Aquí se estudia el efecto del algoritmo IAI en los parámetros anteriores para la convergencia de los métodos iterativos. En el próximo teorema se demuestra que a cada paso el algoritmo IAI mejora estas cuatro propiedades.

Teorema 48 *Sea M_0 la matriz que minimiza $\|AM - I\|_F$ en el subespacio S y suponemos que $\|AM_0 - I\|_F = \varepsilon < 1$. Sea también M_i la i-ésima aproximación obtenida por el algoritmo IAI. Entonces,*
(i) Los valores propios λ_k^i de AM_i se encuentran en un círculo de centro 1 y radio ε^{2^i} y satisfacen

$$\sum_{k=1}^{n} \left|1 - \lambda_k^i\right|^2 \leq \varepsilon^{2^{i+1}}$$

(ii) La desviación de la normalidad de AM_0 se reduce por el algoritmo IAI . (iii) Los valores singulares σ_k^i de AM_i están agrupados alrededor de 1, dentro del intervalo $\left[1 - \delta^{2^i}, 1 + \delta^{2^i}\right]$, con $\delta = \|AM_0 - I\|_2$, y satisfacen

$$\sum_{k=1}^{n} \left(1 - \sigma_k^i\right)^2 \leq \varepsilon^{2^{i+1}} \left(2 + \delta^{2^i}\right)^2$$

(iv) $\kappa_2(AM_i) \leq \frac{1+\delta^{2^i}}{1-\delta^{2^i}}$

Demostración. Del teorema 47 tenemos

$$\|AM_i - I\| \leq \|AM_0 - I\|^{2^i} \tag{5.16}$$

el cual se cumple tanto para la norma 2 como para la norma Frobenius. Entonces, usando la descomposición de Schur QRQ^t de $AM_i - I$, obtenemos [61]

$$\left|1 - \lambda_k^i\right|^2 \leq \sum_{k=1}^{n} \left|1 - \lambda_k^i\right|^2 \leq \|R_i\|_F^2 = \|AM_i - I\|_F^2 \leq \varepsilon^{2^{i+1}}$$

lo cual prueba (i). Además, como

$$QAM_iQ^t = diag\left(\lambda_1^i, \lambda_2^i, ..., \lambda_n^i\right) + U_i$$

$$\|U_i\|_F^2 \leq \|AM_i - I\|_F^2 \leq \varepsilon^{2^{i+1}}$$

se prueba el apartado (ii). Para demostrar (iii), Se usa la propiedad de que $\sigma_k(A)$ depende de A continuamente, tal y como se hizo en el teorema 43 del Capítulo 4,

$$\left|\sigma_k^i - 1\right| = \left|\sigma_k(AM_i) - \sigma_k(I)\right| \leq \|AM_i - I\|_2 \leq \delta^{2^i}$$

Además, como $1-(\sigma_k^i)^2$ $(k=1,2,...,n)$ son los autovalores de la matriz $I-(AM_i)(AM_i)^t$, procediendo como en [61] obtenemos

$$\sum_{k=1}^{n}\left[1-\sigma_k^i\right]^2 \leq \sum_{k=1}^{n}\left[1-\left(\sigma_k^i\right)^2\right]^2 = \left\|I-(AM_i)(AM_i)^t\right\|_F^2$$

$$= \left\|I+(I-AM_i)M_i^tA^t-M_i^tA^t\right\|_F^2 \leq \|I-AM_i\|_F^2\ (1+\|AM_i\|_2)^2$$

$$\leq \|I-AM_i\|_F^2\ (2+\|AM_i-I\|_2)^2 \leq \varepsilon^{2^{i+1}}\left(2+\delta^{2^i}\right)^2$$

Finalmente, el apartado (iv) se obtiene directamente del (iii). ■

5.3. Experimentos numéricos

En esta sección consideramos algunos problemas obtenidos de aplicaciones científicas e industriales. Mostraremos la efectividad tanto de los precondicionadores *sparse* como IAI con tres de los métodos iterativos más usados hoy en día basados en los subespacios de Krylov para sistemas lineales no simétricos: Bi-CGSTAB [104], QMRCGSTAB [31] y una variante del GMRES [92], [51], con precondicionamiento [97]. Todos los cálculos númericos se realizaron en FORTRAN con doble precisión. Se tomó siempre como aproximación inicial $x_0=0$, $\tilde{r}_0=r_0=b$, y como criterio de parada $\frac{\|b-Ax_i\|_2}{\|b\|_2}<10^{-9}$. A continuación mostramos una breve descripción de los problemas:

convdifhor: matriz obtenida de un problema convección-difusión en dos dimensiones resuelto mediante elementos finitos con una malla adaptativa refinada, de tamaño $n=441$ y entradas no nulas $nz=2927$.

oilgen: matriz de simulación de una reserva de petróleo para una malla completa $21\times21\times5$, de tamaño $n=2205$ y $nz=14133$.

sherman: simulador del almacenamiento de petróleo en medio con paredes de lajas, con malla $10\times10\times10$, de tamaño $n=1000$ y $nz=3750$.

pores: matriz no simétrica de tamaño $n=30$ y $nz=180$.

isla: matriz de tamaño $n=12666$ y $nz=86562$, obtenida en un problema de convección-difusión en un dominio bidimensional definido en la costa de la isla de Gran Canaria, resuelto mediante el método de elementos finitos, usando una malla adaptativa.

La tabla 5.1 muestra cómo se mejora la convergencia del BiCGSTAB cuando se usa el precondicionador *sparse* para *convdifhor*. El número de iteraciones disminuye claramente con ε_k. Sin embargo, para obtener una inversa aproximada

de A, las entradas por colmunas deben aumentarse a 200. En las figuras 5.1, 5.2, 5.3 y 5.4 se muestran los gráficos de la estructura *sparse* de A y M, respectivamente, para este problema. Podemos observar que los patrones de sparsidad de A y A^{-1} son totalmente diferentes y no podemos, por tanto, definir la estructura de M igual a la de A, ni en general a la de una matriz fija.

$\max nz(m_{0k})$	ε_k	*Iter.*	$nz(M_0)$	$\frac{nz(M_0)}{nz(A)}$	$\|M_0A - I\|_F$
50	0,5	86	949	0,32	9,38
50	0,4	68	1905	0,65	8,00
50	0,3	34	3646	1,24	5,99
50	0,2	22	7736	2,64	4,06
50	0,05	11	20106	6,86	2,01
200	0,05	7	43390	14,82	0,99

Tabla 5.1: *Resultados de convergencia para convdifhor con BiCGSTAB precondicionado por la izquierda*

Por otra parte, en la figura 5.5 se muestra que las inversas aproximadas tipo *sparse* pueden mejorar el comportamiento de algunos precondicionadores clásicos como el ILU(0) y ser competitivas cuando se realizan los cálculos en paralelos.

Las tablas 5.2, 5.3 y 5.4 muestran que estos precondicionadores son también efectivos con diferentes métodos iterativos basados en subespacios de Krylov como el GMRES y el QMRCGSTAB. En la tabla 5.5 se comparan los algoritmos IAI con inversas aproximadas por la izquierda y por la derecha con algunos métodos iterativos basados en subespacios de Krylov. Ambos ejemplos muestran que IAI puede ser competitiva si es posible encontrar una inversa aproximada. Desafortunadamente, la construcción de la inversa aproximada es muy costosa para matrices grandes.

Se ha estudiado el comportamiento de estos precondicionadores tipo *sparse* en problemas de orden más elevado como *isla*. Los resultados se muestran en la tabla 5.4. El número de iteraciones disminuye si mejoramos la calidad del precondicionador. A pesar de la convergencia irregular del algoritmo BiCGSTAB en este caso (ver figura 5.6), el precondicionador *sparse* puede mejorar y aún obtener resultados mejores que ILU(0).

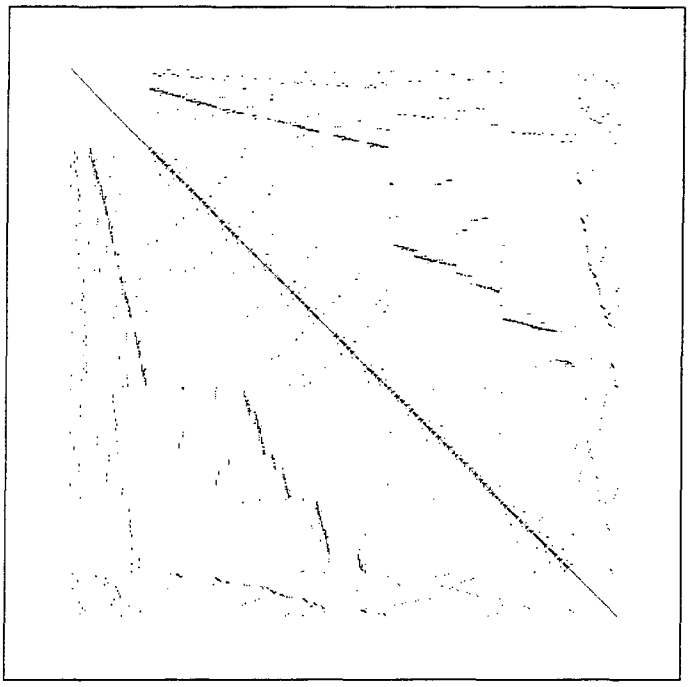

Figura 5.1: *Estructura sparse de la matriz convdifhor.*

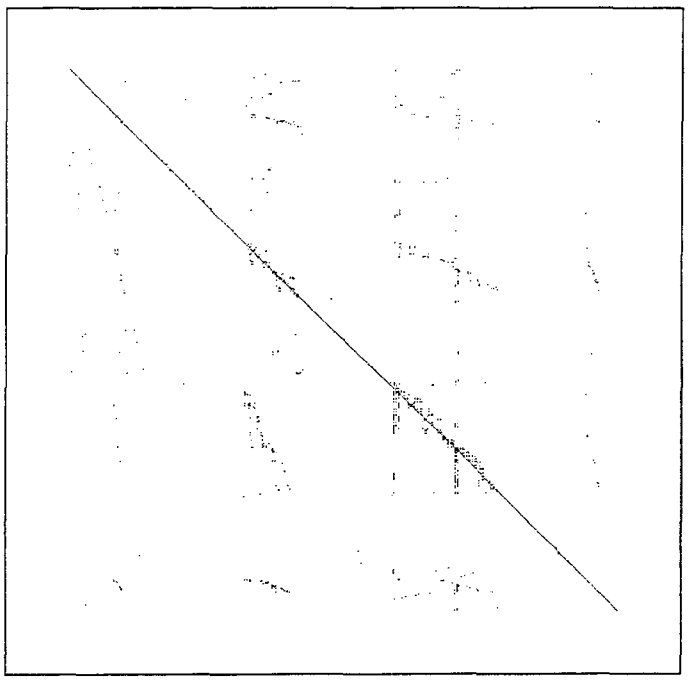

Figura 5.2: *Estructura sparse de la inversa aproximada de la matriz convdifhor con* $\varepsilon_k = 0{,}5$ *y* máx $nz(m_{0k}) = 50$.

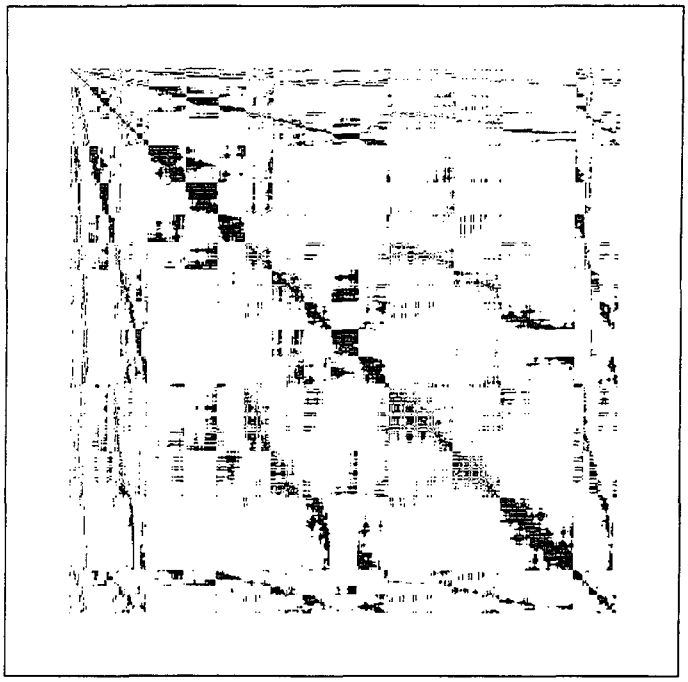

Figura 5.3: *Estructura sparse de la inversa aproximada de la matriz convdifhor con* $\varepsilon_k = 0{,}05$ *y* máx $nz(m_{0k}) = 50$.

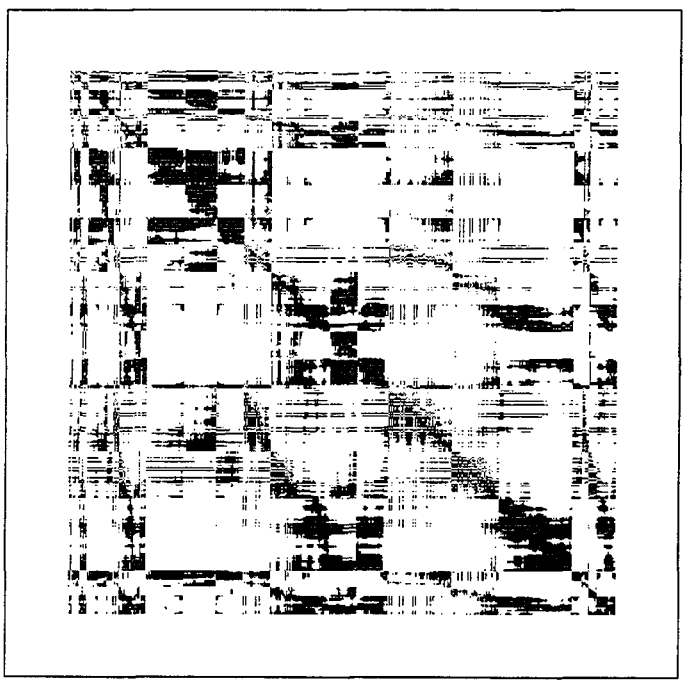

Figura 5.4: *Estructura sparse de la inversa aproximada de la matriz convdifhor con* $\varepsilon_k = 0{,}05$ *y* máx $nz(m_{0k}) = 200$.

ε_k	k	*Iter.*	$nz(M_0)$	$\frac{nz(M_0)}{nz(A)}$	$\|M_0A - I\|_F$
0,6	100	31	3087	0,21	26,76
0,4	100	2	10353	0,73	13,66
0,3	85	1	11025	0,78	11,44
0,2	58	1	27964	1,98	8,89

Tabla 5.2: *Resultados de convergencia para oilgen con GMRES precondicionado por la izquierda.*

	Iter.	$nz(M_0)$	$\frac{nz(M_0)}{nz(A)}$	$\|M_0A - I\|_F$
$\varepsilon_k = 0{,}6$	122	1297	0,35	13,33
$\varepsilon_k = 0{,}5$	73	1880	0,50	11,53
$\varepsilon_k = 0{,}4$	50	2709	0,72	9,39
$\varepsilon_k = 0{,}3$	37	4294	1,14	7,38
$\varepsilon_k = 0{,}2$	25	10072	2,69	5,03
$M_0 = I$	408	1000	0,26	68,01
$M_0^{-1} = ILU(0)$	38	3750	1,00	

Tabla 5.3: *Resultados de convergencia para sherman con QMRCGSTAB precondicionado por la izquierda.*

	Iter.	$nz(M_0)$	$\frac{nz(M_0)}{nz(A)}$	$\|M_0A - I\|_F$
$\varepsilon_k = 0{,}5$	795	19227	0,22	51,36
$\varepsilon_k = 0{,}4$	656	39956	0,46	41,54
$\varepsilon_k = 0{,}3$	270	78003	0,90	31,64
$\varepsilon_k = 0{,}2$	142	209186	2,42	21,64
$M_0^{-1} = ILU(0)$	218	86562	1,00	

Tabla 5.4: *Resultados de convergencia para isla con BiCGSTAB precondicionado por la izquierda.*

	pores *Iter.*	*convdifhor* *Iter.*
BiCGSTAB	5	7
QMRCGSTAB	5	7
GMRES(6)	2	2
Left-IAI	3	4
Right-IAI	5	6

Tabla 5.5: *Comparación de los resultados de convergencia para pores con métodos de Krylov e IAI.*

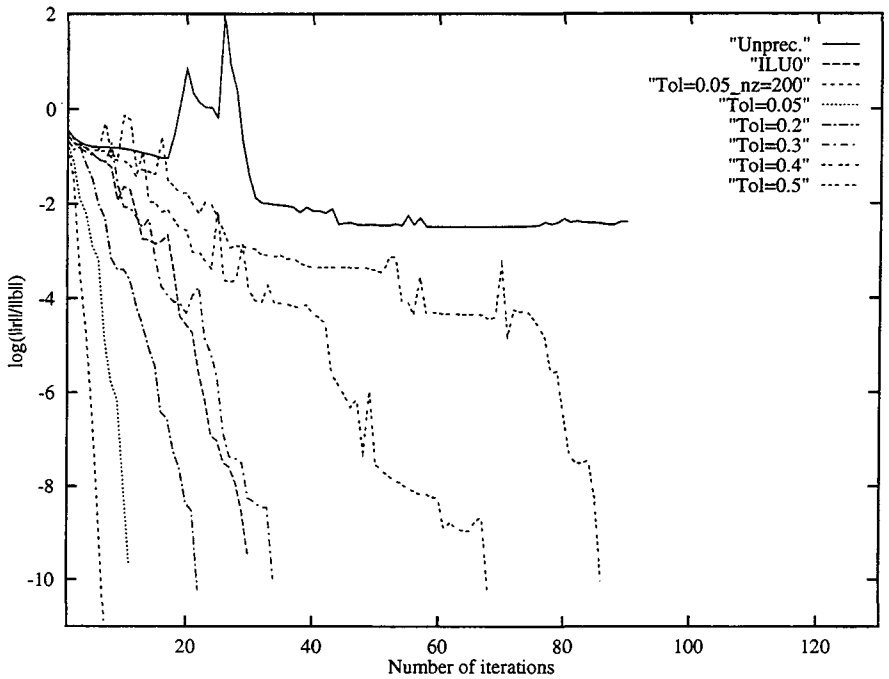

Figura 5.5: *Comportamiento de los prencondicionadores con BiCGSTAB para convdif-hor.*

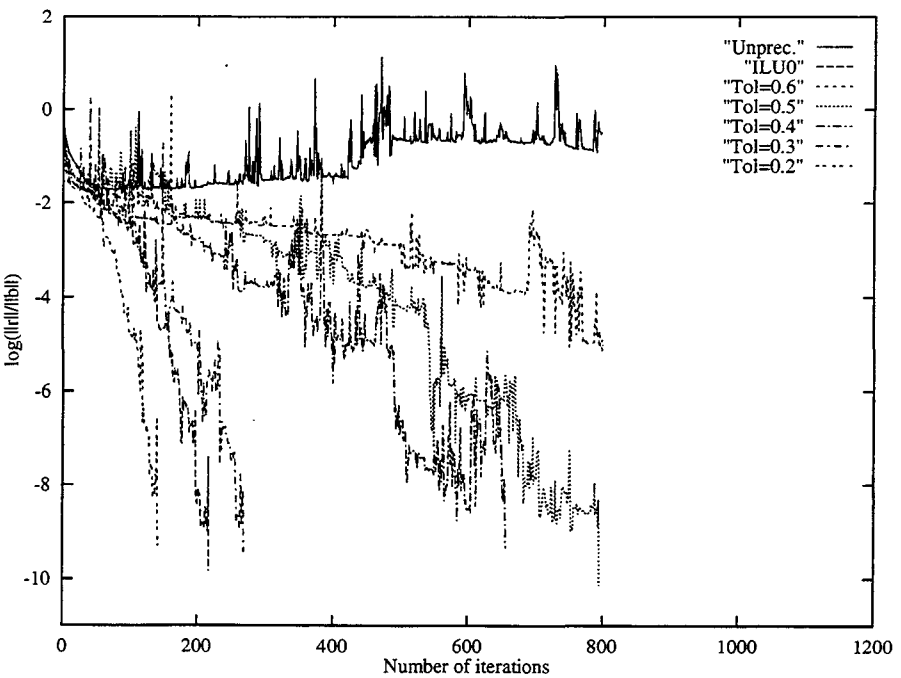

Figura 5.6: *Comportamiento de los prencondicionadores con BiCGSTAB para isla.*

Capítulo 6

Efecto de la reordenación en precondicionadores del tipo inversa aproximada *sparse*

El objetivo de este Capítulo es estudiar el efecto de la reordenación sobre nuestra versión del precondicionador SPAI (inversa aproximada *sparse*) propuesta por Grote y otros [61], cuyos aspectos tanto teóricos como computacionales se han analizado en [56, 84] . Presentamos los resultados sobre el efecto de la reordenación no sólo en la cantidad de entradas en los factores de las inversas, sino también en el número de pasos del método iterativo (ver [49]). Aunque la inversa A^{-1} está normalmente llena, independientemente de la reordenación seleccionada, mostramos experimentalmente cómo el efecto *fill-in* de las inversas aproximadas *sparse* depende de la ordenaciíon de A. Benzi y otros [25] han realizado un estudio similar para inversas aproximadas factorizadas. También es un punto de partida interesante, el artículo de Benzi, Szyld and van Duin [21] sobre el efecto de la reordenación para la factorización incompleta en la convergencia de los métodos basados en subespacios de Krylovs no simétricos.

En primer lugar, se resume el algoritmo de la inversa aproximada *sparse* en la sección 6.1. En la sección 6.2, se discuten algunas consideraciones sobre las técnicas de reordenación tales como Grado Mínimo, Mínimo Vecino, Reverse Cuthill-Mckee, [1], [54]. Finalmente, se resuelven algunos experimentos numéricos para mostrar el efecto de los algoritmos de reordenación sobre la convergencia del método de Bi-CGSTAB [104] para la solución de sistemas de ecuaciones lineales no simétricos donde se utilizan dichas inversas aproximadas como precondicionadores. Se presentan los resultados de los sistemas cuyas matrices peretenecen a la colección Harwell-Boeing [43] y otros que provienen de la discretización por el Método de elementos finitos de diferentes problemas de contorno.

6.1. Algoritmo de la inversa aproximada *sparse*

Sean $r_k^t = m_k^t A - e_k^t$ el residuo correspondiente a la fila k de M e $\mathcal{I}_k$ el conjunto de índices de entradas no nulas en r_k^t, es decir, $\mathcal{I}_k = \{i \in \{1, 2, ..., n\} \ / \ r_{ki} \neq 0\}$. Si $\mathcal{L}_k = \{l \in \{1, 2, ..., n\} \ / \ m_{lk} \neq 0\}$, entonces la nueva entrada se busca en el conjunto $\mathcal{J}_k = \{j \in \mathcal{L}_k^c \ / \ a_{ji} \neq 0, \forall i \in \mathcal{I}_k\}$. Así, las únicas entradas consideradas en m_k^t son aquellas que afectan las entradas no nulas en r_k^t. Asumimos que $\mathcal{L}_k \cup \{j\} = \{i_1^k, i_2^k, ..., i_{p_k}^k\}$ es no vacío, con p_k el número actual de entradas no nulas de m_k^t y $i_{p_k}^k = j$, para todo $j \in \mathcal{J}_k$. Para cada j, calculamos

$$||m_k^t A - e_k^t||_2^2 = 1 - \sum_{l=1}^{p_k} \frac{\left[\det\left(D_l^k\right)\right]^2}{\det\left(G_{l-1}^k\right) \det\left(G_l^k\right)} \tag{6.1}$$

donde, para todo k, $\det\left(G_0^k\right) = 1$ y G_l^k es la matriz de Gram de las filas $i_1^k, i_2^k, ..., i_l^k$ de la matriz A con respecto al producto escalar euclídeo, D_l^k resulta de reemplazar la última fila de la matriz G_l^k por $a_{i_1^k k}, a_{i_2^k k}, ..., a_{i_l^k k}$, con $1 \leq l \leq p_k$. Se selecciona el índice j_k que minimiza el valor de $||m_k^t A - e_k^t||_2$. Esta estrategia define el nuevo índice seleccionado j_k atendiendo solamente al conjunto $\mathcal{L}_k$, lo que nos lleva a un óptimo donde todas las entradas correspondientes a los índices de $\mathcal{L}_k$ se actualizan. Así m_k^t se busca en el conjunto $\mathcal{S}_k = \{m_k^t \in \mathbb{R}^n / m_{ik} = 0; \forall i \notin \mathcal{L}_k \cup \{j_k\}\}$, y

$$m_k = \sum_{l=1}^{p_k} \frac{\det\left(D_l^k\right)}{\det\left(G_{l-1}^k\right) \det\left(G_l^k\right)} \widetilde{m}_l^t \tag{6.2}$$

donde $\widetilde{m}_l^t$ es el vector de entradas no nulas i_h^k $(1 \leq h \leq l)$. Cada una de ellas se obtiene evaluando el determinante correspondiente que resulta de reemplazar la última fila en $\det\left(G_l^k\right)$ por e_h^t, con $1 \leq l \leq p_k$.

6.2. Algunos comentarios sobre reordenación

Hemos considerado diferentes técnicas de reordenación para mostrar el efecto de la reordenación sobre la solución iterativa de sistemas de ecuaciones lineales usando los precondicionadores SPAI. La ordenación original corresponde a las matrices provenientes de la aplicación del Método de Elementos Finitos con mallas no estructuradas y refinamiento adaptativo de mallas. Los algoritmos de reordenación han sido expuestos en el capítulo 2, sección 2.5.

Ahora nuestro principal objetivo es investigar si la reordenación reduce la cantidad de entradas en el precondicionador SPAI y si el uso de los precondicionadores SPAI mejora la convergencia del método iterativo.

Sea P la matriz de permutación asociada a un algoritmo de reordenación. Como $(P^T A P)^{-1} = P^T A^{-1} P$, es decir, la inversa de la matriz reordenada es la

reordenada de la matriz inversa, cuando reordenamos una matriz, su inversa aproximada debe tender a la reordenada de la inversa.

Si la tolerancia de la inversa aproximada está dada por ε, en el subespacio $\mathcal{S} \subset M_n(\mathbb{R})$,

$$\min_{M \in \mathcal{S}} \|MA - I\|_F = \|NA - I\|_F < \varepsilon \tag{6.3}$$

entonces,

$$\min_{M' \in P^T SP} \left\|M' P^T A P - I\right\|_F = \min_{M' \in P^T SP} \left\|P M' P^T A - I\right\|_F = \min_{M \in \mathcal{S}} \|MA - I\| < \varepsilon \tag{6.4}$$

Sea $\mathcal{S}'$ un subespacio de $M_n(\mathbb{R})$ correspondiente al mismo número de entradas no nulas que $\mathcal{S}$, donde la inversa aproximada óptima es obtienida para dicho número de entradas no nulas. Cabe destacar, también, que el número de entradas no nulas en $\mathcal{S}$ es el mismo que el de $P^T SP$. En este caso obtenemos,

$$\left\|N' P^T A P - I\right\|_F = \min_{M' \in \mathcal{S}'} \left\|M' P^T A P - I\right\|_F \leq \min_{M' \in P^T SP} \left\|M' P^T A P - I\right\|_F < \varepsilon \tag{6.5}$$

Evidentemente, el número de entradas no nulas necesarias en S' será menor o igual que el número de entradas no nulas en $P^T SP$ y en S. Concluimos que la reordenación reduce la cantidad de entradas no nulas en la inversa aproximada para una tolerancia dada ε, o, al menos, no lo aumenta.

Debido a los resultados dados en (6.5), los precondicionadores tipo inversa aproximada reordenados adquieren mejores propiedades desde el punto de vista de su utilización para mejorar la convergencia de los métodos iterativos. La cercanía del número de condición de $M' P^T A P$ a 1 se caracteriza por,

$$K_2(M' P^T A P) \leq \frac{1 + \left\|M' P^T A P - I\right\|_2}{1 - \left\|M' P^T A P - I\right\|_2} \tag{6.6}$$

La desviación de la normalidad de $(M' P^T A P)$ está acotada por,

$$\frac{1}{n} \sum_{k=1}^{n} (|\lambda_k| - \sigma_k)^2 \leq \frac{2}{n} \left\|M' P^T A P\right\|_F^2 (1 - \sigma_n) \tag{6.7}$$

siendo $\{\lambda_k\}_{k=1}^n, \{\sigma_k\}_{k=1}^n$ los autovalores y valores singulares de $P^T A P M'$ (en secuencia de módulos decreciente). Finalmente, la agrupación de los autovalores y valores singulares estará dada por,

$$\sum_{k=1}^{n} (1 - \sigma_k)^2 \leq \left\|M' P^T A P - I\right\|_F^2 \tag{6.8}$$

$$\sum_{k=1}^{n} (1 - \lambda_k)^2 \leq \left\|M' P^T A P - I\right\|_F^2 \tag{6.9}$$

6.3. Experimentos numéricos

Comenzamos con el estudio de un problema cuya matriz pertenece a la colección Harwell-Boeing: *orsreg1*. Ésta es una matriz de simulación de un almacenamiento de petróleo para una malla de $21 \times 21 \times 5$ de tamaño $n = 2205$ y entradas no nulas $nz = 14133$.

Las tablas 6.1, 6.2 y 6.3 muestran los resultados obtenidos con el algoritmo BiCGSTAB precondicionado para la ordenación Original, Grado Mínimo y Cuthill-Mckee Inverso de la matriz A, respectivamente. El comportamiento de ILU(0) es comparado con varios precondicionadores SPAI correspondientes a diferentes niveles de llenado. Presentamos el número de iteraciones, el número de entradas de M y la norma de Frobenius de la matriz residuo. Los valores de nz para los precondicionadores SPAI en los casos reordenados llegan a ser menores que el caso de la ordenación original a partir de $\varepsilon_k = 0{,}2$. Sin embargo, en los otros casos al menos la norma de la matriz residuo se reduce con la reordenación, y por tanto, no contradicen los resultados teóricos anteriores. Por otro lado, el número de iteraciones del BiCGSTAB siempre disminuyó cuando usamos reordenación, excepto para el primer SPAI, que dió inestabilidad a la convergencia del algoritmo. También observamos en estos experimentos que, con requerimientos de almacenamiento similares a ILU(0), se produce una convergencia más rápida con SPAI $\varepsilon_k = 0{,}3$, o incluso $\varepsilon_k = 0{,}2$ si se reordena con Cuthill-Mckee Inverso). La figura 6.1 representa el comportamiento del algoritmo BiCGSTAB precondicionado cuando ILU(0) y SPAI(0.3) se construyen después de la reordenación. La principal conclusión es que SPAI puede competir con ILU en máquinas paralelas. Además, si aplicamos un algoritmo de reordenación adecuado a las dos estrategias, esta competitividad se mantiene.

Precondicionador	Iter.	$nz(M)$	$nz(M)/nz(A)$	$\|\|MA - I\|\|_F$
Sin precondicionar	548	2205	0,16	854454
ILU(0)	47	14133	1,00	–
SPAI $\varepsilon_k = 0{,}6$	299	3087	0,22	26,77
SPAI $\varepsilon_k = 0{,}5$	169	6615	0,47	22,31
SPAI $\varepsilon_k = 0{,}4$	86	10353	0,73	13,67
SPAI $\varepsilon_k = 0{,}3$	59	11025	0,78	11,45
SPAI $\varepsilon_k = 0{,}2$	37	31782	2,25	8,56

Tabla 6.1: *Resultados de convergencia para orsreg1 con el orden original y BiCGSTAB precondicionado por la izquierda.*

Precondicionador	Iter.	$nz(M)$	$nz(M)/nz(A)$	$\|\|MA-I\|\|_F$
Sin precondicionar	> 2205	2205	0,16	854454
ILU(0)	381	14133	1,00	–
SPAI $\varepsilon_k = 0{,}6$	1396	3593	0,25	25,62
SPAI $\varepsilon_k = 0{,}5$	73	7199	0,51	20,93
SPAI $\varepsilon_k = 0{,}4$	36	10678	0,76	13,02
SPAI $\varepsilon_k = 0{,}3$	21	13850	0,98	8,24
SPAI $\varepsilon_k = 0{,}2$	14	19694	1,39	5,36

Tabla 6.2: *Resultados de convergencia para orsreg1 con Grado Mínimo y BiCGSTAB precondicionado por la izquierda.*

Precondicionador	Iter.	$nz(M)$	$nz(M)/nz(A)$	$\|\|MA-I\|\|_F$
Sin precondicionar	> 2205	2205	0,16	854454
ILU(0)	26	14133	1,00	–
SPAI $\varepsilon_k = 0{,}6$	810	4152	0,29	23,84
SPAI $\varepsilon_k = 0{,}5$	88	7005	0,50	19,57
SPAI $\varepsilon_k = 0{,}4$	27	9304	0,66	11,62
SPAI $\varepsilon_k = 0{,}3$	19	10608	0,75	8,08
SPAI $\varepsilon_k = 0{,}2$	12	13322	0,94	5,74

Tabla 6.3: *Resultados de convergencia para orsreg1 con Reverse CutHill McKee y BiCGSTAB precondicionado por la izquierda.*

El segundo ejemplo (*convdifhor*) es un problema de convección difusión definido en $[0, 1] \times [0, 1]$ por la ecuación,

$$v_1 \frac{\partial u}{\partial x} - K\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) = F$$

donde $v_1 = 10^4\left(y - \frac{1}{2}\right)\left(x - x^2\right)\left(\frac{1}{2} - x\right)$, $K = 10^{-5} - 10^2$, y $F = 10^3 - 1$. La matriz corresponde a una malla no estructurada de elementos finitos con $n = 1960$ y $nz = 13412$.

Las tablas 6.4, 6.5, 6.6 y 6.7 son similares a las del problema anterior. La reducción del número de entradas en la SPAI es del 40-50 % para los casos de reordenación con Grado Mínimo y Cuthill-Mckee Inverso. El algoritmo de Mínimo Vecino no afecta a nz. Además, el número de iteraciones del BiCGSTAB se redujo mediante las renumericaciones del 60-70 %. Como estamos interesados en el efecto de la reordenación de A en las características de los precondicionadores SPAI, en las figuras 6.3, 6.4, 6.5 y 6.6 se muestra la estructura *sparse* de la matriz M con $\varepsilon_k = 0{,}3$ para ordenación Original, Grado Mínimo, Cuthill-Mckee Inverso y Mínimo Vecino, respectivamente. Las entradas no nulas se representan por un punto. La estructuras correspondientes representan matri-

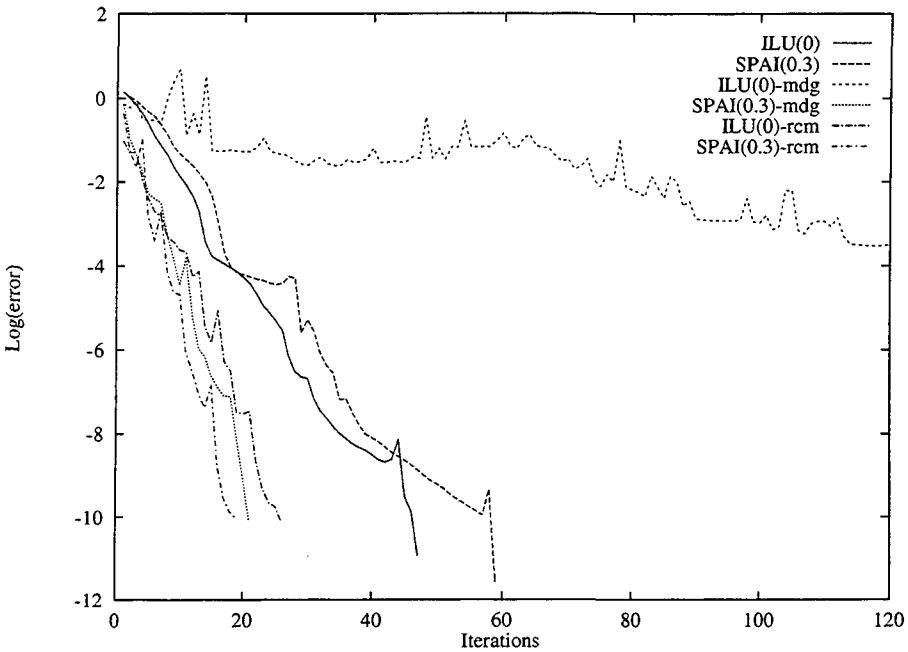

Figura 6.1: *Comparación del comportamiento de BiCGSTAB-ILU(0) y BiCGSTAB-SPAI con reordenación para orsreg1.*

ces llenas, como cabía esperar. Sin embargo, se advierte cierto paralelismo con la estructura de A para las diferentes reordenaciones.

La reducción del ancho de banda y del perfil llevada a cabo por el algoritmo de Cuthill-Mckee Inverso en la matriz A se conserva de alguna manera en la matriz M, incluso cuando existe una tendencia a explotar algunas entradas fuera del perfil. Esto queda ilustrado claramente en la figura 6.5. Los patrones de las matrices SPAI correspondientes a Grado Mínimo y Mínimo Vecino también conservan en parte las estructuras de la matriz A reordenada, respectivamente, aún cuando nuestro algoritmo SPAI no tiene por qué producir matrices M con estructura simétrica. En la figura 6.2 se compara la convergencia de BiCGSTAB-SPAI(0.2) para todos estas reordenaciones. Claramente, las reordenaciones producidas por los algoritmos de Grado Mínimo y Cuthill-Mckee Inverso tiene un efecto beneficioso en la velocidad de convergencia del algoritmo BiCGSTAB-SPAI.

Precondicionador	Iter	$nz(M)$	$nz(M)/nz(A)$	$\|\|MA-I\|\|_F$
Sin precondicionar	>1960	1960	0,15	13279
ILU(0)	74	13412	1,00	–
SPAI $\varepsilon_k = 0{,}6$	414	3161	0,24	22,42
SPAI $\varepsilon_k = 0{,}4$	302	10693	0,80	16,99
SPAI $\varepsilon_k = 0{,}3$	171	21734	1,62	12,78
SPAI $\varepsilon_k = 0{,}2$	83	54406	4,06	8,70
SPAI $\varepsilon_k = 0{,}1$	21	167678	12,50	4,36

Tabla 6.4: *Resultados de convergencia para convdifhor con el orden originial y BiCGSTAB precondicionado por la izquierda.*

Preconditioner	Iter	$nz(M)$	$nz(M)/nz(A)$	$\|\|MA-I\|\|_F$
Unprecond.	>1960	1960	0,15	13279
ILU(0)	57	13412	1,00	–
SPAI $\varepsilon_k = 0{,}6$	166	2617	0,20	19,82
SPAI $\varepsilon_k = 0{,}4$	99	6255	0,47	15,18
SPAI $\varepsilon_k = 0{,}3$	68	11461	0,85	11,42
SPAI $\varepsilon_k = 0{,}2$	40	26992	2,01	7,78
SPAI $\varepsilon_k = 0{,}1$	21	92864	6,92	3,85

Tabla 6.5: *Resultados de convergencia convdifhor con Grado Mínimo y BiCGSTAB precondicionado por la izquierda.*

Preconditioner	Iter	$nz(M)$	$nz(M)/nz(A)$	$\|\|MA-I\|\|_F$
Unprecond.	1477	1960	0,15	13279
ILU(0)	31	13412	1,00	–
SPAI $\varepsilon_k = 0{,}6$	144	2510	0,19	19,51
SPAI $\varepsilon_k = 0{,}4$	92	6126	0,46	15,51
SPAI $\varepsilon_k = 0{,}3$	66	11355	0,85	11,67
SPAI $\varepsilon_k = 0{,}2$	41	26270	1,96	7,98
SPAI $\varepsilon_k = 0{,}1$	18	88093	6,57	4,01

Tabla 6.6: *Resultados de convergencia para convdifhor con Reverse CutHill McKee y BiCGSTAB precondicionado por la izquierda.*

Preconditioner	Iter	$nz(M)$	$nz(M)/nz(A)$	$\|\|MA-I\|\|_F$
Unprecond.	> 1960	1960	0,15	13279
ILU(0)	45	13412	1,00	–
SPAI $\varepsilon_k = 0{,}6$	397	3161	0,23	22,41
SPAI $\varepsilon_k = 0{,}4$	294	10693	0,80	16,98
SPAI $\varepsilon_k = 0{,}3$	173	21734	1,62	12,78
SPAI $\varepsilon_k = 0{,}2$	86	54406	4,06	8,69
SPAI $\varepsilon_k = 0{,}1$	21	167678	12,5	4,36

Tabla 6.7: *Resultados de convergencia para convdifhor con Mínimo vecino y BiCGSTAB precondicionado por la izquierda.*

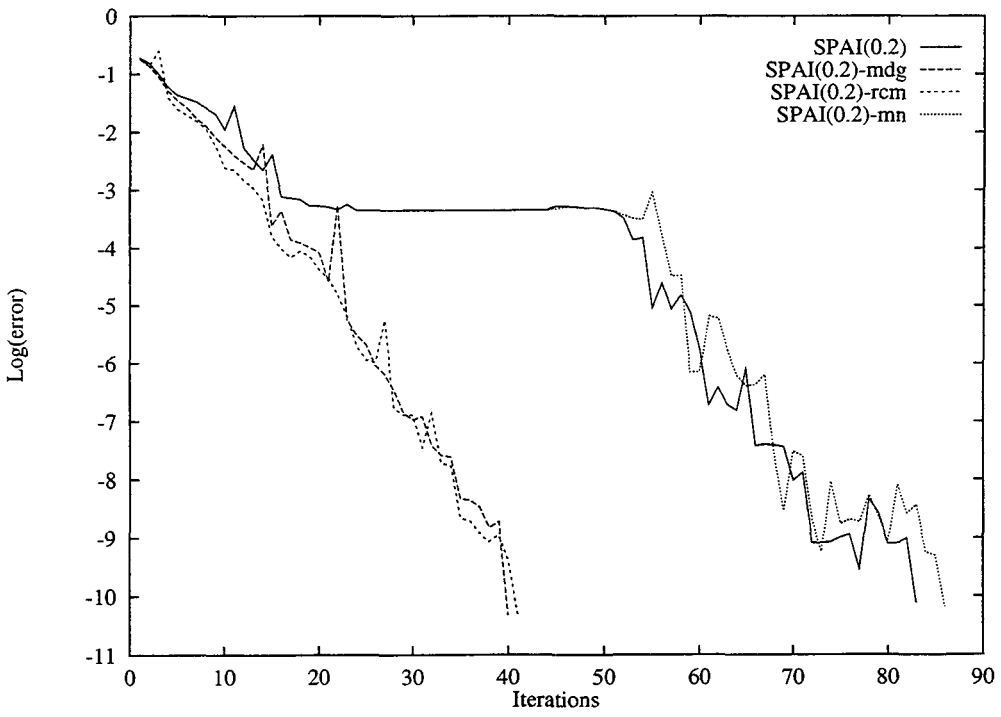

Figura 6.2: *Comparación del comportamiento de BiCGSTAB-SPAI con reordenación para convdifhor.*

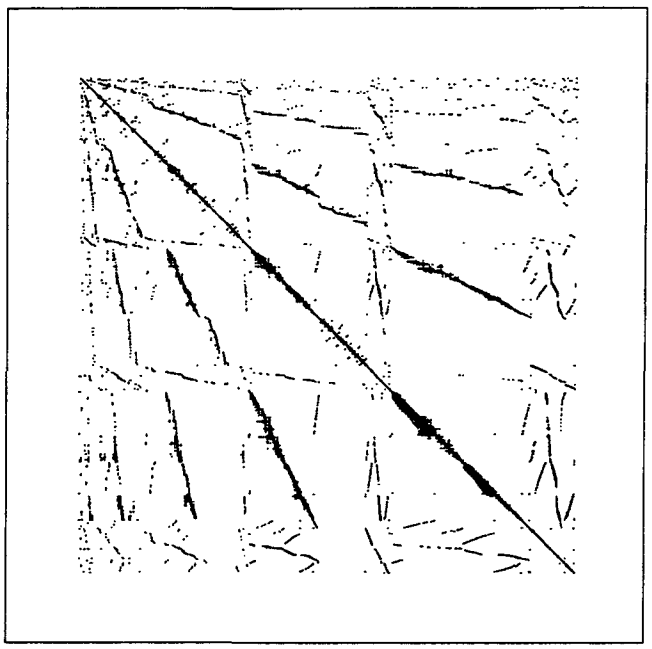

Figura 6.3: *Patrón de sparsidad de la matriz SPAI(0.3) original para convdifhor.*

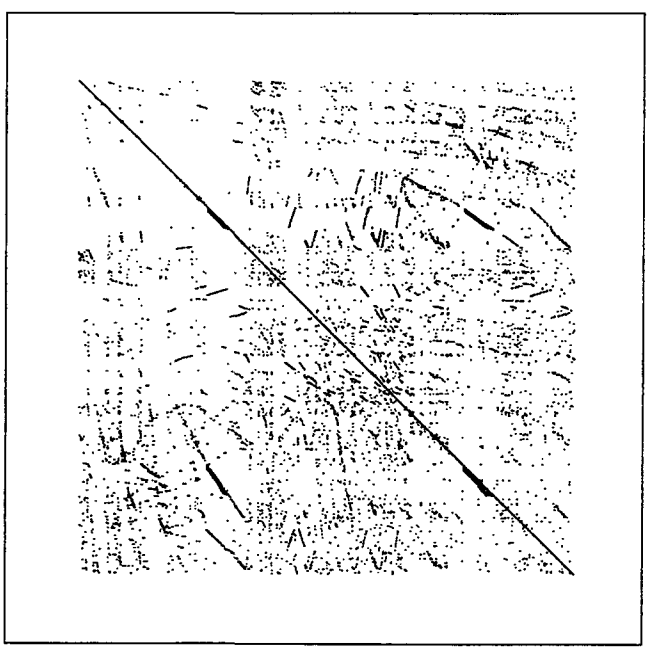

Figura 6.4: *Patrón de sparsidad de la matriz SPAI(0.3) reordenada con grado mínimo para convdifhor.*

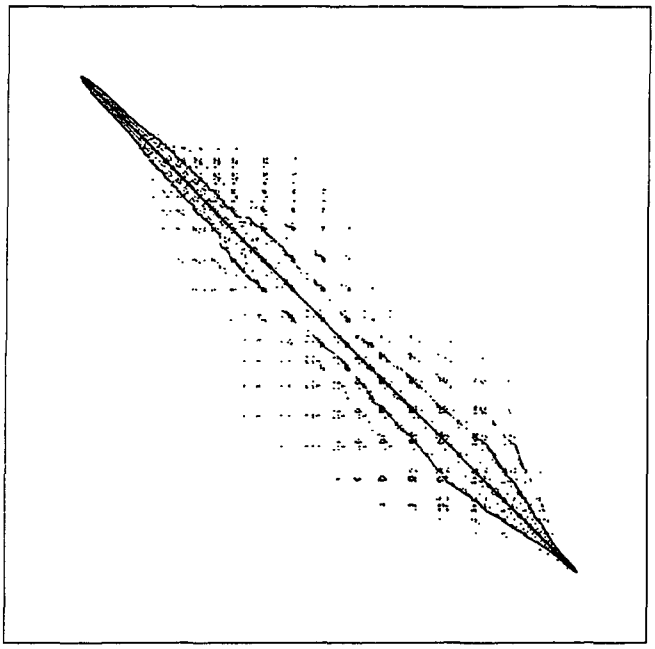

Figura 6.5: *Patrón de sparsidad de la matriz SPAI(0.3) reordenada con Reverse Cuthill-Mckee para convdifhor.*

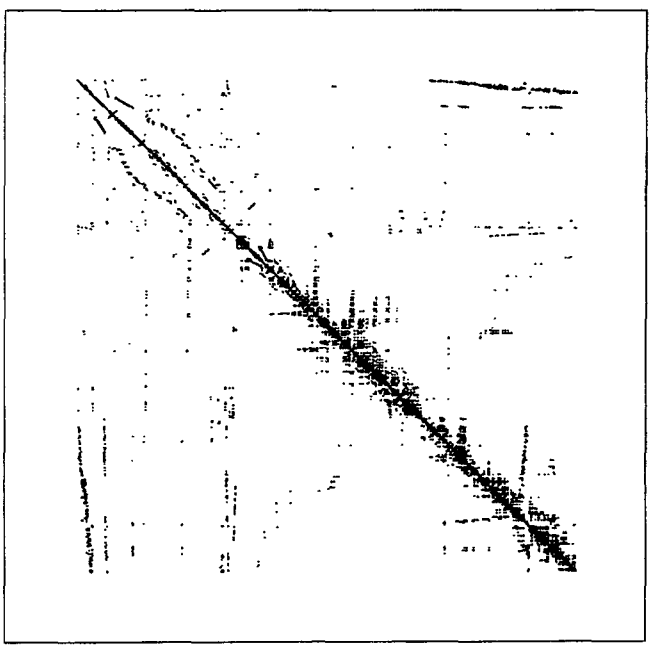

Figura 6.6: *Patrón de sparsidad de la matriz SPAI(0.3) reordenada con mínimo vecino para convdifhor.*

El tercer ejemplo es un problema de convección difusión (*cuaref*) definido en $[0,1] \times [0,1]$ por la ecuación,

$$v_1 \frac{\partial u}{\partial x} + v_2 \frac{\partial u}{\partial y} - K\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) = 0$$

donde $v_1 = C\left(y - \frac{1}{2}\right)\left(x - x^2\right), v_2 = C\left(\frac{1}{2} - x\right)\left(y - y^2\right)$, $K = 1$, y $C = 10^5$ La matriz corresponde a un paso de refinamiento de una malla no estructurada de elementos finitos con $n = 7520$ y $nz = 52120$.

Precondicionador	Iter	$nz(M)$	$nz(M)/nz(A)$	$\|\|MA - I\|\|_F$
Sin precondicionar	1740	7520	0,14	725,81
ILU(0)	378	52120	1,00	–
SPAI $\varepsilon_k = 0{,}4$	584	32227	0,62	32,78
SPAI $\varepsilon_k = 0{,}3$	455	61400	1,18	25,24
SPAI $\varepsilon_k = 0{,}2$	214	152078	2,92	17,12

Tabla 6.8: *Resultados de convergencia para cuaref con ordenación original y BiCGSTAB precondicionado por la izquierda.*

Precondicionador	Iter	$nz(M)$	$nz(M)/nz(A)$	$\|\|MA - I\|\|_F$
Sin precondicionar	395	7520	0,14	705,14
ILU(0)	69	52120	1,00	–
SPAI $\varepsilon_k = 0{,}4$	125	25256	0,48	31,86
SPAI $\varepsilon_k = 0{,}3$	79	47147	0,90	24,66
SPAI $\varepsilon_k = 0{,}2$	51	106302	2,04	16,77

Tabla 6.9: *Resultados de convergencia para cuaref con Grado Mínimo y BiCGSTAB precondicionado por la izquierda.*

Precondicionador	Iter	$nz(M)$	$nz(M)/nz(A)$	$\|\|MA - I\|\|_F$
Sin precondicionar	> 7520	7520	0,14	703,54
ILU(0)	20	52120	1,00	–
SPAI $\varepsilon_k = 0{,}4$	215	24684	0,47	31,69
SPAI $\varepsilon_k = 0{,}3$	178	45808	0,88	24,56
SPAI $\varepsilon_k = 0{,}2$	72	103391	1,98	16,67

Tabla 6.10: *Resultados de convergencia para cuaref con Reverse CutHill McKee y BiCGSTAB precondicionado por la izquierda.*

Precondicionador	Iter	$nz(M)$	$nz(M)/nz(A)$	$\|\|MA-I\|\|_F$
Sin precondicionar	1739	7520	0,14	725,81
ILU(0)	102	52120	1,00	–
SPAI $\varepsilon_k = 0{,}4$	577	32227	0,62	32,78
SPAI $\varepsilon_k = 0{,}3$	410	61400	1,18	25,24
SPAI $\varepsilon_k = 0{,}2$	209	152078	2,92	17,12

Tabla 6.11: *Resultados de convergencia para cuaref con Mínimo vecino y BiCGSTAB precondicionado por la izquierda.*

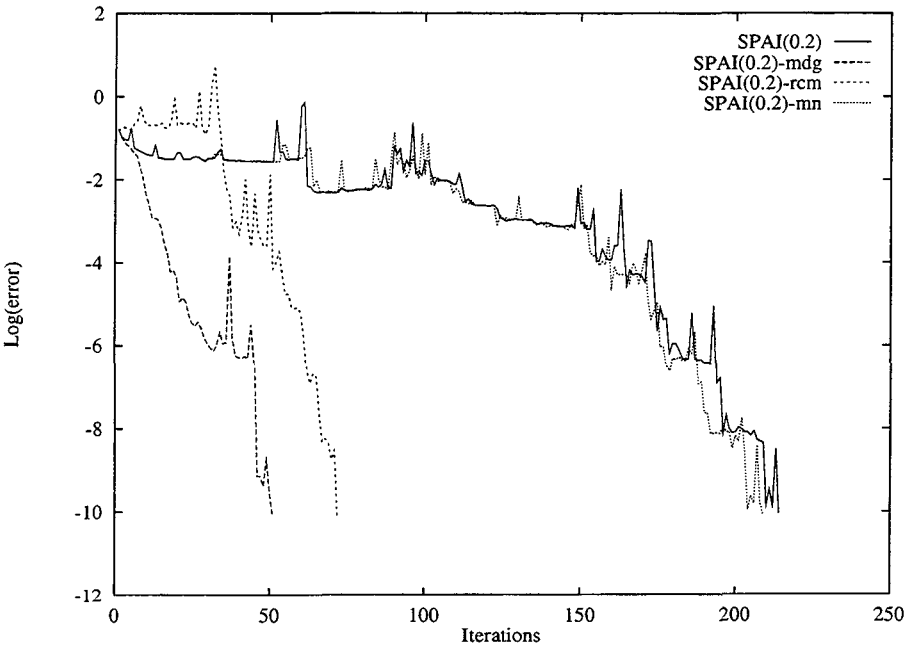

Figura 6.7: *Comparación del comportamiento de BiCGSTAB-SPAI con reordenación para cuaref.*

Las tablas 6.8, 6.9, 6.10 y 6.11 indican el comportamiento de los precondicionadores ILU(0) y SPAI para *cuaref*. La reducción del número de entradas en SPAI para una tolerancia dada para la aproximada inversa, es también evidente aquí (del 20 al 30 % aproximadamente para Grado Mínimo y Cuthill-Mckee Inverso). Además, el número de iteraciones de BiCGSTAB se reduce drásticamente un 75-85 % y un 60-70 %, respectivamente. El Patrón de *sparsidad* de las matrices SPAI(0.2) correspondientes a la ordenación Original y a los algoritmos de reordenación considerados aquí se muestran en las figuras 6.8, 6.9, 6.10 y 6.11.

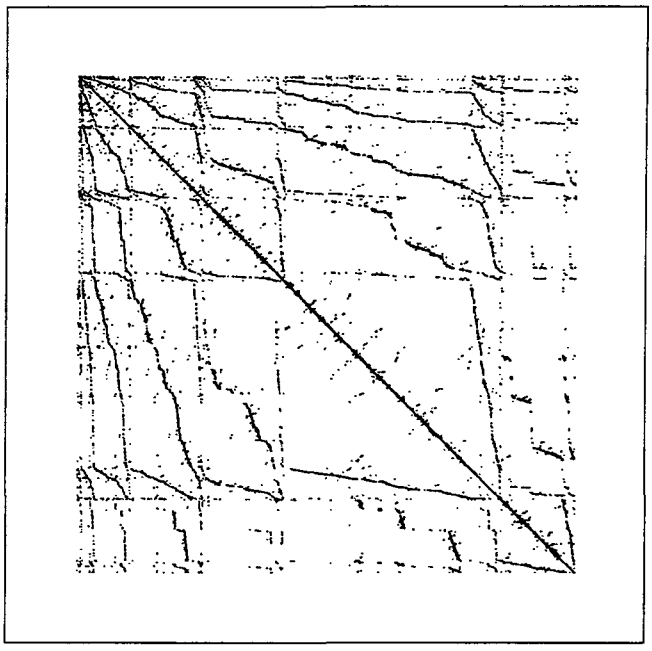

Figura 6.8: *Patrón de sparsidad de la matriz SPAI(0.3) original para cuaref.*

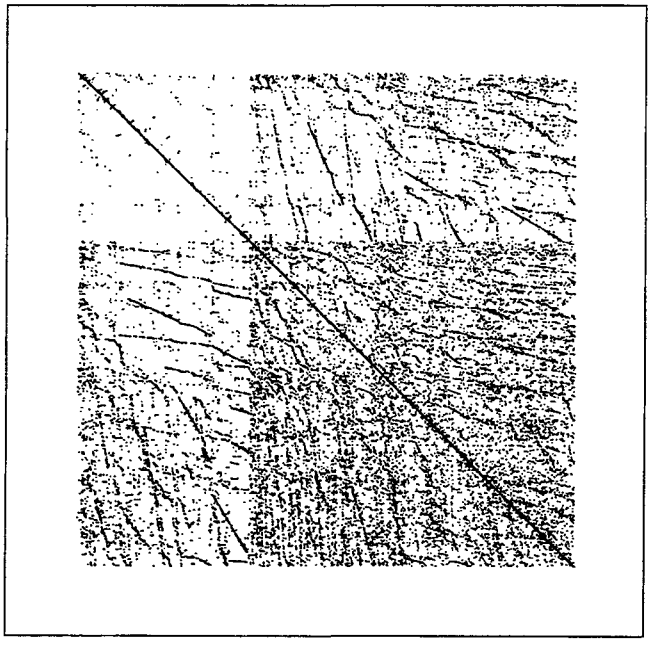

Figura 6.9: *Patrón de sparsidad de la matriz SPAI(0.3) reordenada con Grado M'inimo para cuaref.*

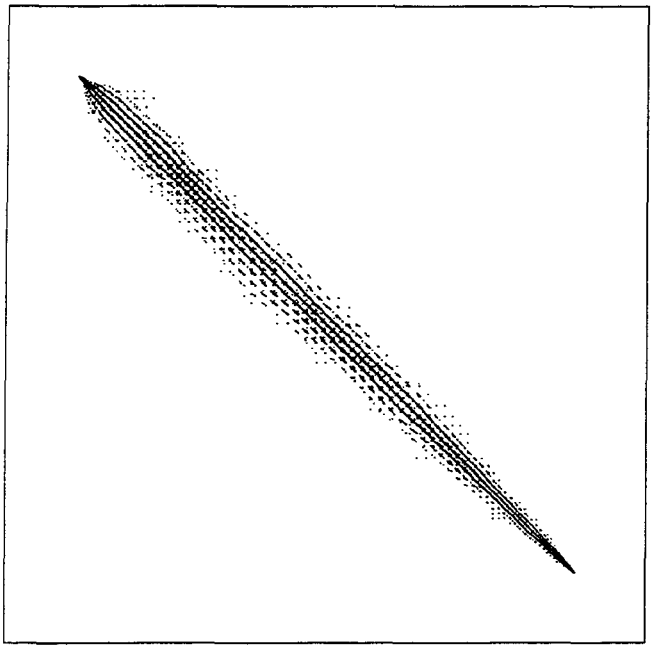

Figura 6.10: *Patrón de sparsidad de la matriz SPAI(0.3) reordenada con Reverse Cuthill-Mckee para cuaref.*

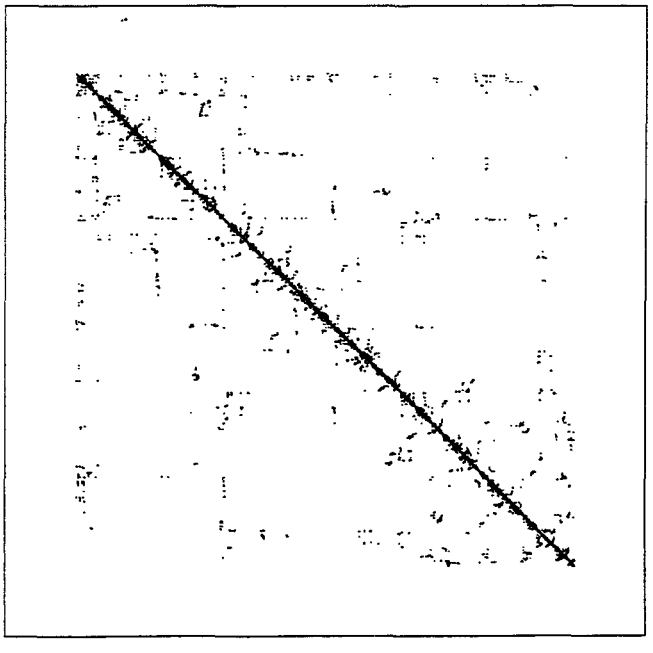

Figura 6.11: *Patrón de sparsidad de la matriz SPAI(0.3) reordenada con Mínimo Vecino para cuaref.*

Concluimos, como en el problema anterior y en otros llevados a cabo no incluidos aquí, que la estructura *sparse* de SPAI parece partir de una estructura similar a la típica de A obtenida de la reordenación, y tiende a una matriz llena a medida que aumentamos la precisión. La figura 6.7 no aporta diferencias significativas a la figura 6.2 del segundo problema. Los algoritmos de Grado Mínimo y de Cuthill-Mckee Inverso son preferibles al Mínimo Vecino o a la ordenación Original. Sin embargo, hemos notado que si nz aumenta, las diferencias entre Grado Mínimo y Cuthill-Mckee Inverso son más apreciables a favor del primero.

Capítulo 7

Conclusiones y líneas futuras

El carácter prehilbertiano de la norma matricial de Frobenius permite obtener en el Capítulo 3 el mejor precondicionador N para el sistema (2.1) en cada subespacio $\mathcal{S}$ de $\mathcal{M}_n(\mathbb{R})$ mediante la proyección ortogonal de la identidad sobre el subespacio $A\mathcal{S}$. Consecuentemente, se obtiene la ecuación (3.4), que representa la distancia mínima $d(I, A\mathcal{S})$. Estas consideraciones hacen posible el desarrollo de expresiones explícitas tanto para N como para $||AN-I||_F$ usando una base ortogonal de $A\mathcal{S}$. Posteriormente, el método de ortogonalización de Gram-Schmidt generaliza estas fórmulas para cualquier base de $\mathcal{S}$. Además, el coste computacional de las últimas expresiones se reduce considerablemente usando la descomposición de $A\mathcal{S}$ como suma directa de espacios mutuamente ortogonales $A\mathcal{S}_j$.

La aplicación de los resultados anteriores al caso de los precondicionadores *sparse* nos lleva evidentemente a expresiones cuyos cálculos son inherentemente paralelos, puesto que las columnas n_j de N pueden obtenerse independientemente unas de otras. Asimismo, el patrón de *sparsidad* del precondicionador N se captura automáticamente (no es un patrón de *sparsidad* fijo) puesto que cada nueva entrada n_{ij} es equivalente a extender el subespacio precondicionador $\mathcal{S}$ a $\mathcal{S} \oplus span\{M_{i,j}\}$.

Por otra parte, el mejor precondicionador simétrico N se obtiene, de forma alternativa, de la condición

$$AN - I \in (A\mathcal{S}_n)^{\perp} = \mathcal{H}_n A$$

Entonces, usando la descomposición en valores singulares de la matriz A, se obtienen las cotas superiores e inferiores de $||AN - I||_F$. Estas cotas dependen de $||A - A^t||_F$ y están más cercanas cuanto más cercano esté el número de condición $\kappa_2(A)$ a la unidad.

Las diferentes expresiones obtenidas para $||AN - I||_F$ directamente, caracterizan aquellas matrices A para las cuales el subespacio precondicionador $\mathcal{S}$ contiene inversa aproximada en el sentido estricto, es decir, $||AN - I||_F < 1$.

El algoritmo propuesto en el Capítulo 5 permite calcular un precondicionador *sparse* M_0 de una matriz *sparse* no simétrica A. Como las columnas (o filas)

de M_0 se obtienen independientemente unas de otras, los cálculos pueden realizarse en paralelo. El patrón de *sparsidad* de estos precondicionadores se construye dinámicamente partiendo del diagonal aumentando el número de entradas no nulas. Evidentemente, este precondicionador puede ser competitivo, si se trabaja en paralelo, con los tradicionales precondicionadores implícitos. Esto dependerá en gran medida del problema y de la arquitectura del ordenador. No obstante, se ha demostrado de forma teórica y práctica la eficacia de este precondicionador en la mejora la convergencia de los métodos iterativos.

Asimismo, el algoritmo IAI propuesto calcula un precondicionador mejor a partir de una inversa aproximada dada. Sólo es necesario calcular productos matriz-vector, por lo que éste puede ser usado directamente para mejorar la efectividad de la aproximada inversa en la convergencia de los métodos iterativos. Se muestran los resultados teóricos sobre esta mejora. Finalmente, el algoritmo IAI permite resolver sistemas lineales en un número de pasos conocidos *a priori* para una tolerancia prefijada. La cantidad necesaria de productos matriz-vector depende directamente de la torerancia y de la calidad de la inversa aproximada inicial, pero nunca del número de incógnitas, lo cual es una propiedad bastante interesante.

Se ha estudiado por muchos autores el efecto de la reordenación sobre la convergencia de los métodos basados en subespacios de Krylov con precondicionamiento para resolver sistemas de ecuaciones lineales no simétricos. Este efecto beneficioso se muestra en, [96], [21] para precondicionadores basados en factorización incompleta. En el Capítulo 6 hemos probado experimentalmente que las técnicas de reordenación tienen efectos beneficiosos en la ejecución de inversas aproximadas *sparse* utilizadas como precondicionadores en los métodos iterativos basados en subespacios de Krylov. La reducción del número de entradas no nulas debido a la reordenación permite obtener inversas aproximadas *sparse* con una exactitud similar a las obtenidas sin reordenación, pero con menores requerimientos de almacenamiento y coste computacional. Además, la reordenación produce precondicionadores con mejores cualidades puesto que generalmente se reduce el número de pasos para alcanzar la convergencia de los métodos iterativos. Los experimentos numéricos, pues, parecen indicar que un orden adecuado puede mejorar la eficiencia de las inversas aproximadas aquí propuestas. Sin embargo, deben realizarse más investigaciones sobre la reducción de las entradas no nulas en M_0 para un número similar de iteraciones de los métodos de Krylov. También sería interesante estudiar el comportamiento de diferentes algoritmos de reordenación para un sistema de ecuaciones dado, proveniente de la discretización de problemas similares (por ejemplo, nos hemos centrado aquí en la ecuación de convección-difusión).

Proponemos la investigación del efecto de otras técnicas de reordenación que tengan en cuenta las entradas numéricas de A (ver [79], [83]). Aún cuando estas técnicas son caras, en el caso en que haya que resolver muchos sistemas de ecuaciones lineales con la misma matriz, dichas técnicas pueden ser competitivas en máquinas en paralelo.

Bibliografía

[1] P. ALMEIDA, *Solución directa de sistemas sparse mediante grafos*, Departmento de Matemáticas, Universidad de Las Palmas de Gran Canaria, España, 1990.

[2] F. ALVARADO Y H. DAG, Sparsified and incomplete sparse factored inverse preconditioners, in: *Proceedings of the 1992 Copper Mountain conference on Iterative Methods*, **1**, 9-14, april 1992.

[3] W.E. ARNOLDI, The Principle of Minimized Iteration in the Solution of the Matrix Eingenvalue Problem, *Quart. Appl. Math.*, **9**, pp.17-29 (1951).

[4] E. ASPLUND, Inverses of matrices $\{a_{ij}\}$ which satisfy $a_{ij} = 0$ for $j > i + p$, *Mathematica Scandinavia*,**7**,57-60, 1959.

[5] O. AXELSSON, Incomplete block-matrix factorization preconditioning methods. The ultimate answer?, *J. Comp. Appl. Math.*, **12,** 13, 3-18, 1985.

[6] O. AXELSSON, A Restarted Version of a Generalized Preconditioned Conjugate Gradient Method, *Comunications in Applied Numerical Methods*, **4**, 521-530, 1988.

[7] O. AXELSSON, A survey of preconditioned iterative methods for linear systems of algebraic equations, *BIT*, **25**, 166-187, 1995.

[8] O. AXELSSON *Iterative Solution Methods*, Cambridge University Press, Cambridge, U.K., 1994.

[9] O. AXELSSON, S. BRINKKEMPER Y V.P. IL'IN, On some versions of incomplete block-matrix factorization iterative methods, *Lin. Alg. Appl.*, **58**, 3-15, 1984.

[10] O. AXELSSON Y I. KAPORIN, Error norm estimation and stopping criteria in preconditioned conjugated gradient iterations, *Num. Lin. Alg. Appl.*, **8**, 265-286, 2001.

[11] O. AXELSSON Y L. YU. KOLOTILINA, Diagonally compensated reduction and related preconditioning methods, *Num. Lin. Alg. Appl.*, **1**, 155-177, 1994.

[12] M. BENSON, *Iterative solution of large scale linear systems*, Master's thesis, Lakehead University, Thunder Bay, Canada, 1973.

[13] M.W. BENSON, P.O. FREDERICKSON, Iterative solution of large sparse linear systems arising in certain multidimensional approximation problems, *Utilitas Math.*, **22**, 127-140, 1982.

[14] M. BENZI, *A direct row-projection method for sparse linear systems*. Department of Mathematics. Norh Carolina State University, Raleigh, NC, 1993.

[15] M. BENZI, J.K. CULLUM Y M. TŮMA, Robust approximate inverse preconditioning for the conjugate gradient method, *Los Alamos Technical Report LA-UR-99-2899*, 1-15, 1999.

[16] M. BENZI Y G.H. GOLUB, Bounds for the entries of matrix functions with applications to preconditioning, *BIT*, **39**, 3, 417-438, 1999.

[17] M. BENZI, J.C. HAWS Y M. TŮMA, Preconditioning highly indefinite and nonsummetric matrices, *Los Alamos Technical Report LA-UR-994857*, 1-25, 1999.

[18] M. BENZI, W. JOUBERT Y G. MATEESCU, Numerical experiments with parallel orderings for ILU preconditioners, *Elect., Transact., Num. Anal.*, **8**, 88-114, 1999.

[19] M. BENZI, J. MARÍN, M. TŮMA, A two -level parallel preconditioner based on sparse approximate inverses, *Iter. Meth. Sci. Comp. II*, D.R., Kincaid et. al. editores, 1-11, 1999.

[20] M. BENZI, C.D. MEYER Y M. TŮMA, A Sparse Approximate Inverse Preconditioner for the Conjugate Gradient Method, *SIAM J. Sci. Comput.*, **17**, 5, 1135-1149, 1996.

[21] M. BENZI, D.B. SZYLD Y A. VAN DUIN, Orderings for incomplete factorization preconditioning of nonsymmetric problems, *SIAM J. Sci. Comput.*, **20**, 5, 1652-1670, 1999.

[22] M. BENZI Y M. TUMA, A sparse approximate inverse preconditioner for nonsymmetric linear systems, *SIAM J. Sci. Comput.*, **19**, 3, 968-994, 1998.

[23] M. BENZI Y M. TUMA, Sparse matrix orderings for factorized inverse preconditioners, in: *Proceedings of the 1998 Cooper Mountain Conference on Iterative Methods*, marzo 30-abril 3, 1998.

[24] M. BENZI Y M. TŮMA, A comparative study of sparse approximate inverse preconditioners, *Appl. Num. Math.*, **30**, 305-340, 1999.

[25] M. BENZI Y M. TŮMA, Orderings for factorized sparse approximate inverse preconditioners, *SIAM J. Sci. Comput.*, to appear.

[26] L. BERGAMASCHI, G. PINI Y F.SARTORETTO, Approximate inverse preconditioning in the parallel solution of sparse eigenproblems, *Num. Lin. Alg. Appl.*, **7**, 99-116, 2000.

[27] E. BODEWIG, *Matrix calculus, 2nd revised and enlarged edition*, Interscience, New York, 1959.

[28] R. BRIDSON Y W.P. TANG, Ordering, anisitropy and factored sparse approximate inverses, Preprint, Department of Computer Science, University of Waterloo., 1998.

[29] B. CARPENTIENRI, I.S. DUFF Y L. GIRAUD, Sparse pattern selecion strategies for robust Frobenius-norm minimization preconditioners in electromagnetism, *Num. Lin. Alg. Appl.*, **7**, 667-685, 2000.

[30] L. CESARI, Sulla risoluzione dei sistemi di equazione lineari per approssimazioni succesive, *Atti. Accad. Naz. Lincei, Rend. Cl. Sci. Fis. Mat. Nat.*, **25**, 422-428, 1937.

[31] T.F. CHAN, E. GALLOPOULOS, V. SIMONCINI, T. SZETO Y C.H. TONG, A Quasi-Minimal Residual Variant of the Bi-CGSTAB Algorithm for Nonsymmetric Systems, *SIAM J. Sci. Statist. Comput.*, **15**, 338-247, 1994.

[32] E. CHOW, *Robust preconditioning for sparse linear systems*, Department of Computer Science, University of Minnesota, Minneapolis, MN, 1997.

[33] E. CHOW, A priori sparsity patterns for parallel sparse approximate inverse preconditioners, *SIAM J. Sci. Comput.*, **21**, 5, 1804-1822, 2000.

[34] E. CHOW Y Y. SAAD, Approximate inverse techniques for block-partitioned matrices, *SIAM J. Sci. Comput.*, **18**, 6, 1657-1675, 1997.

[35] E.,CHOW Y Y. SAAD, Experimental study of ILU preconditioners for indefinite matrices, *J. Comp. Appl. Math*, **86**, 387-414, 1997.

[36] E. CHOW Y Y. SAAD, Approximate inverse preconditioners via sparse-sparse iterations, *SIAM J. Sci. Comput.*, **19**, 3, 995-1023, 1998.

[37] P. CONCUS, G.H. GOLUB Y G. MEURANT, Block predondicionint for the conjugate gradient method,*SIAM J. Sci. Stat. Comp.*, **6**, 220-252, 1985.

[38] D.J.F. COSGROVE, J.C. DÍAZ Y A. GRIEWANK, Approximate inverse preconditionings for sparse linear systems, *Int. J. Comput. Math.*, **44**, 91-110, 1992.

[39] E.H. CUTHILL Y J.M. MCKEE, Reducing the Bandwidth of Sparse Symmetric Matrices, *Proc. 24th National Conference of the Association for Computing Machinery*, Brondon Press editores, New Jersey, 157-172, 1969.

[40] H. DAG, *Iterative methods and parallel computation for power systems*. Department of Electrical Engineering, University of Wisconsin, Madison, WI, 1996.

[41] J.C. DÍAZ Y C.G. MACEDO JR., Fully vectorizable block preconditionings with approximate inverses for non-symmetric systems of equations, *Int. Journ. Num. Meth. Eng.*, **27**, 501-522, 1989.

[42] P.F. DUBOIS, A. GREENBAUM Y G.H. RODRIGUE, Approximating the inverse of a matrix for use in iterative algorithms on vector processors, *Computing*, **22**, 257-268, 1979.

[43] I.S. DUFF, R.G. GRIMESY Y J.G. LEWIS, Sparse matrix test problems, *ACM Trans. Math. Software*, **15** ,1-14, 1989.

[44] A.C.N. VAN DUIN Y H. WIJSHOFF, Scalable parallel preconditioning with the sparse approximate inverse triangular systems, Preprint, Computer Science Departmente, University of Leiden, Leiden, the Netherland, 1996.

[45] L.C. DUTTO, The effect of ordering on preconditioned GMRES algorithm, for solving the compressible Navier-Stokes equations, *Int. J. Num. Meth. Eng.*, **36**, 457-497, 1993.

[46] H.C. ELMAN, A stability analysis of incomplete LU factorization, *Math. Comp.*, **47**, 191-217, 1986.

[47] A.M. ERISMAN Y W.F. TINNEY, On computing certain elements of the inverse of a sparse matrix, *Comm. ACM*, **18**, 177-179, 1975.

[48] M.R. FIELD, An efficient parallel preconditioner for the conjugate gradient algorithm, Hitachi Dublin Laboratory Technical Report HDL-TR-97-175, Cublin, Ireland, 1997.

[49] E. FLÓREZ, M.D. GARCÍA, L. GONZÁLEZ Y G. MONTERO, The effect of orderings on sparse approximate inverse preconditioners for non-symmetric problems, *Adv. Engng. Software*, **33**, 7-10, 611-619, 2002.

[50] P.O. FREDERICKSON, Fast approximate inversion of large sparse linear systems, *Math. Report*, **7**, Lakehead University, Thunder Bay, Canada, 1975.

[51] M. GALÁN, G. MONTERO Y G. WINTER, Variable GMRES: an Optimizing Self-Configuring Implementation of GMRES(k) with Dynamic Memory Allocation, *Tech. Rep. of CEANI*, Las Palmas, 1994.

[52] M. GALÁN, G. MONTERO Y G. WINTER, A Direct Solver for the Least Square Problem Arising From GMRES(k), *Com. Num. Meth. Eng.*, **10**, 743-749, 1994.

[53] A. GEORGE, Computer Implementation of the Finite Element Method, *Report Stan CS-7*, 1-208, 1971.

[54] A. GEORGE Y J.W. LIU, The Evolution of the Minimum Degree Ordering Algorithms, *SIAM Rev.*, **31**, 1-19, 1989.

[55] I.C. GOHBERGH Y M.G. KREIN, *Introduction to the Theory of Linear Nonselfadjoint Operators in Hilbert Space*, Ed. Translations of Mathematical Monographs, **18**, *American Mathematical Society*, Providence, Rhode Island, U.S.A., 1991.

[56] L. GONZÁLEZ, G. MONTERO Y E. FLÓREZ, Aproximate Inverse Preconditioners Using Frobenius Inner Product I: Theorical Results, *8th ILAS Conference*, Barcelona, 1999.

[57] N.I.M. GOULD Y J.A. SCOTT, Sparse approximate-inverse preconditioners using norm-minimization techniques, *SIAM, J. Sci. Comput.*, **19**, 2, 605-625, 1998.

[58] G.A. GRAVVANIS, Approximate inverse banded matrix techniques, *Eng. Comp.*, **16**, 3, 337-346, 1999.

[59] G.A. GRAVVANIS, The convergence rate and complexity of explicit preconditioned conjugate gradient methods based on approximate inverse banded matrix techniques, *Neural, Parallel & Scientific Computations*, **9**, 355-368, 2001.

[60] M. GROTE Y H.D. SIMON, Parallel Preconditioning and Approximate Inverses on the Connection Machine, *Sixth SIAM Conference on Parallel Processing for Scientific Computing*, **2**, 519-523, Philadelphia, 1992.

[61] M. GROTE Y T. HUCKLE, Parallel preconditioning with sparse approximate inverses, *SIAM J. Sci. Comput.* **18**, 3, 838-853, 1997.

[62] I. GUSTAFSSON Y D. LINDSKOG, Completely parallelizable preconditioning methods, *Num. Lin. Alg. Appl.*, **2**, 447-465, 1995.

[63] M.R. HESTENES Y E. STIEFEL, Methods of Conjugate Gradients for Solving Linear Systems, *Jour. Res. Nat. Bur. Sta.* **49**, 6, 409-436, 1952.

[64] R.A. HORN Y C.R JOHNSON, *Topics in Matrix Analysis*, Cambridge University Press, Cambridge, U.K., 1991.

[65] R.A. HORN Y C.R. JOHNSON, *Topics in Matrix Analysis.* Ed. Cambridge University Press, Cambridge, 1999.

[66] T.K. HUCKLE, Approximate sparsity patterns for the inverse of a matrix and preconditioning, in: *Proceedings of the 15th IMACS World Congress 1997 on Scientific Computation, Modelling and Applied Mathematics,* R.,Weiss and W. Schönauer editores, **2**, 569-574, 1997.

[67] T.K. HUCKLE, Efficient computation of sparse approximate inverses, *Num. Lin. Alg. Appl.*, **5**, 57-71, 1998.

[68] O.G. JOHNSON, C.A. MICCHELLI Y G. PAUL, Polynomial preconditioning for conjugate gradient calculations, *SIAM J. Num. Anal.*, **20**, 362-376, 1986.

[69] I.E. KAPORIN, New convergence results and preconditioning strategies for the conjugate gradient method, *Num. Lin. Alg. Appl.*, **1**, 179-210, 1994.

[70] I.E. KAPORIN, High quality preconditioning of a general summetric positive definite matrix based on its $U^TU + U^TR + R^TU$- decomposition, *Num. Lin. Alg. Appl.*, **5**, 483-509, 1998.

[71] C.T. KELLEY, , *Iterative methods for Linear and Nonlinear Equations,* Frontiers in Applied Mathematics, *SIAM,* Philadelphia, 1995.

[72] S.A. KHARCHENKO, L.YU. KOLOTILINA, A.A. NIKISHIN Y A. YU. YEREMIN, A robust AINV-type method for constructing sparse approximate inverse preconditioners in factored form, *Num. Lin. Alg. Appl.*, **8**, 165-179, 2001.

[73] L.Y. KOLOTILINA, On approximate inverses of block H-matrices, *Num. Anal. Math. Mod.*, Moscu, 1989.

[74] L. YU. KOLOTILINA, A.A. NIKISHIN Y A.YU. YEREMIN, Factorized sparse approximate inverse preconditionings IV: simple approaches to rising efficiency, *Num. Lin. Alg. Appl.*, **6**, 515-531, 1999.

[75] L. YU. KOLOTILINA Y A. YU. YEREMIN, On a family of two-level preconditionings of the incomplete block factorization type, *Sov. J. Numer. Anal. Math. Modelling*, **1**, 292-320, 1986.

[76] L. YU. KOLOTILINA Y A. YU. YEREMIN, Factorized sparse approximate inverse preconditioning I. Theory, *SIAM J. Matrix Anal. Appl.*, **14**, 45-58, 1993.

[77] L. YU. KOLOTILINA Y A. YU. YEREMIN, Factorized sparse approximate inverse preconditioning II. Solution of 3D FE systems on massively parallel computers, *Internat. J. High Speed Comp.*, **7**, 191-215, 1995.

[78] C. LANCZOS, Solution of Systems of Linear Equations by Minimized Iterations, *Jour. Res. Nat. Bur. Sta.* 49, 1, pp. 33-53, (1952).

[79] R.R. LEWIS, Simulated annealing for profile and fill reduction of sparse matrices. *Int. J. Num. Meth. Eng*, **37**, 905-925, 1994.

[80] T.A. MANTEUFFEL, An incomplete factorization technique for positive definite linear systems, *Math. Com.*, **34**, 473-497, 1980.

[81] G. MARTIN, Methodes de Preconditionnemen par Factorisation Incomplete, *Memoire de Maitrise*, Universite Laval, Quebec, Canada, 1991.

[82] M. MENDOZA, M. RAYDAN Y P. TARAZAGA, Computing the nearest diagonally dominant matrix, *Num. Lin. Alg. Appl.*, **5**, 461-474, 1998.

[83] G. MONTERO, M. GALÁN, P. CUESTA Y G. WINTER, Effects of stochastic ordering on preconditioned GMRES algorithm, in B.H.V. Topping & M. Papadrakakis editors,.*Advances in Structural Optimization*, 241-246. Civil-Comp Press, Edinburgh, 1994.

[84] G. MONTERO, L. GONZÁLEZ, D. GARCÍA, Y A. SUÁREZ, Aproximate Inverse Using Frobenius Inner Product II: Computational Aspects, *8th ILAS Conference*, Barcelona, 1999.

[85] G. MONTERO, L. GONZÁLEZ, E. FLÓREZ, M.D. GARCÍA Y A. SUÁREZ, Short Communication: Approximate inverse computation using Frobenius inner product, *Num. Lin. Alg. Appl.*, **1**,1-30, 2000.

[86] G. MONTERO, R. MONTENEGRO, G. WINTER Y L. FERRAGUT, Aplicación de Esquemas EBE en Procesos Adaptativos, *Rev. Int. Met. Num. Cal. Dis. Ing.*, **6**, 311-332, 1990.

[87] G. MONTERO Y A. SUÁREZ, Left-Right Preconditioning Versions of BCG-Like Methods, *Int. J. Neur., Par. & Sci. Comput.*, **3**, 487-501, 1995.

[88] N.M. NACHTTIGAL, L. REICHEL Y L.N. TREFETHEN, A Hybrid GMRES Algorithm for Nonsymmetric Linear Systems, *SIAM J. Matr. Anal. Appl.*, **13**, 3, 778-795 1992.

[89] H.L. ONG, Fast approximate solution of large-scale sparse linear systems, *J. Comp. Appl. Math.*, **10**, 45-54, 1984.

[90] A. PIETSCH, *Eigenvalues and s-numbers*, Cambridge Studies in Advanced Mathematics, **13**, Cambridge University Presss, Cambridge, U.K., 1987.

[91] C. RIDDELL, B. BENDRIEM, M.H. BOURGUIGNON Y J. KERNEVEZ, The approximate inverse and conjugate gradient: non-symmetrical algorithms for fast attenuation correction in SPECT, *Phys. Med. Biol.*, **40**, 260-281, 1995.

[92] Y. SAAD Y M. SCHULTZ, GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems, *SIAM J. Sci. Statist. Comput.*, **7**, 856-869, 1986.

[93] Y. SAAD, A Flexible Inner-Outer Preconditioned GMRES Algorithm, *SIAM J. Sci. Comput.*, **14**, 461-469, 1993.

[94] Y. SAAD, Iterative Methods for Sparse Linear Systems, Editorial PWS Publishing Company, Boston, 1996.

[95] P. SONNEVELD, CGS: a Fast Lanczos-Type Solver for Nonsymmetric Linear Systems, *SIAM J.Sci. Statist. Comput.* **10**, 36-52, 1989.

[96] A. SUÁREZ, Contributions to Biorthogonalizations Algorithms for the Solution of Linear Equations, Ph. D. thesis, Mathematics Department, University of Las Palmas de Gran Canaria, Canary Islands, Spain (in Spanish), 1995.

[97] A. SUÁREZ, D. GARCÍA, E. FLÓREZ Y G. MONTERO, Preconditioning Krylov Methods, G. Winter, E. Spedicato editores, *Algorithms for Sparse Large Scale Linear Algebraic Systems. Applications in Science and Engineering, 151-174*, Kluwer Academic Publishers, Dordrecht, 1997.

[98] A. SUÁREZ, G. MONTERO, D. GARCÍA Y E. FLÓREZ, Efecto de la Elección del Vector Inicial en la Convergencia de los Algoritmos CGS y BICGSTAB, F. Cobos, J. Gómez, F. Mateos editores, *Encuentro de Análisis Matricial y Aplicaciones*, 210-217, Sevilla, 1997.

[99] R. SUDA, Large scale circuite analysis by preconditioned relaxation methods, in: *Proceedings of PCG'94*, Keio University 189-205, 1994.

[100] R. SUDA, *New iterative linear solvers for parallel circuit simulation*, Department of Information Sciences, University of Tokyo, 1996.

[101] K. TAKAHISHI, J. FAGAN Y M.S. CHEN, Formation of a sparse bus impedance matrix and its application to short circuit study, *Proc. 8th PICA Conference*, Minneapolis, Minnesota, 63-69, 1973.

[102] W. TANG, Toward an effective sparse approximate inverse preconditioner, *SIAM J. Matrix Anal. Appl.*, **20**, 4, 970-986, 1999.

[103] H.A. VAN DER VORST, A vectorizable variant of some ICCG methods, *SIAM J. Sci. Statist. Comput.*, **3**, 350-356, 1982.

[104] H.A. VAN DER VORST, Bi-CGSTAB: A Fast and Smoothly Converging Variant of Bi-CG for the Solution of Nonsymmetric Linear Systems, *SIAM J. Sci. Comput.*, **13**, 631-644, 1992.

[105] C. WEI Y Z. ZHANG, The hadamard product of a psoitive reciprocal matrix and some results in AHP, *Math. Comp. Mod.*, **28**, 10, 59-71, 1998.

[106] H. WEYL, Inequalities between the two kinds of eigenvalues of a linear transformation, *Proc. Nat. Acad. Sci.*, **35**, 408-411, 1949.

[107] A. YU. YEREMIN, L. YU. KOLOTILINA Y A.A. NIKISHIN, Fatorized sparse approximate inverse preconditionings III. Iterative construction of preconditioners, Research Report 11/97, Center for Supercomputing and Massively Parallel Applications, Computer Center of Russian Academy of Sciences, 1997.