Universidad de Valladolid

Trabajo Fin de Grado

Grado en Matemáticas

Métodos de Krylov para sistemas lineales dispersos de ecuaciones.

El teorema de Faber-Manteuffel

Autor:

Ana Méndez Pérez

Tutor:

Dr. Luis M. Abia LLera

Índice general

Resumen

El trabajo consta de cuatro capítulos, cuyo contenido pasamos a describir:

1. El primer capítulo motiva los métodos iterativos para la resolución de sistemas lineales, describiendo con carácter general la metodología de los métodos que en cada iteración determinan la aproximación a la solución mediante proyección sobre un subespacio apropiado. En los métodos de Krylov estos subespacios son precisamente subespacios de Krylov. La construcción de bases ortogonales de estos subespacios es el objeto del algoritmo de Arnoldi, para matrices no simétricas, y el algoritmo simétrico de Lanczos cuando la matriz del sistema es simétrica, para los que se dan las correspondientes interpretaciones matriciales.

2. Aunque en el Grado se estudia el método gradiente conjugado para sistemas con matriz simétrica y definida positiva, se recupera en este capítulo su formulación como un método de proyección sobre un subespacio apropiado de Krylov, probando el teorema de convergencia. La parte más novedosa del capítulo aborda la descripción de otros métodos de Krylov, para sistemas lineales con matrices simétricas pero no definidas positivas o sistemas con matrices simplemente no simétricas: FOM (ortogonalización completa), GMRES (residuos mínimos generalizados), GMRES(m) (residuos mínimos generalizados con truncación) y MINRES (residuos mínimos). Estos métodos se implementarán en rutinas MATLAB propias en la experimentación numérica que ilustra los mismos en el capítulo 4. Las limitaciones de tiempo y espacio dejan fuera de cobertura otros métodos de Krylov de interés práctico: Bi-CG (gradientes biconjugados), Bi-CGSTAB (gradientes biconjugados estabilizados), etc., pero los métodos presentados ilustran ya el gran costo que supone el no disponer de recurrencias cortas para la construcción de los sucesivos iterantes, y que motiva el problema teórico que abordamos en el siguiente capítulo.

3. En 1981 Gene Golub planteó la cuestión de si sería posible para sis-

temas lineales con matrices no simétricas construir un método que en cada iteración optimizara sobre un espacio de Krylov con respecto a una norma proveniente de un producto interno y que requiriera únicamente recurrencias de tres términos, tal y como ocurre en el método gradiente conjugado. Faber y Manteuffel caracterizaron las matrices para las que esto es posible, respondiendo negativamente a la cuestión salvo para situaciones que ya eran bien conocidas: o bien el polinomio mínimo de la matriz es cúbico, o la matriz es hermítica o la matriz adjunta de la matriz del sistema (matriz traspuesta conjugada) se expresa como un polinomio lineal de la propia matriz. La formulación y prueba del teorema de Faber y Manteuffel es el objetivo del tercer capítulo. La prueba necesita de bastantes resultados técnicos de la teoría de matrices, de algunos de los cuales hacemos uso en base a su estudio a lo largo del Grado: por ejemplo, polinomio mínimos de un vector y de una matriz, propiedades generales de las matrices normales, producto exterior de vectores, etc.

4. El cuarto capítulo ilustra algunos experimentos numéricos con los métodos gradiente conjugado, MINRES y GMRES sobre problemas test con matrices grandes y dispersas que se toman de la colección Harwell-Boeing en el Matrix Market, un depósito de matrices test para la comparación de algoritmos de álgebra lineal. Como en algunos de los experimentos fué necesario preacondicionar previamente las matrices, se describe este proceso de forma somera cuando se implementa utilizando la factorización LU o de Cholesky incompletas de la matriz del sistema.

Capítulo 1

Métodos de proyección

La solución numérica de sistemas lineales $Ax = b$ siempre ha sido un área de interés en el campo de las matemáticas debido a la diversidad de sus aplicaciones en distintos ámbitos. Cuando aumenta la dimensión de la matriz A, como es el caso de los grandes sistemas lineales dispersos que surgen en la discretización de ecuaciones en derivadas parciales, la resolución efectiva del sistema se complica ya que los métodos directos derivados de la eliminación Gaussiana no son eficientes ni en términos de costo computacional ni en términos de las necesidades de almacenamiento en la memoria. Es entonces cuando se considera el uso de métodos iterativos para obtener una aproximación a la solución del sistema que se considera. Concretamente, los métodos de Krylov aparecen como una de las técnicas iterativas más importantes para resolver estos sistemas. Estos métodos están basados en procesos de proyección (tanto ortogonales como oblicuos) sobre un tipo de subespacios que se llaman de Krylov, de los que toman su nombre. Este capítulo sirve de introducción a la estructura general de esta clase de métodos.

1.1. Introducción

Buscamos una solución del sistema lineal

$$Ax = b \tag{1.1}$$

con A y b dados, una matriz $n \times n$ no singular y un vector de $\mathbb{R}$ respectivamente. Estas hipótesis se asumirán implícitamente a lo largo de toda la memoria. Las solución puede ser hallada a través de un método directo, que permite la aproximación a la solución exacta del sistema en un numero finito de pasos (hasta la precisión de máquina), o mediante un método iterativo.

Sin embargo, cuando la matriz sea de grandes dimensiones y dispersa (sobre todo si tiene una estructura concreta de sus elementos no nulos -sparsity pattern-) la mejor opción suele ser un método iterativo, en el que se genera una sucesión de vectores que convergen a la solución. Para estos sistemas los métodos directos pueden no ser adecuados por las necesidades de memoria inducidas por la destrucción de los elementos que son inicialmente nulos en la matriz A debido al proceso de factorización.

1.1.1. Métodos iterativos clásicos

Los métodos iterativos no modifican la estructura de la matriz A, en particular su dispersión. En ellos se parte de una aproximación inicial x_0 a la solución del sistema $Ax = b$, con $det(A) \neq 0$, y se construye una sucesión de vectores $\{x_k\}_{k\geq 0}$ que bajo adecuadas hipótesis convergen a la solución exacta del sistema (hasta la precisión de la máquina). Es decir:

$$\lim_{k\to\infty} x_k = x \quad \text{con} \quad x = A^{-1}b$$

El proceso de construcción de los iterantes x_k, $k = 0, \ldots$, involucra la solución de un sistema lineal mucho más fácil que el original.

En los métodos clásicos la matriz A se descompone como suma de dos matrices

$$A = D + C$$

con D no singular, para reescribir el sistema lineal

$$(D + C)x = b \Longleftrightarrow Dx = -Cx + b$$

Esta ecuación sugiere la iteración

$$Dx_{k+1} = -Cx_k + b \qquad k = 0, 1, ...$$

que implica la resolución en cada iteración de un sistema lineal con matriz D (para la que se supone que la resolución de sistemas lineales con matriz D es simple).

Para el análisis de convergencia, ponemos

$$x_{k+1} = -D^{-1}Cx_k + D^{-1}b \qquad k = 0, 1, ...$$

y llamamos a la matriz $B := D^{-1}C$ matriz de iteración del método. Si restamos de esta ecuación la que satisface la solución

$$x = D^{-1}Cx + D^{-1}b$$

obtenemos para los errores sucesivos $e_k = x - x_k$, $k = 0, \ldots,$ la recurrencia

$$e_{k+1} = Be_k = B^2 e_{k-1} = ... = B^{(k+1)} e_0$$

La convergencia a cero de los errores $\{e_k\}_{k\geq 0}$, cualquiera que sea el vector error inicial e_0, se caracteriza por la convergencia a la matriz cero de la sucesión de potencias $\{B^k\}_{k\geq 1}$ de la matriz de iteración. El siguiente teorema caracteriza esta convergencia

Teorema. *Sea $\rho(B)$ el radio espectral de B, entonces*

$$\lim_{k\to\infty} B^k = 0 \quad \Longleftrightarrow \quad \rho(B) < 1$$

donde $\rho(B)$ denota el radio espectral de la matriz B.

La desigualdad $\rho(B) \leq \|B\|$, siendo $\|\cdot\|$ cualquier norma matricial natural, implica que $\|B\| < 1$ es una condición suficiente para la convergencia del método iterativo.

El valor del radio espectral no sólo nos indica la convergencia o divergencia del método, sino también la velocidad de convergencia: el método iterativo converge más rápido cuánto más cercano sea el radio espectral a 0 y viceversa, más lento cuanto más cerca este el radio espectral del valor 1.

En el método de Jacobi, la matriz D es la parte diagonal de la matriz A del sistema (se supone, naturalmente, que todos los elementos diagonales de A son no nulos). En el método de Gauss-Seidel la matriz D es la parte triangular inferior de la matriz A. El método SOR pertenece también a esta clase de métodos iterativos basados en escisiones de la matriz A. La memoria no va a tratar estos métodos, que sólo mencionamos a titulo informativo.

1.1.2. Métodos de proyección

La mayoría de técnicas iterativas para resolver sistemas lineales de grandes dimensiones utilizan un proceso de proyección en una forma u otra.

Consideramos el sistema lineal (1.1) donde A es una matriz real $n \times n$, regular. La idea detrás de un método de proyección es extraer una solución aproximada de (1.1) desde un subespacio de $\mathbb{R}^n$. Sea $\mathcal{K}$ dicho subespacio, llamado subespacio de aproximaciones candidatas o subespacio de búsqueda y sea i su dimensión. Un elemento de este subespacio se determina con i condiciones lineales e independientes, proporcionando la aproximación deseada. Una manera típica de describir estas condiciones es imponer i relaciones de

ortogonalidad independientes. Concretamente, se impone que el vector residuo $r = b - Ax$ sea ortogonal a i vectores linealmente independientes. Estos vectores definen otro subespacio $\mathcal{L}$ de dimensión i llamado subespacio de restricciones. Este marco teórico es utilizado en muchas técnicas matemáticas de aproximación bajo la denominación de método de Petrov-Galerkin. Es decir, un método de proyección dentro del subespacio $\mathcal{K}$ y ortogonal a $\mathcal{L}$ es un proceso que encuentra una solución aproximada $\tilde{x}$ a (1.1) imponiendo:

$$\tilde{x} \in \mathcal{K} \quad \text{tal que} \quad b - A\tilde{x} \perp \mathcal{L} \tag{1.2}$$

Para explotar el conocimiento de una aproximación inicial x_0 de la solución, la aproximación se determina en el espacio afín $x_0 + \mathcal{K}$ en vez de en el espacio $\mathcal{K}$.

$$\tilde{x} \in x_0 + \mathcal{K} \quad \textit{tal que} \quad b - A\tilde{x} \perp \mathcal{L} \tag{1.3}$$

Escribiendo $\tilde{x} = x_0 + \delta$ y el vector residual inicial $r_0 = b - Ax_0$ entonces

$$b - A(x_0 + \delta) = r_0 - A\delta \perp \mathcal{L}$$

En otras palabras la solución de un método de proyección puede ser definida por:

$$\tilde{x} = x_0 + \delta, \quad \delta \in \mathcal{K} \tag{1.4}$$

$$(r_0 - A\delta, w) = 0, \forall w \in \mathcal{L} \tag{1.5}$$

Hay dos clases de métodos de proyección: ortogonales y oblicuos. En las técnicas de proyección ortogonal $\mathcal{L} = \mathcal{K}$ mientras que en las técnicas de proyección oblicua $\mathcal{L} \neq \mathcal{K}$, y puede que no tengan ninguna relación entre sí.

A continuación abordamos una representación matricial para este tipo de proyecciones lineales. Sea $V = [v_1, ..., v_i]$ y $W = [w_1, ..., w_i]$ matrices cuyos vectores columna forman una base de $\mathcal{K}$ y $\mathcal{L}$ respectivamente. Si la solución aproximada es escrita como $\tilde{x} = x_0 + Vy$, entonces la condición de ortogonalidad deja el siguiente sistema de ecuaciones para el vector y:

$$W^T AVy = W^T r_0 \tag{1.6}$$

Si la matriz $W^T AV$ es no singular, el vector y está perfectamente definido y obtenemos una expresión para la aproximación de la solución $\tilde{x}$:

$$\tilde{x} = x_0 + V(W^T AV)^{-1} W^T r_0 \tag{1.7}$$

Lema 1. *Sea A no singular, suponemos que las matrices V y W son como arriba. Entonces, $W^T AV$ es singular si y sólo si existe $v \in A\mathcal{K} \setminus \{0\}$ tal que $v \perp \mathcal{L}$, siendo $A\mathcal{K} = \{AVy \;/\; y \in \mathbb{R}^i\}$*

Demostración. $\Leftarrow$) Supongamos que existe $v \in A\mathcal{K}$, no nulo, tal que v es ortogonal a $\mathcal{L}$. Por tanto, existe $y \in \mathbb{R}^i$, $y \neq 0$, tal que $v = AVy$ y $W^TAVy = 0$, por tanto W^TAV es singular.

$\Rightarrow$) Por hipótesis, existe $y \in \mathbb{R}^i$, $y \neq 0$, tal que $W^TAVy = 0$. Además, $AVy \neq 0$ pues $\{v_1, v_2, ..., v_i\}$ son linealmente independientes y A es regular. Por tanto, si ponemos $v = AVy$ tenemos que $W^Tv = 0$, como se quería demostrar. □

Como consecuencia del lema previo sabemos que el método de proyección sobre $\mathcal{K}$ ortogonal a $\mathcal{L}$ tiene solución única si y solo si $A\mathcal{K} \cap \mathcal{L}^\perp = 0$, donde $\mathcal{L}^\perp$ denota el complemento ortogonal de $\mathcal{L}$.

Hay dos casos importantes en los que está garantizada la no singularidad de W^TAV:

Proposición 1. *Sean A, $\mathcal{L}$ y $\mathcal{K}$ satisfaciendo una de las siguientes condiciones,*

1. *A es positiva definida y $\mathcal{L} = \mathcal{K}$,*
2. *A es no singular y $\mathcal{L} = A\mathcal{K}$*

Entonces la matriz $B = W^TAV$ es no singular.

Demostración. Consideramos el primer caso. Por hipótesis tenemos que $\mathcal{L}$ y $\mathcal{K}$ son iguales, y la matriz $W = [w_1, ..., w_i]$ cuyas columnas son la base de $\mathcal{L}$ puede ser expresada como $W = VG$, donde G es una matriz no singular $i \times i$. Entonces

$$B = W^TAV = G^TV^TAV \tag{1.8}$$

Como A es positiva definida, también lo es V^TAV y por tanto B es no singular.

Consideramos ahora el segundo caso. Ya que $\mathcal{L} = A\mathcal{K}$ en este caso W puede ser expresado como $W = AVG$, donde G es una matriz no singular $i \times i$. Entonces

$$B = W^TAV = G^T(AV)^TAV \tag{1.9}$$

Como A es no singular, la matriz $n \times i$, AV, tiene rango máximo y como consecuencia $(AV)^TAV$ es no singular, por tanto por (1.9) la matriz B es no singular. □

1.2. Métodos de Krylov

Recordamos del apartado anterior que un método de proyección para resolver el sistema lineal $Ax = b$ es un procedimiento para obtener una solución aproximada x_i dentro del subespacio afín $x_0+\mathcal{K}_i$, de dimensión i, imponiendo la condición Petrov-Galerkin

$$b - Ax_i \perp \mathcal{L}_i$$

donde $\mathcal{L}_i$ es otro espacio de dimensión i.

Un método de Krylov es un método para el cuál el subespacio $\mathcal{K}_i$ es el subespacio de Krylov

$$\mathcal{K}_i(A, r_0) = span\{r_0, Ar_0, A^2r_0, ..., A^{i-1}r_0\}$$

donde $r_0 = b - Ax_0$. Denotaremos con $\mathcal{K}_i$ al subespacio $\mathcal{K}_i(A, r_0)$ cuando no exista ambigüedad. Diferentes métodos de Krylov difieren en la elección del subespacio $\mathcal{L}_i$ y en la forma en que el sistema se preacondiciona, como describiremos más adelante.

La aproximación obtenida desde un método de Krylov es de la forma

$$A^{-1}b \approx x_i = x_0 + q_{i-1}(A)r_0,$$

donde q_{i-1} es un cierto polinomio de grado $i - 1$.De esto sigue que

$$r_i = b - Ax_i = (I - Aq_{i-1}(A))r_0 = \tilde{p}_{i-1}(A)r_0 \tag{1.10}$$

con $\tilde{p}_i(0) = 1$.

Veamos un razonamiento breve que nos da una idea de por qué estos métodos basados en dichos subespacios cobran tanta importancia.

Denotemos a la solución del sistema por $x^* = A^{-1}b$. Si x_0 es una aproximación inicial a x^*, entonces resolver $Ax = b$ equivale a resolver $Az = r_0$, con $z := x^* - x_0$. Así, tenemos también $z = A^{-1}r_0$. Sea ahora $p(x)$ el polinomio mónico de menor grado γ tal que $p(A)r_0 = 0$, que será de la forma $p(x) = \alpha_0 + \alpha_1 x + ... + \alpha_{\gamma-1}x^{\gamma-1} + x^\gamma$, con $\alpha_0 \neq 0$. Se debe observar que $\gamma \leq n$ pues por el teorema de Caley-Hamilton toda matriz es anulada por su polinomio característico. Por tanto,

$$A^{-1}r_0 = \frac{-1}{\alpha_0}[\alpha_1 I + \alpha_2 A + ... + \alpha_{\gamma-1}A^{\gamma-2} + A^{\gamma-1}]r_0,$$

y entonces

$$z \in span\{r_0, Ar_0, A^2r_0, ..., A^{\gamma-1}r_0\}$$

o bien

$$x^* \in x_0 + span\{r_0, Ar_0, A^2r_0, ..., A^{\gamma-1}r_0\}.$$

Así llegamos a que la solución x^* del sistema $Ax = b$ se encuentra necesariamente en el espacio afín $x_0 + span\{r_0, Ar_0, A^2r_0, ..., A^{\gamma-1}r_0\} = x_0 + \mathcal{K}_\gamma(A, r_0)$. El espacio de Krylov $\mathcal{K}_\gamma(A, r_0)$ aparece de forma natural como el espacio de direcciones de la variedad afín que contiene a la solución del sistema.

Abordamos ahora algunas propiedades generales de estos espacios de Krylov

$$K_i(A, v) = span\{v, Av, A^2v, ..., A^{i-1}v\}. \tag{1.11}$$

La primera propiedad es que K_i es el subespacio de todos los vectores $x \in \mathbb{R}^n$ que pueden ser escritos de la forma $x = p(A)v$ donde p es un polinomio de grado $\leq i - 1$. Recordamos que el polinomio mínimo de un vector v es el polinomio mónico de menor grado $p(x)$, distinto de cero, tal que $p(A)v = 0$. Al grado de este polinomio lo llamamos grado de v con respecto a A. Por el teorema de Cayley-Hamilton, el grado de v con respecto a A no excede del orden de la matriz A.

Enunciamos sin prueba las siguientes propiedades:

Proposición 2. *$\mathcal{K}_i(A, v)$ es el subespacio de todos los vectores $x \in \mathbb{R}^n$ que pueden ser escritos en la forma $x = p(A)v$, donde p es un polinomio de grado $\leq i - 1$.*

Proposición 3. *Sea μ el grado de v con respecto a A. Entonces $\mathcal{K}_i = \mathcal{K}_i(A, v)$ es invariante por A y $K_i = K_\mu$ para todo $i \geq \mu$.*

Proposición 4. *El subespacio de Krylov $\mathcal{K}_i = \mathcal{K}_i(A, v)$ es de dimensión i si y sólo si el grado μ de v con respecto a A no es menor que i, es decir,*

$$dim(\mathcal{K}_i) = i \quad \Leftrightarrow \quad grado(v) \geq i$$

Por tanto,

$$dim(\mathcal{K}_i) = min\{i, grado(v)\}$$

Demostración. Los vectores $v, Av, ..., A^{i-1}v$ forman una base de $\mathcal{K}_i$ si y sólo si para cualquier conjunto de i escalares α_j, $j = 0, ..., i - 1$, dónde al menos uno de los α_j es distinto de cero, la combinación lineal $\sum_{j=0}^{i-1} \alpha_j A^j v$ es distinto de cero. Esto es equivalente a que el único polinomio de grado $\leq i - 1$ para el cual $p(A)v = 0$ es el polinomio cero. La segunda parte es consecuencia de la proposición previa. □

1.2.1. Algoritmo de Arnoldi

Dado un espacio de Krylov $\mathcal{K}_i(A, v) = span\{v, Av, \ldots, A^{i-1}v\}$, es de interés construir una base ortonormal de dicho espacio. El algoritmo de Arnoldi no es más que una implementación del proceso de ortonormalización de Gram-Schmidt aplicado a dicho subespacio. En aritmética exacta, una variante del algoritmo es la dada por

Algoritmo de Arnoldi:

```
1  Elegimos v_1 = v/‖v‖, de norma 1
2  for j = 1,2,...,i
3        Computamos h_ℓj = (Av_j, v_ℓ) para ℓ = 1,2,...,j
4        w_j := Av_j − Σ_{ℓ=1}^{j} h_ℓj v_ℓ
5        h_{j+1,j} = ‖w_j‖_2
6        if h_{j+1,j} = 0 then Stop
7        end
8        v_{j+1} = w_j/h_{j+1,j}
9  end
```

En cada iteración el algoritmo multiplica el vector previo v_j por A y mediante Gram-Schmidt lo ortonormaliza con respecto a todos los vectores anteriores v_ℓ. El algoritmo parará si el vector w_j de la línea 4 es cero.

Originalmente, el algoritmo de Arnoldi (1951) fue diseñado para reducir matrices densas no hermíticas a forma Hessenberg superior.

Proposición 5. *Suponemos que el algoritmo no para antes de la i-ésima iteración. Entonces los vectores $v_1, ..., v_i$ forman una base ortonormal del subespacio de Krylov*

$$\mathcal{K}_i = span\{v_1, Av_1, ..., A^{i-1}v_1\}$$

Demostración. Los vectores v_j, $j = 1, \ldots, i$, son ortonormales por construcción.

Para ver que estos vectores generan $\mathcal{K}_i$ veamos que cada vector v_j es de la forma $q_{j-1}(A)v_1$ donde q_{j-1} es un polinomio de grado $j-1$. Probemos esto por inducción.

El resultado es claro para $j = 1$ ya que $v_1 = q_0(A)v_1$ con $q_0(t) \equiv 1$. Asumimos que el resultado es verdad para todo entero $\leq j$ y consideramos v_{j+1}. Tenemos

$$h_{j+1}v_{j+1} = Av_j - \sum_{\ell=1}^{j} h_{\ell j}v_\ell = Aq_{j-1}(A)v_1 - \sum_{\ell=1}^{j} h_{\ell j}q_{\ell-1}(A)v_1$$

que demuestra que v_{j+1} puede ser expresado como $q_j(A)v_1$ donde q_j es de grado j y esto finaliza la prueba. □

La siguiente proposición reformula en forma matricial el resultado del algoritmo de Arnoldi:

Proposición 6. *Denotemos por $V_i = [v_1, ..., v_i]$ la matriz $n \times i$ con dichos vectores columna, por $H_{i+1,i}$ la matriz Hessenberg $(i+1) \times i$ cuyos elementos no nulos son los $h_{\ell j}$ definidos por el algoritmo de Arnoldi y H_i la matriz obtenida al borrar la ultima fila de $H_{i+1,i}$. Entonces se verifican las siguientes relaciones:*

$$AV_i = V_i H_i + w_i e_i^T = V_{i+1} H_{i+1,i} \tag{1.12}$$

$$V_i^T A V_i = H_i \tag{1.13}$$

Demostración. La relación (1.12) sigue de la siguiente igualdad, la cual se deriva de las líneas 4, 5 y 7 del algoritmo de Arnoldi.

$$Av_j = w_j + \sum_{\ell=1}^{j} h_{\ell j} v_i = v_{j+1} h_{j+1,j} + \sum_{\ell=1}^{j} h_{\ell j} v_\ell = \sum_{\ell=1}^{j+1} h_{\ell j} v_\ell \ , \ j = 1, 2, ..., i.$$

La relación (1.13) viene de multiplicar esta igualdad a ambos lados por V_i^T y haciendo uso de la ortonormalidad de $\{v_1, v_2, ..., v_i\}$. □

Como ya dijimos anteriormente el algoritmo para en la iteración j cuando la norma de w_j se anula. Todavía no hemos determinado las condiciones bajo las que ocurre esta situación

Proposición 7. *El algoritmo de Arnoldi para en la iteración j (es decir, $h_{j+1,j} = 0$ en la línea 5 del algoritmo), si y sólo si el polinomio minimal de v_1 con respecto a A es de grado j. Además en este caso el subespacio $\mathcal{K}_j$ es invariante bajo A.*

Demostración. $\Leftarrow$) Si el grado del polinomio minimal es j, entonces w_j debe ser 0. De hecho, si no fuese así v_{j+1} podría ser definido y resultaría que $\mathcal{K}_{j+1}$ sería de dimensión $j+1$. Entonces por la proposición 4 esto implicaría que el grado μ del polinomio de v_1 es $\geq j+1$, lo cual es una contradicción.

$\Rightarrow$) Para probar el sentido inverso asumimos que $w_j = 0$. Entonces el grado μ del polinomio minimal de v_1 con respecto a A es $\mu \leq j$. Además es imposible que $\mu < j$ ya que por la primera parte de la prueba el vector w_μ sería cero y el algoritmo habría parado con antelación a la iteración μ. El resto del resultado sigue de la proposición 3 □

Un corolario de la proposición anterior es que un método de proyección sobre el subespacio $\mathcal{K}_j$ será exacto cuando el algoritmo pare en la iteración j por ser $h_{j+1,j} = 0$.

El algoritmo de Gram-Schmidt modificado es una reordenación de los cálculos en el algoritmo clásico de Gram-Schmidt. En aritmética exacta el algoritmo de Arnoldi básico y el algoritmo de Arnoldi modificado son equivalentes. En aritmética coma flotante la versión modificada es más fiable porque resulta en una menor pérdida de la ortogonalidad de los vectores que se computan. Existen casos, donde la cancelación numérica es tan severa en el paso de la ortogonalización de cada vector que incluso el algoritmo de Arnoldi modificado resulta inadecuado. En estas situaciones una doble ortogonalización o un proceso de ortogonalización mediante reflectores de Householder pueden salvar estos inconvenientes. La versión del algoritmo de Arnoldi modificado se detalla a continuación.

Algoritmo de Arnoldi (Gram-Schmidt modificado):

```
1  Elegimos v_1 = v/‖v‖, de norma 1
2  for j = 1, ..., i
3          w_j = A v_j
4          for ℓ = 1, ..., j
5             h_ℓj = (w_j, v_ℓ)
6             w_j = w_j − h_ℓj v_ℓ
7          end
8          h_{j+1,j} = ‖w_j‖_2
9          if h_{j+1,j} = 0
10                    stop
11         end
12         v_{j+1} = w/h_{j+1,j}
13 end
```

1.2.2. Algoritmo simétrico de Lanczos

El algoritmo simétrico de Lanczos puede ser visto como una simplificación del método de Arnoldi para el caso particular en que la matriz es simétrica. Cuando A es simétrica entonces la matriz Hessenberg H_i es tridiagonal y simétrica, lo cuál deja relaciones de recurrencia de tres términos en cada paso de ortogonalización del proceso de Arnoldi.

Para introducir el algoritmo de Lanczos empezamos con la observación establecida en el siguiente teorema.

Teorema. *Suponemos que el método de Arnoldi es aplicado a una matriz real simétrica A. Entonces los coeficientes $h_{\ell,j}$ generados por el algoritmo*

son de la siguiente forma

$$h_{\ell j} = 0, \quad para \quad 1 \leq \ell < j-1, \tag{1.14}$$

$$h_{j,j+1} = h_{j+1,j}, \quad j = 1, 2, \cdots, i. \tag{1.15}$$

En otras palabras, la matriz tridiagonal H_i obtenida por el proceso de Arnoldi es tridiagonal y simétrica.

Demostración. La demostración es una consecuencia inmediata del hecho de que $H_i = V_i^T A V_i$ es una matriz simétrica, que por construcción también es Hessenberg. Por tanto H_i debe ser una matriz simétrica tridiagonal. □

Denotamos por T_i a la matriz tridiagonal y simétrica H_i, que asignando

$$\alpha_j \equiv h_{j,j}, \quad \beta_j \equiv h_{j-1,j}$$

queda de la forma:

$$T_i = \begin{pmatrix} \alpha_1 & \beta_2 & & & \\ \beta_2 & \alpha_2 & \beta_3 & & \\ & . & . & . & \\ & & \beta_{i-1} & \alpha_{i-1} & \beta_i \\ & & & \beta_i & \alpha_i \end{pmatrix} \tag{1.16}$$

Esto nos deja la siguiente variante del Algoritmo de Arnoldi con Gram-Schmidt modificado.

Algoritmo de Lanczos:

```
1  Elegimos v_1 = v/||v||, de norma 1. Sean β_1 = 0, v_0 = 0
2  for j = 1, ..., i
3          w_j = A v_j − β_j v_{j−1}
4          α_j = (w_j, v_j)
5          w_j = w_j − α_j v_j
6          β_{j+1} = ||w||_2
7          if β_{j+1} = 0 then Stop
8          end
9          v_{j+1} = w/β_{j+1}
10 end
```

La idea principal de este algoritmo es una relación de recurrencia de tres términos:

$$\beta_{j+1} v_{j+1} = A v_j - \alpha_j v_j - \beta_{j-1} v_{j-1} \tag{1.17}$$

Esta relación recuerda a la relación de recurrencia de tres términos de los polinomios ortogonales. De hecho hay una relación estrecha entre el algoritmo

de Lanczos y los polinomios ortogonales. Si el grado de v_1 con respecto a A es $\geq i$ entonces la dimension del subespacio $\mathcal{K}_i$ es i y está formado por todos los vectores de la forma $q(A)v_1$ donde $grado(q) \leq i-1$. De hecho existe un isomorfismo entre $\mathcal{K}_i$ y el espacio de los polinomios de grado $\leq i-1$, llamemoslo $\mathbb{P}_{i-1}$, el cual está definido por:

$$q \in \mathbb{P}_{i-1} \rightarrow x = q(A)v_1 \in \mathcal{K}_i$$

Consideramos ahora el siguiente producto interno en el subespacio $\mathbb{P}_{i-1}$:

$$\langle p, q \rangle_{v_1} = (p(A)v_1, q(A)v_1) \tag{1.18}$$

Ya que los vectores v_j son de la forma

$$v_j = q_{j-1}(A)v_1 \tag{1.19}$$

la ortogonalidad de los v_j se transforma en ortogonalidad de los polinomios con respecto al producto interno (1.18).

Por tanto el algoritmo de Lanczos no es más que el algoritmo de Sietjes, (ver por ejemplo [1]) para calcular una sucesión de polinomios ortogonales respecto a un producto interno (1.18). Sabemos [5] que el polinomio característico de la matriz tridiagonal generada por el algoritmo de Lanczos minimiza la norma $\|\cdot\|_{v_1}$ sobre los polinomios mónicos. La relación de recurrencia entre los polinomios característicos de matrices diagonales también muestra que el algoritmo de Lanczos calcula la secuencia de vectores $p_{T_i}(A)v_1$ donde p_{T_i} es el polinomio característico de T_i. Precisamente el teorema de Faber-Manteuffel, que estudiaremos en el capítulo 3 busca identificar para qué matrices se produce este tipo de relaciones de recurrencia.

Capítulo 2

Métodos iterativos de Krylov

2.1. Clasificación de algoritmos

2.1.1. Aproximación Ritz-Galerkin

En la clase de métodos Ritz-Galerkin la solución aproximada x_i se determina imponiendo las condiciones de ortogonalidad $r_i \perp \mathcal{K}_i(A; r_0)$, esto es equivalente a

$$V_i^T(b - Ax_i) = 0 \tag{2.1}$$

donde recordamos que $V_i = [v_1, \ldots, v_i]$ denota la matriz cuyas columnas forman una base del subespacio de Krylov $\mathcal{K}_i$.

Si $x_0 = 0$, $b = r_0 = \|r_0\|_2 v_1$ y de esto se deriva que $V_i^T b = \|r_0\|_2 e_1$ con e_1 el vector unitario de la base canónica de $\mathbb{R}^n$.

Como $x_i = V_i y_i$, ya que debe pertenecer al subespacio de Krylov K_i, sustituyendo todo esto en la ecuación (2.1) obtenemos

$$V_i^T A V_i y_i = \|r_0\|_2 e_1 \tag{2.2}$$

Este sistema puede ser interpretado como el sistema $Ax = b$ proyectado dentro del subespacio de Krylov $\mathcal{K}_i(A; r_0)$.

Obviamente tenemos que construir la matriz $i \times i$, $V_i^T A V_i$, pero como hemos visto del proceso de ortogonalización (1.13) $V_i^T A V_i = H_{i,i}$, así que la solución aproximada x_i para la cual $r_i \perp \mathcal{K}_i(A; r_0)$ puede calcularse fácilmente resolviendo

$$\begin{cases} H_{i,i} y_i = \|r_0\|_2 e_1 \\ x_i = V_i y_i \end{cases} \tag{2.3}$$

Este algoritmo es conocido como FOM (método de ortogonalización completa).

Cuando A es simétrica la matriz $H_{i,i}$ se reduce a la matriz tridiagonal $T_{i,i}$ y resulta el método Lanczos descrito anteriormente en el apartado 1.2.2.

Cuando A además es definida positiva obtenemos el método del Gradiente Conjugado. En las implementaciones comunmente usadas de este método, se realiza implícitamente una factorización LU de la matriz $T_{i,i}$ sin generar explícitamente $T_{i,i}$, lo cual deja recurrencias elegantes de tres términos para x_i y el correspondiente residuo r_i.

Que A sea definida positiva es suficiente para garantizar la existencia de la factorización LU, pero también nos da otra interpretación útil. Del hecho de que $r_i \perp \mathcal{K}_i(A; r_0)$ obtenemos que $A(x_i - x) \perp \mathcal{K}_i(A; r_0)$ o $x_i - x \perp_A \mathcal{K}_i(A; r_0)$. La última observación muestra que el error es A-ortogonal al subespacio de Krylov y esto es equivalente a que $\|x_i - x\|_A$ es minimal. [1]

2.1.2. Aproximación mínima norma residual

En estos métodos x_i se determina minimizando la norma euclídea $\|b - Ax_i\|$ del residuo en el espacio de Krylov $\mathcal{K}_i(A; r_0)$. La creación de una base ortogonal del subespacio de Krylov $\{v_1, ..., v_{i+1}\}$ deja las ecuaciones

$$V_i^T A V_i = H_{ii} \tag{2.4}$$

$$AV_i = V_{i+1} H_{i+1,i} \tag{2.5}$$

De nuevo buscamos una aproximación $x_i \in K_i(A; r_0)$, por tanto escogemos $x_i = V_i y_i$ tal que minimice la norma residual $\|b - Ax_i\|_2$

Esta norma puede ser reescrita con $b = r_0$, $v_1 = r_0/\|r_0\|$, y poniendo $\beta = \|r_0\|_2$, como

$$\|b - Ax_i\|_2 = \|b - AV_i y_i\|_2 = \|\beta V_{i+1} e_1 - V_{i+1} H_{i+1,i} y_i\|_2 \tag{2.6}$$

Ya que V_{i+1} es ortonormal

$$\|b - Ax_i\|_2 = \|\beta e_1 - H_{i+1,i} y_i\|_2 \tag{2.7}$$

y esta norma puede ser minimizada resolviendo el correspondiente problema de mínimos cuadrados. Para ello se construye la factorización QR de la matriz $H_{i+1,i}$ que como es una matriz superior de Hessenberg puede ser realizada mediante rotaciones de Givens. El método GMRES se basa en esta estrategia.

[1] La A-norma está definida por $\|y\|_A^2 \equiv (y, y)_A \equiv (y, Ay)$ y necesitamos que A sea definida positiva para poder conseguir un producto interno adecuado $(\cdot, \cdot)_A$.

2.1.3. Aproximación Petrov-Galerkin

Adelantándonos al teorema de Faber-Manteuffel, para sistemas lineales con matriz no simétrica no podemos reducir la matriz A a un sistema simétrico en un subespacio de menor dimensión mediante proyecciones ortogonales. La razón es que no podemos crear una base ortogonal para el subespacio de Krylov usando recurrencias de tres términos. Podemos, sin embargo, obtener una base no ortogonal conveniente con una elegante recurrencia de tres términos, forzando que esta base sea ortogonal con respecto a otro subespacio.

Empezamos construyendo una base arbitraria para el subespacio de Krylov:

$$h_{i+1,i}v_{i+1} = Av_i - \sum_{j=1}^{i} h_{j,i}v_j \tag{2.8}$$

que en notación matricial sería $AV_i = V_{i+1}H_{i+1,i}$. Los coeficientes $h_{i+1,i}$ definen la norma de v_{i+1}, y una elección natural sería elegirlos de manera que $\|v_{i+1}\|_2 = 1$. En el método Bi-CG (bigradiente conjugado) es común seleccionar $h_{i+1,i}$ tal que $\|v_{i+1}\|_2 = \|r_{i+1}\|_2$.

Claramente no podemos usar V_i para la proyección, pero supongamos que tenemos una matriz $W_i = [w_1, \ldots, w_i]$ tal que $W_i^T V_i = D_i$ (siendo D_i una matriz diagonal con elementos diagonales d_i) y para la cual $W_i^T v_{i+1} = 0$. Entonces

$$W_i^T A V_i = D_i H_{i,i}$$

y nuestro objetivo es encontrar W_i para la cual $H_{i,i}$ es tridiagonal. Por tanto $V_i^T A^T W_i$ debe ser también tridiagonal. Esto nos sugiere generar los w_i a partir de A^T.

Elegimos un vector arbitrario $w_1 \neq 0$, tal que $w_1^T v_1 \neq 0$. Generamos ahora v_2 con la relación (2.8) y lo ortogonalizamos con respecto a w_1, es decir, $w_2^T v_1 = 0$, que multiplicando el vector w_2 por la izquierda de la relación (2.8) $h_{2,1}v_2 = Av_1 - h_{1,1}v_1$ obtenemos que debe ser $h_{1,1} = (w_1^T A v_1)/(w_1^T v_1)$. Ya que $w_1^T A v_1 = (A^T w_1)^T v_1$ esto implica que w_2, que es generado por

$$h_{2,1}w_2 = A^T w_1 - h_{1,1}w_1,$$

es también ortogonal a v_1. Continuando este razonamiento vemos que podemos crear bases v_i y w_j biortogonales haciendo que los nuevos v_i sean ortogonales a los vectores $w_1, \ldots, w_{i-1}$ y generando w_i con la misma recurrencia, pero reemplazando A por A^T.

De esta manera tenemos $W_i^T A V_i = D_i H_{i,i}$ y $V_i^T A^T W_i = D_i H_{i,i}$. Por tanto $D_i H_{i,i}$ es simétrica y $H_{i,i}$ es una matriz tridiagonal, que nos da la deseada relación de recurrencia de tres términos para los v_j y los w_j.

Varios métodos tales como BI-CG y QMR están basados en estos conjuntos de vectores biortogonales.

2.1.4. Aproximación mínima norma del error

Estos métodos determinan la solución aproximada x_i en $A^T\mathcal{K}_i(A^T; r_0)$ para la cual la norma euclidea $\|x_i - x\|_2$ es minimal. Esta aproximación no es tan obvia como el resto, pero para $A = A^T$ es más natural, resultando en este caso el método SYMMLQ.

Minimizamos la norma del error $x - x_i$, para $x_i = x_0 + AV_i y_i$, lo cual significa que y_i es la solución de las ecuaciones normales

$$V_i^T AAV_i y_i = V_i^T A(x - x_0) = V_K^T r_0 = \|r_0\|_2 e_1 \tag{2.9}$$

Este sistema puede simplificarse utilizando la relación de Lanczos (1.12). En este caso como A es simétrica la escribiremos con $H_{i+1,i} = T_{i+1,i}$.

$$V_i^T AAV_i = T_{i+1,i}^T V_{i+1}^T V_{i+1} T_{i+1,i} = T_{i+1,i}^T T_{i+1,i}$$

Una manera estable de resolver este sistema de ecuaciones normales es basándonos en una descomposición $L\tilde{Q}$ de $T_{i+1,i}^T$, pero notemos que esta es equivalente a la traspuesta de la factorización $Q_{i+1,i}R_i$ de $T_{i+1,i}$, donde R_i es una matriz tridiagonal superior.

$$T_{i+1,i}^T = R_i^T Q_{i+1,i}^T$$

Por tanto esto nos deja la relación (2.9) de la siguiente forma:

$$T_{i+1,i}^T T_{i+1,i} y_i = R_i^T R_i y_i = \|r_0\|_2 e_1,$$

de donde obtenemos la formula básica para SYMMLQ

$$\begin{aligned} x_i &= x_0 + AV_i R_i^{-1} R_i^{-T} \|r_0\|_2 e_1 \\ &= x_0 + V_{i+1} T_{i+1,i} R_i^{-1} R_i^{-T} \|r_0\|_2 e_1 \\ &= x_0 + V_{i+1} Q_{i+1,i} R_i R_i^{-1} R_i^{-T} \|r_0\|_2 e_1 \\ &= x_0 + (V_{i+1} Q_{i+1,i})(L_i^{-1} \|r_0\|_2 e_1) \end{aligned}$$

con $L_i \equiv R_i^T$.

En SYMMLQ es posible generar las aproximaciones Galerkin como un subproducto del proceso. Esto significa que para matrices simétricas definidas positivas los resultados del método gradiente conjugado pueden ser reconstruidos con un coste relativamente bajo a partir de los resultados de SYMMLQ. Esto en matrices simétricas definidas positivas no nos da ventaja pero puede ser usado para matrices simétricas indefinidas, donde la ventaja que obtenemos es que SYMMLQ evita la descomposición LU de $T_{i,i}$ que podría no existir o ser singular en matrices indefinidas.

2.2. Algoritmo del gradiente conjugado

2.2.1. Derivación del método

Como explicamos en el apartado 2.1.1 el método del gradiente conjugado puede ser visto como una variante del método de Lanczos. En esta sección seguimos asumiendo que $x_0 = 0$ sin perdida de generalidad. El método del gradiente conjugado se basa en la relación (1.12)

$$AV_i = V_{i+1}H_{i+1,i}$$

que cuando A es simétrica se reduce a $H_{i+1,i}$ tridiagonal. Para la i-ésima columna de V_i tenemos

$$Av_i = h_{i+1,1}v_{i+1} + h_{i,i}v_i + h_{i-1,i}v_{i-1} \tag{2.10}$$

En la aproximación Ritz-Galerkin el nuevo residuo $b - Ax_i$ es ortogonal al espacio generado por $v_1, ..., v_i$, por lo que r_i está en la dirección de v_{i+1}. Veamos esto más detalladamente:

$$\begin{aligned} b - Ax_i &= b - AV_iy_i = r_0 - AV_iy_i \\ &= \|r_0\|v_1 - V_iH_iy_i - h_{i+1,i}e_i^T y_i v_{i+1} \end{aligned}$$

En la tercera igualdad usamos la relación(1.13). Por la definición de y_i, $H_iy_i = \|r_0\|e_1$ y por consiguiente

$$\|r_0\|v_1 - V_iH_iy_i = \|r_0\|(v_1 - V_ie_1) = \|r_0\|(v_1 - v_1) = 0.$$

Por tanto

$$b - Ax_i = -h_{i+1,i}e_i^T y_i v_{i+1} \tag{2.11}$$

Por tanto podemos seleccionar el factor escalar $h_{i+1,i}$ para que v_{i+1} coincida con r_i. Esto sería conveniente ya que el residuo nos da información útil de nuestra solución y no queremos trabajar con dos secuencias de vectores auxiliares. Por la relación (1.10) tenemos que r_i puede ser escrito como

$$r_i = (I - AQ_{i-1}(A))r_0$$

donde Q_{i-1} es un polinomio de grado $i-1$. Insertando esta relación en (2.10) con $v_{i+1} = r_i$ e igualando los coeficientes de r_0 a ambos lados obtenemos

$$h_{i+1,i} + h_{i,i} + h_{i-1,i} = 0$$

lo cual define $h_{i+1,i}$. Denotamos por R_i la matriz con columnas r_i:

$$R_i = (r_0, ..., r_{i-1})$$

por tanto tenemos

$$AR_i = R_{i+1}T_{i+1,i} \tag{2.12}$$

con $T_{i+1,i}$ la matriz diagonal cuyos elementos distintos de cero son los definidos por $h_{i,j}$.

Como estamos buscando una solución $x_i \in \mathcal{K}_i(A; r_0)$, este vector puede escribirse como una combinación de los vectores base del subespacio de Krylov, es decir, $x_i = R_i y_i$.

Por la condición de Ritz-Galerkin, el residuo x_i debe ser ortogonal respecto a $r_1, ..., r_i$:

$$R_i^T(Ax_i - b) = 0,$$

y por tanto

$$R_i^T AR_i y_i - R_i^T b = 0.$$

Usando la ecuación (2.12), obtenemos

$$R_i^T R_i T_{i,i} y_i = \|r_0\|_2^2 e_1$$

Ya que $R_i^T R_i$ es una matriz diagonal con elementos $\|r_0\|_2^2$ hasta $\|r_{i-1}\|_2^2$, encontramos la solución resolviendo

$$T_{i,i}y_i = e_1 \Rightarrow y_i \Rightarrow x_i = R_i y_i$$

Hasta ahora sólo hemos usado el hecho de que A sea simétrica y que $T_{i,i}$ es singular. Este método de Krylov es conocido como método de Lanczos para sistemas simétricos. Notemos que para algún $j \leq n-1$ la construcción de la base ortogonal debe finalizar. En esa iteración tendremos $AR_j = R_{j+1}T_{j+1,j+1}$. Sea y la solución del sistema $T_{j+1,j+1}y = e_1$ y $x_{j+1} = R_{j+1}y$. Entonces sigue que $x_{j+1} = x$, es decir, hemos llegado a la solución exacta, ya que $Ax_{j+1} - b = AR_{j+1}y - b = R_{j+1}T_{j+1,j+1}y - b = R_{j+1}e_1 - b = r_0 - b = 0$ (recordamos que hemos asumido que $x_0 = 0$).

El método del Gradiente Conjugado es una variante de la anterior aproximación que ahorra almacenamiento y costo computacional. Si seguimos el enfoque anterior vemos que para resolver las ecuaciones proyectadas necesitamos almacenar todas las columnas de R_i para poder recuperar la aproximación x_i en la iteración actual. Esto puede hacerse de una manera más eficiente en cuanto a memoria se refiere. Si asumimos que la matriz A además es definida positiva, entonces debido a la relación

$$R_i^T AR_i = R_i^T R_i T_{i,i}$$

vemos que $T_{i,i}$ puede ser transformada en una matriz simétrica definida positiva. Por tanto $T_{i,i}$ tiene descomposición LU sin pivoting:

$$T_{i,i} = L_i U_i,$$

con L_i bidiagonal inferior y U_i bidiagonal superior con diagonal unitaria. Por tanto,

$$x_i = R_i y_i = R_i T_{i,i}^{-1} e_1 = (R_i U_i^{-1})(L_i^{-1} e_1) \tag{2.13}$$

Nos centramos ahora, por separado, en los factores colocados entre paréntesis.

1. $$L_i = \begin{pmatrix} \delta_0 & & & & \\ \phi_0 & \delta_1 & & & \\ & \phi_1 & \ddots & & \\ & & \ddots & \ddots & \\ & & & \phi_{i-2} & \delta_{i-1} \end{pmatrix}$$

   Con $q = L_i^{-1} e_1$ podemos obtener q resolviendo $L_i q = e_1$. Por tanto $q_0 = 1/\delta_0$ y $q_{i-1} = -\phi_{i-2} q_{i-2}/\delta_{i-1}$, y así podemos calcular los elementos de q de manera recurrente.

2. $$R_i = P_i U_i = \begin{pmatrix} p_0 & \cdots & p_{i-2} & p_{i-1} \\ & & & \end{pmatrix} \begin{pmatrix} 1 & \epsilon_0 & & & \\ & 1 & \epsilon_1 & \ddots & \\ & & & \ddots & \epsilon_{i-2} \\ & & & & 1 \end{pmatrix}$$
   $$\Rightarrow r_{i-1} = \epsilon_{i-2} p_{i-2} + p_{i-1},$$

así que el vector p_{i-1} puede ser computado recurrentemente como

$$p_{i-1} = r_{i-1} - \epsilon_{i-2} p_{i-2}$$

Juntando ambas relaciones de recurrencia obtenemos

$$x_i = \begin{pmatrix} p_0 & \cdots & p_{i-1} \\ & & \end{pmatrix} \begin{pmatrix} \vdots \\ \vdots \\ q_{i-1} \end{pmatrix}$$
$$= x_{i-1} + q_{i-1} p_{i-1}$$

En principio el método no parece muy complejo: la matriz tridiagonal es generada a partir de una simple recurrencia de tres términos y esta matriz

es factorizada y resuelta para ambos factores. Sin embargo, veremos que no es necesario generar T_{ii} explícitamente.

Para ver esto simplificamos la notación de nuestras relaciones de recurrencia y entonces explotamos las propiedades de ortogonalidad bajo el método de Lanczos. Primero escribimos $\alpha_i \equiv q_i$ y $\beta_i \equiv \epsilon_i$.

Entonces nuestras relaciones de recurrencia de dos términos pueden reescribirse cómo

$$p_{i-1} = r_{i-1} - \beta_{i-2} p_{i-2} \tag{2.14}$$

$$x_i = x_{i-1} + \alpha_{i-1} p_{i-1} \tag{2.15}$$

$$r_i = r_{i-1} - \alpha_{i-1} A p_{i-1}. \tag{2.16}$$

El vector $\alpha_{i-1} p_{i-1}$ es el vector de corrección que deja el nuevo mínimo de $\|x - x_i\|_A$. Por tanto es tangente a la superficie $\|x - z\|_A = \|x - x_i\|_A$, para $z \in \mathcal{K}^{i+1}(A; r_0)$. Los vectores p_j son A-ortogonales y pueden ser interpretados como gradientes conjugados para $\|x - z\|_A$, vista como función de z. Esto da nombre al método.

Proposición 8. *Sea $r_i = b - Ax_i$, $i = 0, 1, ...$ el vector residuo producido por el algoritmo de Lanczos y p_i, $i = 0, 1, ...$, los vectores auxiliares definidos anteriormente. Entonces,*

1. *Cada vector residuo es de la forma $r_i = \sigma_i v_{i+1}$, donde σ_i es cierto escalar. Cómo resultado, los vectores residuo son ortogonales entre sí, es decir $(r_i, r_j) = 0 \quad i \neq j$.*
2. *Los vectores auxiliares p_i forman un sistema de vectores A-conjugados, es decir, $(Ap_i, p_j) = 0$ para $i \neq j$*

Demostración. La primera parte se deduce de la relación (2.11) cómo explicamos anteriormente.

Para la segunda tenemos que probar que $P_i^T A P_i$ es una matriz diagonal , donde recordamos que $P_i = V_i U_i^{-1}$. De esto sigue que

$$\begin{aligned} P_i^T A P_i &= U_i^{-T} V_i^T A V_i U_i^{-1} \\ &= U_i^{-T} T_i U_i^{-1} \\ &= U_i^{-T} L_i \end{aligned}$$

Observemos que $U_i^{-T} L_i$ es triangular inferior y simétrica por ser igual a la matriz simétrica $P_i^T A P_i$:

$$(P_i^T A P_i)^T = P_i^T A^T (P_i^T)^T = P_i^T A P_i$$

Por tanto debe ser una matriz diagonal. □

Ya que $r_i^T r_{i-1} = 0$, derivamos α_{i-1}:

$$\alpha_{i-1} = \frac{r_i^T r_{i-1}}{r_{i-1}^T A p_{i-1}}$$

Usando (2.14) obtenemos que

$$r_{i-1}^T A p_{i-1} = (p_{i-1} - \beta_{i-2} p_{i-2})^T A p_{i-1} = p_{i-1}^T A p_{i-1} - \beta_{i-2} p_{i-2}^T A p_{i-1} = p_{i-1}^T A p_{i-1}$$

Y llegamos a la elegante expresión:

$$\alpha_{i-1} = \frac{r_i^T r_{i-1}}{p_{i-1}^T A p_{i-1}}$$

Derivamos ahora una expresión similar para β_{i-1}. Aplicando de nuevo (2.14) obtenemos:

$$r_i^T A p_{i-1} = p_i^T A p_{i-1} - \beta_{i-1} p_{i-1}^T A p_{i-1}$$

Por tanto

$$\beta_{i-1} = \frac{r_i^T A p_{i-1}}{p_{i-1}^T A p_{i-1}}$$

Manipulamos ahora numerador y denominador utilizando las relaciones anteriores, aplicando al numerador (2.16) $A p_{i-1} = -\frac{1}{\alpha_{i-1}}(r_i - r_{i-1})$:

$$r_i^T A p_{i-1} = -\frac{1}{\alpha_{i-1}}(r_i^T r_i - r_i^T r_{i-1}) = -\frac{1}{\alpha_{i-1}} r_i^T r_i$$

De igual forma el denominador:

$$p_{i-1}^T A p_{i-1} = r_{i-1}^T A p_{i-1} = -\frac{1}{\alpha_{i-1}} r_{i-1}^T r_i + \frac{1}{\alpha_{i-1}} r_{i-1}^T r_{i-1} = \frac{1}{\alpha_{i-1}} r_{i-1}^T r_{i-1}$$

Por tanto

$$\beta_{i-1} = \frac{r_i^T r_i}{r_{i-1}^T r_{i-1}}$$

Notemos que necesitamos sólo dos nuevos productos internos en la iteración i-ésima para calcular ambos coeficientes, precisamente los mismos que en el proceso de Lanczos.

Así pues hemos llegado al conocido método del Gradiente Conjugado. El nombre viene de la propiedad de que los vectores p_i actualizados son A-ortogonales. Cabe destacar que la matriz A sea positiva definida solo se utiliza para garantizar la descomposición de la matriz T_{ii} generada implícitamente. Esto sugiere que el método del Gradiente Conjugado podría también ser efectivo para sistemas no definidos positivos, pero bajo nuestro riesgo.

2.2.2. Notas computacionales

Presentamos a continuación el método del Gradiente conjugado estándar sin precondicionamiento para resolver el sistema $Ax = b$.

Gradiente Conjugado sin precondicionamiento:

```
1  x_0 aproximacion inicial , r_0 = b - Ax_0
2  for i = 1,2,...
3          ρ_{i-1} = r_{i-1}^T r_{i-1}
4          if i = 1
5                  p_i = r_{i-1}
6          else
7                  β_{i-1} = ρ_{i-1}/ρ_{i-2}
8                  p_i = r_{i-1} + β_{i-1} p_{i-1}
9          end
10         q_i = A p_i
11         α_i = ρ_{i-1}/p_i^T q_i
12         x_i = x_{i-1} + α_i p_i
13         r_i = r_{i-1} - α_i q_i
14         if x_i sufiente preciso entonces exit
15         end
16 end
```

El método del gradiente conjugado es con mucha frecuencia usado en combinación con una aproximación conveniente K para A, la cual es llamada precondicionador. Hablaremos de esto más detalladamente en el último capitulo de la memoria.

2.2.3. La convergencia del Gradiente Conjugado

El método del gradiente conjugado (en todo lo que sigue $K = I$) construye en la i-ésima iteración un x_i, que puede ser escrito como,

$$x_i - x = P_i(A)(x_0 - x), \tag{2.17}$$

tal que $\|x_i - x\|_A$ sea minimal sobre todos los polinomios P_i de grado i, con $P_i(0) = 1$. Denotamos los autovalores y autovectores normalizados de A por λ_j, z_j. Escribimos $r_0 = \sum_j \gamma_j z_j$. De esto sigue que

$$r_i = P_i(A)r_0 = \sum_j \gamma_j P_i(\lambda_j) z_j \tag{2.18}$$

y por tanto

$$\|x_i - x\|_A = \sum_j \frac{\gamma_j^2}{\lambda_j} P_i^2(\lambda_j). \tag{2.19}$$

Notemos que los únicos λ_j que juegan un papel en este proceso son aquellos para los cuales $\gamma_j \neq 0$. En particular, si A es semidefinida, es decir, existe algún $\lambda = 0$, entonces esto no será ningún problema en el proceso de minimización ya que el correspondiente γ es cero también. La situación en que γ es pequeño debido a errores de redondeo, se discute en [4].

Obtenemos así una cota superior para el error (en A-norma)

$$\|x_i - x\|_A = \sum_j \frac{\gamma_j^2}{\lambda_j} P_i^2(\lambda_j) \leq \sum_j \frac{\gamma_j^2}{\lambda_j} Q_i^2(\lambda_j) \tag{2.20}$$

$$\leq \max_{\lambda_j} Q_i^2(\lambda_j) \sum_j \frac{\gamma_j^2}{\lambda_j}, \tag{2.21}$$

para cualquier polinomio arbitrario Q_i de grado i con $Q_i(0) = 1$, donde el máximo es alcanzado entre aquellos λ para los cuales $\gamma \neq 0$.

Conseguimos cotas superiores descriptivas seleccionando polinomios adecuados para Q_i. Una cota muy conocida viene de escoger como Q_i el polinomio de Chebyshev C_i de grado i transformado para el intervalo $[\lambda_{min}, \lambda_{max}]$ y escalado de tal forma que su valor en 0 sea 1:

$$Q_i(\lambda) = \frac{C_i(\frac{2\lambda - (\lambda_{min}+\lambda_{max})}{\lambda_{max}-\lambda_{min}})}{C_i(-\frac{(\lambda_{min}+\lambda_{max})}{\lambda_{max}-\lambda_{min}})}.$$

Recordamos las siguientes propiedades de los polinomios de Chebyshev:

$$c_i(x) = \cos(i \arccos(x)) \quad para \quad -1 \leq x \leq 1 \tag{2.22}$$

$$|C_i(x)| \leq 1 \quad -1 \leq x \leq 1 \tag{2.23}$$

$$C_i(x) = \frac{1}{2}\left[(x - \sqrt{x^2-1})^i] + (x + \sqrt{x^2-1})^{-i}\right] \quad x \geq 1 \tag{2.24}$$

$$|C_i(x)| = |C_i(-x)|. \tag{2.25}$$

Con Q_i en vez de el polinomio optimal CG P_i, tenemos

$$\|x_i - x\|_A \leq \sum_j \frac{\gamma_j^2}{\lambda_j} Q_i^2(\lambda_j) \tag{2.26}$$

$$\leq \max_{\lambda_{min}, \lambda_{max}} |Q_i^2(\lambda_j)| \|x_i - x\|_A^2. \tag{2.27}$$

Utilizando propiedades de los polinomios de Chebyshev podemos derivar una elegante cota superior para el error en A-norma. Primero notemos que:

$$|Q_i(\lambda)| \leq \frac{1}{|C_i(\frac{(\lambda_{min}+\lambda_{max})}{\lambda_{max}-\lambda_{min}})|} \quad para \lambda_{min} \leq \lambda \leq \lambda_{max}$$

Usaremos también que para $0 < a < b$ tenemos que

$$C_i(\frac{b+a}{b-a}) \geq \frac{1}{2}(x+\sqrt{x^2-1})^i,$$

con $x = (b+a)/(b-a)$.

Con $a \equiv \lambda_{min}$ y $b \equiv \lambda_{max}$, tenemos que para $a \leq \lambda \leq b$

$$|Q_i(\lambda)| \leq \frac{1}{|C_i(\frac{(b+a)}{b-a})|} \tag{2.28}$$

$$\leq 2\left(\frac{1}{(x+\sqrt{x^2-1})^i}\right) \quad con\ x \equiv \frac{b+a}{b-a} \tag{2.29}$$

$$= 2(x-\sqrt{x^2-1})^i \tag{2.30}$$

$$= 2\left(\frac{b+a-2\sqrt{ba}}{b-a}\right)^i \tag{2.31}$$

$$= 2\left(\frac{\sqrt{b}-\sqrt{a}}{\sqrt{b}-\sqrt{a}}\right)^i. \tag{2.32}$$

Con $\kappa \equiv b/a \equiv \lambda_{max}/\lambda_{min}$ obtenemos una cota muy conocida para la A-norma del error:

$$\|x_i - x\|_A \leq 2\left(\frac{\sqrt{k}-1}{\sqrt{k}+1}\right)^i \|x_0 - x\|_A. \tag{2.33}$$

Esta cota muestra que tendremos una convergencia más rápida para números de condición más pequeños. Podemos obtener cotas similares para distribuciones específicas de autovalores. Por ejemplo, si consideramos la situación en que $\lambda_n \gg \lambda_{n-1}$. Entonces seleccionamos para Q_i, el polinomio

$$Q_i(\lambda) = \frac{\lambda_n - \lambda}{\lambda} C_{i-1}\left(\frac{2\lambda-(a+b)}{b-a}\right) / C_{i-1}\left(\frac{-(a+b)}{b-a}\right)$$

con $a \equiv \lambda_{min}, b \equiv \lambda_{max}$.

Esto muestra que para la situación donde λ_{n-1} es relativamente mucho más pequeño que λ_n, la cota del error para el proceso del gradiente conjugado se queda sólo un paso atrás de la cota para un proceso con un número de condición λ_{n-1}/λ_1.

Estos tipos de cotas muestran que es importante tener pequeños números de condición, o, en caso de tener números de condición grandes, tener una buena distribución de los autovalores que causan este alto número de condición. En este caso decimos que parte del espectro está agrupado. El propósito del precondicionamiento es reducir el número de condicionamiento κ y/o agrupar los autovalores.

Efectos locales en el comportamiento de la convergencia

Cotas superiores como (2.33) muestran que tenemos convergencia global, pero estas no nos ayudan a explicar todo tipo de efectos locales en el comportamiento de la convergencia del gradiente conjugado. Un efecto muy conocido es la convergencia superlineal: en muchas ocasiones la velocidad media convergencia parece aumentar a medida que avanzan las iteraciones. Como hemos visto, el algoritmo del Gradiente Conjugado no es más que una implementación más eficiente del algoritmo de Lanczos. Los autovalores de la matriz tridiagonal T_i implícitamente generada son los Ritz values de A con respecto al actual subespacio de Krylov. Es conocido de la teoría de Lanczos que estos Ritz values convergen hacia los autovalores de A y que, en general, los autovalores extremos de A tienen buenas aproximaciones desde iteraciones tempranas. Además, la velocidad de convergencia depende en cómo de bien estos autovalores están separados los unos de los otros. Estos nos ayuda a entender el comportamiento de la convergencia superlineal en el método del gradiente conjugado. Puede mostrarse que tan pronto cómo uno de los autovalores extremos está moderadamente bien aproximado por un Ritz value, el proceso converge a partir de entonces como un proceso en el cual este autovalor no esté, es decir, un proceso con un número de condición reducido. Notemos que nos referimos convergencia superlineal como convergencia lineal con un factor que es gradualmente decreciente durante el proceso, tanto cómo los autovalores extremos van siendo mejor aproximados. En este texto no entraremos en los detalles teóricos de esto, para más información ver [9].

2.3. FOM

FOM (de las siglas Full Orthogonalization Method) es un método de proyección ortogonal con $\mathcal{L} = \mathcal{K} = \mathcal{K}_i(A, r_0)$, como ya hemos explicado en el

apartado 2.1.1 dada una aproximación inicial al sistema original (1.1) en FOM la solución viene dada por:

$$x_i = x_0 + V_i y_i \tag{2.34}$$

Donde

$$y_i = H_i^{-1}(\beta e_1) \tag{2.35}$$

Algoritmo: FOM

```
1  Calculo r_0 = b - Ax_0, β := ||r_0||_2 y v_1 := r_0/β
2  Define la matriz i × i H_i = {h_kj}_{k,j=1,...,i}. Establecemos H_i = 0
3  for j = 1 ,... i
4          w_j = Av_j
5          for ℓ = 1, ..., j
6                  h_ℓj = (w_j, v_ℓ)
7                  w_j = w_j - h_ℓj v_k
8          end
9          h_{j+1,j} = ||w||_2
10         if h_{j+1,j} = 0
11                 i = j
12                 Salimos del bucle a la linea 15
13         end
14         v_{j+1} = w_j/h_{j+1,j}
15 end
16 Calculo y_i = H_i^{-1}(βe_1) y x_i = x_0 + V_i y_i
```

El anterior algoritmo depende del parámetro i, el cuál designa la dimensión del subespacio de Krylov. En la práctica es deseable seleccionar i de forma dinámica, para ello debemos poder calcular la norma residual de x_i de una forma con un costo computacional factible. Entonces podríamos parar el algoritmo en la iteración adecuada usando dicha información. Pero para FOM, igual que para el gradiente conjugado se satisface la relación (2.11), dejamos esto enunciado en la siguiente proposición omitiendo la prueba que ya vimos en el caso del Gradiente Conjugado.

Proposición 9. *El vector residuo de la solución aproximada x_i calculado por el algoritmo FOM es tal que*

$$b - Ax_i = -h_{i+1,i} e_i^T y_i v_{i+1}$$

y, entonces,

$$\|b - Ax_i\|_2 = h_{i+1,i} |e_i^T y_i| \tag{2.36}$$

Determinamos ahora una estimación del costo computacional de cada iteración del algoritmo. Sea $Nz(A)$ el número de elementos distintos de cero

de A, entonces i iteraciones del algoritmo de Arnoldi requieren i productos matriz-vector lo cuál denota un costo de $2i \times Nz(A)$. Cada iteración de Gram-Schmidt cuesta aproximadamente $4 \times j \times n$ operaciones, que en un total de i operaciones será aproximadamente $2i^2n$. Por tanto una iteración de FOM tiene un costo aproximado de

$$2Nz(A) + 2in$$

En cuánto a almacenamiento, i vectores de longitud n son requeridos para guardar la base V_i. Además debemos usar vectores para almacenar la solución actual y el lado derecho de la ecuación y un vector de ceros para los productos matriz-vector. La matriz Hessenberg H_i también debe ser almacenada. El total es

$$(i+3)n + \frac{i^2}{2}$$

En la mayoría de casos i es pequeño comparado con n, por tanto dicho costo de almacenamiento está dominado por el primer término.

Cuando i incrementa el costo computacional incrementa al menos como $O(i^2)n$ debido a la ortogonalización de Gram-Schmidt. El costo en cuánto a memoria incrementa como $O(in)$. Si n es grande este limita cuanto de grande puede ser i y por tanto el número de aproximaciones que podemos hacer a nuestra solución. Hay dos posibles soluciones, la primera es reiniciar el algoritmo periódicamente y la segunda "truncar" la ortogonalización en el algoritmo de Arnoldi, pero no entraremos en más detalle en este documento.

2.4. GMRES

GMRES, abreviatura de Generalized Minimal Residual Method, es un método de proyección basado en coger $\mathcal{K} = \mathcal{K}_i$ y $\mathcal{L} = A\mathcal{K}_i$, dónde $\mathcal{K}_i$ es el subespacio de Krylov de dimensión i con $v_1 = r_0/\|r_0\|_2$. En este caso se trata de una proyección oblicua al espacio de Krylov en vez de una proyección ortogonal. Cómo ya hemos descrito en el apartado 2.1.2, dicha técnica se basa en minimizar la norma residual sobre todos los vectores pertenecientes a $x_0+\mathcal{K}_i$.

Hay dos maneras de derivar el algoritmo. La primera explotando las propiedades de optimalidad y la relación (1.12) $AV_i = V_{i+1}H_{i+1,i}$. Esto es lo que hemos visto en la sección 2.1.2, donde la aproximación por mínima norma residual nos deja un pequeño problema de mínimos cuadrados por resolver:

$$H_{i+1,i}y = \|r_0\|_2 e_1 \tag{2.37}$$

En GMRES esto es hecho de forma eficiente mediante rotaciones de Givens, que anulan los elementos de la subdiagonal en la matriz Hessenberg superior H_i, más adelante abarcaremos esto en detalle.

La segunda manera de derivar el algoritmo GMRES es usar la formula estándar (1.7) $x = x_0 + V(W^TAV)^{-1}W^Tr_0$ con $V = V_i$ y $W = AV_i$. De la ecuación (2.6) obtenemos la relación para el residuo de la aproximación i-ésima $b - Ax_i = V_{i+1}(\beta e_1 - H_{i+1,i}y)$. Vamos a imponer ahora la ortogonalidad del residuo al espacio $\mathcal{L} = A\mathcal{K}_i$

$$(AV_i^T)(V_{i+1}(\beta e_1 - H_{i+1,i}y)) = H_{i+1,i}^T V_{i+1}^T V_{i+1}(\beta e_1 - H_{i+1,i}y) = 0$$

En la primera igualdad aplicamos la relación (1.12) y teniendo en cuenta que $V_{i+1}^T V_{i+1} = Id$ obtenemos

$$H_i^T H_{i+1,i}y = H_{i+1,i}^T \beta e_1$$

Lo cuál es exactamente la condición obtenida anteriormente al buscar la aproximación x_i que minimice el residuo $r_i = b - Ax_i$.

Una clara dificultad de este algoritmo es que no nos proporciona la solución aproximada x_i en cada iteración y por tanto no es fácil determinar cuando parar. Una posible solución sería calcular la solución aproximada x_i en intervalos regulares y revisar la convergencia testeando el residuo. Sin embargo, hay una manera más elegante de resolver este problema relacionada con la manera en la que resolvemos el problema de mínimos cuadrados (2.37).

Para resolver el problema de mínimos cuadrados $\|\beta e_1 - H_{i+1,i}y\|$, es natural transformar la matriz Hessenberg en una matriz triangular superior usando rotaciones de Givens. Definimos la matriz de rotación

$$\Omega_i = \begin{pmatrix} 1 & & & & & & & \\ & \ddots & & & & & & \\ & & 1 & & & & & \\ & & & c_i & s_i & & & \\ & & & -s_i & c_i & & & \\ & & & & & 1 & & \\ & & & & & & \ddots & \\ & & & & & & & 1 \end{pmatrix}$$

con $c_i^2 + s_i^2 = 1$. Si han sido ejecutadas i iteraciones de GMRES entonces estas matrices tienen dimensión $(i+1) \times (i+1)$.

Multiplicamos la matriz Hessenberg $H_{i+1,i}$ y el correspondiente lado derecho $g_0 \equiv \beta e_1$ por una secuencia de tales matrices. Cada iteración los coeficientes s_i, c_i son seleccionados para eliminar el elemento $h_{i+1,i}$.

Por ejemplo para el caso $i = 5$ tendríamos

$$H_{65} = \begin{pmatrix} h_{11} & h_{12} & h_{13} & h_{14} & h_{15} \\ h_{21} & h_{22} & h_{23} & h_{24} & h_{25} \\ & h_{32} & h_{33} & h_{34} & h_{35} \\ & & h_{43} & h_{44} & h_{45} \\ & & & h_{54} & h_{55} \\ & & & & h_{65} \end{pmatrix}, \quad g_6 = \begin{pmatrix} \beta \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

Entonces, para eliminar $h_{2,1}$, premultiplicamos H_{65} por

$$\Omega_1 = \begin{pmatrix} c_1 & s_1 & & & \\ -s_1 & c_1 & & & \\ & & 1 & & \\ & & & 1 & \\ & & & & 1 \end{pmatrix}$$

con

$$s_1 = \frac{h_{21}}{\sqrt{h_{11}^2 + h_{21}^2}}, \quad c_1 = \frac{h_{11}}{\sqrt{h_{11}^2 + h_{21}^2}}$$

y obtenemos la siguiente matriz y lado derecho

$$H_{65}^{(1)} = \begin{pmatrix} h_{11}^{(1)} & h_{12}^{(1)} & h_{13}^{(1)} & h_{14}^{(1)} & h_{15}^{(1)} \\ & h_{22}^{(1)} & h_{23}^{(1)} & h_{24}^{(1)} & h_{25}^{(1)} \\ & h_{32} & h_{33} & h_{34} & h_{35} \\ & & h_{43} & h_{44} & h_{45} \\ & & & h_{54} & h_{55} \\ & & & & h_{65} \end{pmatrix}, \quad g_6^{(1)} = \begin{pmatrix} c_1\beta \\ -s_1\beta \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

Ahora podemos premultiplicar la anterior matriz y lado derecho de nuevo por una matriz de rotación Ω_2 para eliminar h_{32}. Esto es posible cogiendo

$$s_2 = \frac{h_{32}}{\sqrt{(h_{22}^{(1)})^2 + h_{32}^2}}, \quad c_2 = \frac{h_{22}^{(1)}}{\sqrt{(h_{22}^{(1)})^2 + h_{32}^2}}$$

Este proceso de eliminación continúa hasta que la i-ésima rotación es aplicada, lo cuál transforma el problema inicial en el mismo involucrando la

siguiente matriz y lado derecho,

$$H_{65}^{(5)} = \begin{pmatrix} h_{11}^{(5)} & h_{12}^{(5)} & h_{13}^{(5)} & h_{14}^{(5)} & h_{15}^{(5)} \\ & h_{22}^{(5)} & h_{23}^{(5)} & h_{24}^{(5)} & h_{25}^{(5)} \\ & & h_{33}^{(5)} & h_{34}^{(5)} & h_{35}^{(5)} \\ & & & h_{44}^{(5)} & h_{45}^{(5)} \\ & & & & h_{55}^{(5)} \\ & & & & 0 \end{pmatrix}, \quad g_6^{(5)} = \begin{pmatrix} \gamma_1 \\ \gamma_2 \\ \gamma_3 \\ . \\ . \\ \gamma_6 \end{pmatrix} \tag{2.38}$$

Generalmente, los escalares c_i y s_i de la i-ésima rotación Ω_i están definidos cómo

$$s_i = \frac{h_{i+1,i}}{\sqrt{(h_{ii}^{(i-1)})^2 + h_{i+1,i}^2}}, \quad c_i = \frac{h_{ii}^{(i-1)}}{\sqrt{(h_{ii}^{(i-1)})^2 + h_{i+1,i}^2}} \tag{2.39}$$

Definimos Q_i cómo el producto de matrices Ω_i,

$$Q_i = \Omega_i \Omega_{i-1} ... \Omega_1$$

y

$$R_{i+1,i} = H_{i+1,i}^{(i)} = Q_i H_{i+1,i}, \tag{2.40}$$

$$g_{i+1}^{(i)} = Q_i(\beta e_1) = (\gamma_1, ..., \gamma_{i+1})^T. \tag{2.41}$$

Ya que Q_i es unitaria,

$$min\|\beta e_1 - H_{i+1,i} y\|_2 = min\|g_{i+1}^{(i)} - R_{i+1,i} y\|_2. \tag{2.42}$$

La solución del anterior problema de mínimos cuadrados es obtenida simplemente resolviendo un sistema triangular eliminando la última fila de la matriz $R_{i+1,i}$ y del lado derecho $g_{i+1}^{(i)}$ en (2.42). Además, para la solución y_* que minimiza $\|\beta e_1 - H_{i+1,i} y\|_2$ el residuo no es más que el término γ_6 de la anterior ilustración. Formalizamos esto con la siguiente proposición.

Proposición 10. *Sean Ω_j, $j = 1, ..., i$ las matrices de rotación utilizadas para transformar $H_{i+1,i}$ en la matriz triangular superior $R_{i+1,i}$ y el lado derecho $g_{i+1}^{(i)} = (\gamma_1, ..., \gamma_{i+1})^T$, que ahora por simplicidad denotaremos simplemente g_{i+1}, definidas por* (2.40) *y* (2.41)*. Denotamos por R_i la matriz $i \times i$ triangular superior obtenida al eliminar la última fila de $R_{i+1,i}$ y por g_i el vector m-dimensional obtenido al eliminar la última componente de $g_{m+1}^{(m)}$. Entonces,*

1. *El rango de AV_i es igual al rango de R_i. En particular,*
2. *El vector y_i el cuál minimiza $\|\beta e_1 - H_{i+1,i} y\|_2$ está dado por*

$$y_i = R_i^{-1} g_i$$

3. El vector residual en la iteración i satisface

$$b - Ax_i = V_{i+1}(\beta e_1 - H_{i+1,i}y) = V_{i+1}Q_i^T(\gamma_{i+1}e_{i+1}) \tag{2.43}$$

y, como resultado,

$$\|b - Ax_i\|_2 = |\gamma_{i+1}|. \tag{2.44}$$

Demostración. Para probar la primera parte, usamos (1.12), para obtener la relación

$$AV_i = V_{i+1}H_{i+1,i} = V_{i+1}Q_i^T Q_i H_{i+1,i} = V_{i+1}Q_i^T R_{i+1,i}$$

Ya que $V_{i+1}Q_i^T$ es unitario, el rango de AV_i es el mismo que el de $R_{i+1,i}$, que coincide con el de R_i ya que estas dos matrices se diferencian por sólo una fila de ceros. Si $r_{i,i} = 0$ entonces R_i tiene rango $\leq i-1$ y como resultado AV_i también es de rango $\leq i-1$. Como V_i tiene rango máximo A debe ser singular.

La segunda parte ya la hemos probado esencialmente antes de la proposición. Para cualquier vector y tenemos,

$$\begin{aligned}\|\beta e_1 - H_{i+1,i}y\|_2^2 &= \|Q_i(\beta e_1 - H_{i+1,i}y)\|_2^2 \\ &= \|g_{i+1} - R_{i+1,i}y\|_2^2 \\ &= |\gamma_{i+1}|^2 + \|g_i - R_{i+1,i}y\|_2^2\end{aligned}$$

El mínimo del lado izquierdo es alcanzado cuando el segundo termino del lado derecho es cero. Como R_i es invertible, esto sucede cuando $y = R_i^{-1}g_i$.

Para probar la tercera parte, empezamos con las definiciones usadas para GMRES y la relación (2.6). Para cualquier $x = x_0 + V_i y$,

$$\begin{aligned}b - Ax =& V_{i+1}(\beta e_1 - H_{i+1,i}y) \\ =& V_{i+1}Q_i^T Q_i(\beta e_1 - H_{i+1,i}y) \\ =& V_{i+1}Q_i^T(g_{i+1} - R_{i+1,i}y)\end{aligned}$$

Como hemos visto en la prueba de la segunda parte anterior, la 2-norma de $g_{i+1} - R_{i+1,i}y$ es minimizada cuando y anula todas las componentes de lado derecho g_{i+1} menos la última, la cuál es igual a γ_{i+1}. Como resultado,

$$b - Ax_i = V_{i+1}Q_i^T(\gamma_{i+1}e_{i+1})$$

lo cual es (2.43). El resultado (2.44) sigue de la ortonomalidad de los vectores columna de la matriz $V_{m+1}Q_m^T$. □

Hasta ahora sólo hemos descrito un proceso para computar la solución de mínimos cuadrados y_i de (2.37). Este método con rotaciones puede ser usado también para resolver el sistema lineal (2.35) para el método FOM. La única diferencia es que la última rotación Ω_i debe ser omitida.

Este enfoque nos permite obtener la norma residual en cada iteración sin operaciones aritméticas adicionales. Para ilustrar esto empezamos desde (2.38), es decir, asumimos que las primeras i rotaciones han sido ya realizadas. Asumimos que el test llevado a cabo dicta que debemos continuar, es decir, debemos ejecutar una iteración más en el algoritmo de Arnoldi calculando Av_6 y la sexta columna de H_{76}. Esta columna es adherida a R_{65} a la cuál le añadimos una fila de ceros para alcanzar la dimensión. Entonces la previas rotaciones $\Omega_1, \Omega_2, ..., \Omega_5$ son aplicadas a esta ultima columna. Después de esto se obtiene la siguiente matriz y lado derecho:

$$R_{65} = \begin{pmatrix} h_{11}^{(5)} & h_{12}^{(5)} & h_{13}^{(5)} & h_{14}^{(5)} & h_{15}^{(5)} & h_{16}^{(5)} \\ & h_{22}^{(5)} & h_{23}^{(5)} & h_{24}^{(5)} & h_{25}^{(5)} & h_{26}^{(5)} \\ & & h_{33}^{(5)} & h_{34}^{(5)} & h_{35}^{(5)} & h_{36}^{(5)} \\ & & & h_{44}^{(5)} & h_{45}^{(5)} & h_{46}^{(5)} \\ & & & & h_{55}^{(5)} & h_{56}^{(5)} \\ & & & & 0 & h_{66}^{(5)} \\ & & & & 0 & h_{76} \end{pmatrix}, \quad g_6^{(5)} = \begin{pmatrix} \gamma_1 \\ \gamma_2 \\ \gamma_3 \\ . \\ . \\ \gamma_6 \\ 0 \end{pmatrix} \tag{2.45}$$

Ahora continuamos de igual forma que antes, aplicando la siguiente rotación para eliminar el elemento h_{76}:

$$s_6 = \frac{h_{76}}{\sqrt{(h_{66}^{(5)})^2 + h_{76}^2}}, \quad c_1 = \frac{h_{66}^{(5)}}{\sqrt{(h_{66}^{(5)})^2 + h_{76}^2}}$$

obteniendo

$$H_6^{(5)} = \begin{pmatrix} r_{11} & r_{12} & r_{13} & r_{14} & r_{15} & r_{16} \\ & r_{22} & r_{23} & r_{24} & r_{25} & r_{26} \\ & & r_{33} & r_{34} & r_{35} & r_{36} \\ & & & r_{44} & r_{45} & r_{46} \\ & & & & r_{55} & r_{56} \\ & & & & & r_{66} \\ & & & & & 0 \end{pmatrix}, \quad g_6^{(6)} = \begin{pmatrix} \gamma_1 \\ \gamma_2 \\ \gamma_3 \\ . \\ . \\ c_6\gamma_6 \\ -s_6\gamma_6 \end{pmatrix} \tag{2.46}$$

Si la norma $|\gamma_{i+1}|$ es lo suficiente pequeña detenemos el proceso. Eliminamos la última fila de $R_{i+1,i}$ y $g_i^{(i)}$ y resolvemos el sistema triangular superior

resultante obteniendo y_i, así calculamos la aproximación $x_i = x_0 + V_i y_i$. De (2.46) obtenemos la siguiente relación

$$\gamma_{j+1} = -s_j \gamma_j. \tag{2.47}$$

Si $s_j = 0$ entonces la norma residual debe ser cero, lo cuál significa que la aproximación es exacta en la iteración j-ésima.

Al examinar el algoritmo GMRES vemos que la única posibilidad de salir -breakdown- es dentro del bucle del algoritmo de Arnoli, cuando $\hat{v}_{j+1} = 0$, es decir, $h_{j+1,j} = 0$ en dicha iteración j. Entonces el algoritmo para ya que no podemos generar el siguiente vector en el algoritmo de Arnoldi. En esta situación el vector residual es cero, es decir, alcanzamos la solución exacta en dicha iteración. Es más, lo contrario también es verdad: si el algoritmo para en la iteración j con $b - Ax_j = 0$, entonces $h_{j+1,j} = 0$.

Proposición 11. *Sea A una matriz no singular. Entonces el algoritmo GMRES para en la iteración j, es decir, $h_{j+1,j} = 0$ si y sólo si la solución aproximada es exacta.*

Demostración. Para probar la primera implicación, observamos que si $h_{j+1,j} = 0$, entonces $s_j = 0$. De hecho, ya que A es no singular, por la primera parte de la proposición 10 $r_{jj} = h_{j+1,j}$ es distinto de cero y por (2.39) esto implica que $s_j = 0$. Entonces por las relaciones (2.44) y (2.47) llegamos a que $r_j = 0$. □

De manera similar al algoritmo FOM, GMRES es impracticable cuando i es de gran dimensión ya que aumenta tanto a memoria como el costo computacional requeridos, los cuáles son idénticos que FOM. De la misma forma hay dos remedios o posibles soluciones, la más sencilla es reiniciar el algoritmo cada m iteraciones, este algoritmos es denominado como GMRES(m):

Algoritmo GMRES(m) sin precondicionamiento con Gram-Schmidt modificado:

```
1  r = b − Ax_0 para una primera aproximacion incial x_0
2  x = x_0
3  for j = 1, 2, ...
4          β = ‖r‖_2 , v_1 = r/β , γ = βe_1
5          for i = 1, 2..., m
6                  w = Av_i
7                  for k = 1, ..., i
8                          h_{k,i} = v_k^T w , w = w − h_{k,i} v_k
9                  h_{i+1,i} = ‖w‖_2 , v_{i+1} = w/h_{i+1,i}
10                 r_{1,i} = h_{1,i}
11                 for k = 2, ..., i
```

```
12                                  α = c_{k-1} r_{k-1,k} + s_{k-1} h_{k,i}
13                                  r_{k,i} = -s_{k-1} r_{k-1} + c_{k-1} h_{k,i}
14                                  r_{k-1,k} = α
15                          δ = sqrt(r_{i,i}^2 + h_{i+i,i}^2)  ,  c_i = r_{i,i}/δ ,  s_i = h_{i+1,i}/δ
16
17                          r_{i,i} = c_i r_{i,i} + s_i h_{i+1,i}
18                          γ_{i+1} = -s_i γ_i ,  γ_i = c_i γ_i
19                          ρ = |γ_{i+1}|  (= ||b - A x_{(j-1)m+i}||_2)
20                          if ρ es suficiente pequeno then
21                                  (n_r = i  go to SOL)
22          n_r = m ,  y_{n_r} = γ_{n_r}/r_{n_r,n_r}
23  SOL:    for  k = n_r - 1, ..., 1
24                          y_k = (γ_k - Σ_{i=k+1}^{n_r} r_{k,i} y_i)/r_{k,k}
25          x = x + Σ_{i=1}^{n_r} y_i v_i
26  end
```

Nota que lo discutido anteriormente puede ser implementado cómo criterio de parada y calcular la norma residual en cada iteración j sin necesidad de calcular la solución, parando así el programa cuando esta sea suficientemente pequeña.

2.5. MINRES

Cuando A es simétrica la matriz $H_{i+1,i}$ se reduce a una matriz tridiagonal $T_{i+1,i}$. Podemos explotar esta propiedad para obtener cortas relaciones de de recurrencia. Como en GMRES buscamos

$$x_i \in \{r_0, Ar_0, ..., A^{i-1}r_0\}, \quad x_i = R_i y_i$$

$$\|Ax_i - b\|_2 = \|AR_i y_i - b\|_2$$

$$= \|R_{i+1} T_{i+1,i} y_i - b\|_2,$$

tal que esta norma residual sea mínima. Ahora explotamos el hecho de que $R_{i+1}D_{i+1}^{-1}$, con

$$D_{i+1} = diag\left(\|r_0\|_2, \|r_1\|_2, ..., \|r_i\|_2\right),$$

es una transformación ortonormal con respecto al subespacio de Krylov actual:

$$\|Ax_i - b\|_2 = \|D_{i+1} T_{i+1,i} y_i - \|r_0\|_2 e_1\|_2$$

y esta expresión final puede ser vista como un problema de mínimos cuadrados.

El elemento en la posición $(i+1, i)$ puede transformarse a 0 por una simple rotación Givens y el sistema tridiagonal superior resultante (el resto de elementos subdiagonales serán eliminados en las iteraciones previas) puede ser resuelto de forma simple como veremos ahora.

EL efecto de las rotaciones Givens es que $T_{i+1,i}$ es descompuesta en forma QR:

$$T_{i+1,i} = Q_{i+1,i} R_{i,i},$$

en la cual la matriz ortogonal $Q_{i+1,i}$ es el producto de las rotaciones Givens y $R_{i,i}$ es una matriz triangular superior con tres diagonales distintas de cero. Podemos explotar esta estructura en banda de $R_{i,i}$ para el cálculo de x_i.

La solución de $T_{i+1,i} y_i = \|r_0\| e_1$ puede escribirse como

$$y_i = R_{i,i}^{-1} Q_{i+1,i}^T \|r_0\|_2 e_1,$$

así la solución x_i puede ser calculada como

$$x_i = (V_i R_{i,i}^{-1})(Q_{i+1,i}^T \|r_0\|_2 e_1).$$

Primero calculamos la matriz $W_i = V_i R_i^{-1}$ y es fácil ver que la última columna de W_i es obtenida de las últimas 3 columnas de V_i. El vector $z_i \equiv Q_{i+1,i}^T \|r_0\|_2 e_1$ puede ser también actualizado a partir de una corta recurrencia, ya que z_i sigue de una simple rotación Givens en las dos últimas coordenadas de z_{i-1}. Este par de relaciones de recurrencia nos dejan el método MINRES.

Algoritmo MINRES sin precondicionamiento:

Calculamos $v_1 = b - Ax_0$ para una primera aproximacion incial x_0
$\beta_1 = \|v_1\|_2$; $\eta = \beta_1$;
$\gamma_1 = \gamma_0 = 1$; $\sigma_1 = \sigma_0 = 0$;
$v_0 = 0$; $w_0 = w_{-1} = 0$;
for i = 1 ,2 ,...
La recurrencia de Lanczos:
    $v_i = \frac{1}{\beta_i}$; $\alpha_i = v_i A^T v_i$;
    $v_{i+1} = Av_i - \alpha_i v_i - \beta_i v_{i-1}$
    $\beta_{i+1} = \|v_{i+1}\|_2$
Factorizacion QR:
Rotaciones Givens antiguas en nueva columna de T:
    $\delta = \gamma_i \alpha_i - \gamma_{i-1} \sigma_i \beta_i$;
    $\rho_1 = \sqrt{\delta^2 + \beta_{i+1}^2}$
    $\rho_2 = \sigma_i \alpha_i + \gamma_{i-1} \gamma_i \beta_i$;
    $\rho_3 = \delta_{i-1} \beta_i$
Nuevas rotaciones Givens para el elemento subdiagonal:
    $\gamma_{i+1} = \delta / \rho_1$;
    $\delta_{i+1} = \beta_{i+1} / \rho_1$

```
19  Actualizamos la solucion (con W_i = V_i R_{i,i}^{-1})
20          w_i = (v_i − ρ_3 w_{i−2} − ρ_2 w_{i−1})/ρ_1
21          x_i = x_{i−1} + γ_{i+1} η w_i
22          ‖r_i‖_2 = |σ_{i+1}| ‖r_{i−1}‖_2
23          Comprobamos la convergencia; continuamos si es necesario
24          η = σ_{i+1} η
25  end
```

MINRES es atractivo cuando la matriz A es simétrica indefinida. En el caso de que sea simétrica positiva definida, el gradiente conjugado es preferido. Para un precondicionador simétrico definido positivo de la forma LL^T, el algoritmo MINRES puede ser aplicado para el sistema explícitamente precondicionado

$$L^{-1}AL^{-T}\tilde{x} = L^{-1}b, \quad con\ x = L^{-T}\tilde{x}$$

Sin embargo, no podemos aplicar MINRES sin correr riesgo a $K^{-1}Ax = K^{-1}b$, para K y A simétricas, cuando $K^{-1}A$ no es simétrica. No nos ayuda reemplazar el producto interno por la forma bilineal (x, Ky), con respecto al cual la matriz $K^{-1}A$ es simétrica, ya que esta forma bilineal no define un producto interno si K no es positiva definida. La construcción de precondicionadores eficientes para A simétrica indefinida es un problema muy extenso.

El uso de la relación de recurrencia de tres términos para las columnas W_i hace a MINRES muy vulnerable a los errores de redondeo. Ha sido demostrado que los errores de redondeo se propagan a la solución aproximada con un factor proporcional al cuadrado del número de condición de A, mientras en GMRES estos errores dependen sólo del número de condición en si mismo. Entonces deberíamos tener cuidado con MINRES para sistemas mal condicionados. Si el almacenamiento no es problema GMRES es más recomendable para sistemas mal condicionados, si el almacenamiento sí es un problema entonces podríamos considerar el método SYMMLQ, el cuál no entraremos a explicar más detalladamente en este documento.

Capítulo 3

Teorema de Faber-Manteuffel

Como hemos visto, si A es simétrica el algoritmo de Arnoldi se simplifica en el algoritmo simétrico de Lanczos, el cual requiere cálculos exclusivamente con recurrencias de tres términos. Consecuentemente, en este caso, el método FOM es matemáticamente equivalente al algoritmo del Gradiente Conjugado. De manera similar, el algoritmo GMRES deriva para matrices A hermíticas en el método de residuos conjugados.

Esta claro que los algoritmos tipo CG, es decir, los definidos por recurrencias de pocos términos, son preferibles ya que requieren menos memoria y operaciones por iteración.

En 1981, G. Golub planteó la cuestión de caracterizar condiciones necesarias y suficientes sobre una matriz A para la existencia de métodos tipo gradiente conjugado con recurrencias de tres términos para resolver el sistema lineal $Ax = b$. Esta cuestión fué totalmente resuelta por Faber y Manteuffel [2] en 1984, estableciendo que existe un método tipo gradiente conjugado con recurrencias de $(s+2)$ términos para una matriz A con respecto a un producto interno si y sólo si la matriz adjunta A^* con respecto a dicho producto interno es un polinomio de grado s en A (es decir, A es normal de grado s).

El objetivo de este capítulo, que da título a esta Memoria de Fin de Grado, es exponer este teorema, estableciendo lo que entenderemos genéricamente por un método gradiente, y caracterizando la clase $CG(s)$ de matrices A para las que el sistema lineal $Ax = b$ puede resolverse con un método gradiente conjugado con recurrencias de s términos. El resultado concluye que, excepto ciertas excepciones, estas matrices no son otras que las ya conocidas para las cuales existe un método gradiente conjugado: matrices Hermiticas, $A^* = A$, y matrices de la forma $A = e^{i\theta}(dI + B)$ con $B^* = -B$.

Teorema (teorema de Faber-Manteuffel)**.** *Existe un método gradiente conjugado con recurrencias de s términos existe para la solución de $Ax = b$ si y solo si se cumple una de las dos siguientes condiciones:*

1. *El polinomio mínimo de A tiene grado $\leq s$, o*
2. *A^*, la matriz adjunta de A con respecto a un producto interno, es polinomial en A de grado $\leq s - 2$*

La condición de que A^* sea polinomial en A de algún grado es equivalente a que A sea normal con respecto al producto interno.

La prueba que desarrollamos sigue la de la publicación original de Faber y Manteuffel [2]. Dividimos la prueba en varios apartados, cada uno correspondiente con una sección del capítulo: en la primera sección describimos lo que entendemos por un método gradiente; en la segunda probamos que la optimalidad en un cierto sentido lleva a métodos de gradientes conjungaqdos; la tercera sección discute los métodos gradiente conjugado a la vista de las recurrencias finitas que se utilizan en su implementación, introduciendo la clase $CG(s)$ de matrices. La cuarta sección aborda los lemas y teoremas de caracterización.

3.1. Métodos gradiente

Dado un sistema lineal $Ax = b$ y una aproximación inicial x_0, sea $r_0 = b - Ax_0$ el residuo inicial. Entendemos por un método gradiente un método en el que la solución aproximada en cada iteración adopta la forma

$$x_{i+1} = x_i + \sum_{j=0}^{i} \eta_{ij} r_j, \quad r_j = b - Ax_j. \tag{3.1}$$

Es decir, en cada iteración la nueva aproximación se obtiene de la anterior sumando una combinación lineal de los vectores residuos previos. El término gradiente proviene del hecho de que si A es simétrica ,$r = b - Ax$ es el gradiente de la forma bilineal $Q(x) = \frac{1}{2}((A^{-1}b - x, A(A^{-1}b - x))$. Si A es simétrica definida positiva $Q(x)$ alcanza el mínimo en $x = A^{-1}b$. Por eso se utiliza una combinación lineal de estos residuos para aproximarse a la solución del sistema. Si A no es simétrica definida positiva aún podemos considerar iteraciones de la forma (3.1). Cada r_j es fácilmente computable y si A es invertible, la solución puede ser obtenida con tales iteraciones. Para ver esto, primero mostraremos por inducción que $r_j = p_j(A)r_0$, donde $p_j(z)$

es un polinomio de grado $\leq j$. Tenemos

$$x_1 = x_0 + \eta_{00} r_0$$

Por tanto,

$$r_1 = (I - \eta_{00} A) r_0 = p_1(A) r_0$$

En general,

$$x_{i+1} = x_i + \sum_{j=0}^{i} \eta_{ij} r_j$$

Por tanto, supongamos que es cierto para i ($r_i = p_i(A) r_0$):

$$\begin{aligned} r_{i+1} &= r_i - \sum_{j=0}^{i} \eta_{ij} A r_j = r_i - \eta_{ii} A r_i - \sum_{j=0}^{i-1} \eta_{ij} A r_j \\ &= p_i(A) r_0 - \eta_{ii} A p_i(A) r_0 - A \sum_{j=0}^{i-1} \eta_{ij} p_j(A) r_0 \\ &= [(I - \eta_{ii} A) p_i(A) - A \sum_{j=0}^{i-1} \eta_{ij} p_j(A)] r_0 = p_{i+1}(A) r_0 \end{aligned}$$

Supongamos que $\eta_{ii} \neq 0$ para todo i. Entonces $p_i(z)$ es de grado exacto i y los vectores residuales forman una base para el espacio de Krylov de dimensión i+1 generada por A y r_0, es decir:

$$V_{i+1} = \text{span}\{r_0, Ar_0, ..., A^i r_0\} = \text{span}\{r_0, r_1, ..., r_i\}$$

Por (3.1) vemos que con la adecuada elección de los η_{ij} podemos hacer que $x_{i+1} - x_i$ sea cualquier elemento de V_{i+1}. Ya que A es de dimensión finita, digamos N, existe un primer $\ell \leq N$ tal que $V_\ell = V_{\ell+1}$. Llamemos k a dicho ℓ. Los vectores $\{r_0, ..., A^k r_0\}$ son linealmente dependientes y podemos elegir $\beta_0, .., \beta_k$ tales que

$$\sum_{i=0}^{k} \beta_i A^i r_0 = 0$$

Si A es invertible, o bien $\beta_0 \neq 0$ o bien k puede tomarse más pequeño. Si $\beta_0 \neq 0$ entonces la solución del sistema viene dada por

$$x = x_0 - \frac{1}{\beta_0} \sum_{i=1}^{k} \beta_i A^{i-1} r_0$$

como fácilmente se comprueba de las igualdades siguientes

$$Ax = Ax_0 - \frac{1}{\beta_0}\sum_{i=1}^{k}\beta_i A^i r_0 = Ax_0 - \frac{1}{\beta_0}(-\beta_0 r_0) = Ax_0 + b - Ax_0 = b.$$

Puesto que los vectores residuos generan el espacio de Krylov, la adecuada elección de los η_{ij} en (3.1) proporciona la solución exacta del sistema.

3.2. Optimalidad

Abordamos ahora el problema de elegir los η_{ij} adecuadamente. Una posible solución es forzar una condición de optimalidad. Sea $e_i = A^{-1}b - x_i = x - x_i$ el vector error del i-ésimo iterante.

Supongamos que tenemos una norma asociada a un producto interno:

$$\|x\|^2 = [x, x]$$

Entonces nos gustaría elegir x_i de tal forma que $\|e_i\|$ sea minimizado entre todas las posibles iteraciones de la forma (3.1). En otras palabras, en la iteración i-ésima ponemos

$$x_{i+1} = x_i + \alpha_i p_i$$

donde $p_i \in V_{i+1}$. Queremos elegir $\alpha_i p_i$ tal que $\|e_{i+1}\|$ sea lo más pequeño posible. Tenemos

$$x_{i+1} = x_0 + \sum_{j=0}^{i}\alpha_j p_j;$$

y por tanto

$$e_{i+1} = e_0 - \sum_{j=0}^{i}\alpha_j p_j$$

Calculando la norma tenemos:

$$\|e_{i+1}\|^2 = [e_0, e_0] - \left[e_0, \sum_{j=0}^{i}\alpha_j p_j\right] - \left[\sum_{j=0}^{i}\alpha_j p_j, e_0\right] + \left[\sum_{j=0}^{i}\alpha_j p_j, \sum_{j=0}^{i}\alpha_j p_j\right]$$

Sea $\alpha_j = x_j + iy_j$, y calculemos la derivada parcial respecto a cada x_j, y_j e igualemos a 0 dichas derivadas, para obtener el sistema lineal:

$$\begin{bmatrix} [p_0,p_0] & [p_1,p_0] & \cdots & [p_i,p_0] \\ \vdots & \vdots & \vdots & \vdots \\ [p_0,p_{i-1}] & . & . & [p_i,p_{i-1}] \\ [p_0,p_i] & . & . & [p_i,p_i] \end{bmatrix} \begin{bmatrix} \alpha_0 \\ \vdots \\ \alpha_{i-1} \\ \alpha_i \end{bmatrix} = \begin{bmatrix} [e_0,p_0] \\ \vdots \\ [e_0,p_{i-1}] \\ [e_0,p_i] \end{bmatrix} \tag{3.2}$$

Ya que esto es cierto para todo i, en particular lo será para $i-1$, es decir,

$$\begin{bmatrix} [p_0,p_0] & [p_1,p_0] & \cdots & [p_{i-1},p_0] \\ \vdots & \vdots & \vdots & \vdots \\ [p_0,p_{i-1}] & . & . & [p_{i-1},p_{i-1}] \end{bmatrix} \begin{bmatrix} \alpha_0 \\ \vdots \\ \alpha_{i-1} \end{bmatrix} = \begin{bmatrix} [e_0,p_0] \\ \vdots \\ [e_0,p_{i-1}] \end{bmatrix} \tag{3.3}$$

Por tanto, para que se satisfaga el sistema (3.2) debemos tener necesariamente

$$\alpha_i \begin{bmatrix} [p_0,p_i] \\ \vdots \\ [p_{i-1},p_i] \end{bmatrix} = 0, \quad \alpha_i = \frac{[e_0,p_i]}{[p_i,p_i]} \tag{3.4}$$

Si $\alpha_i \neq 0$ entonces $[p_j,p_i]=0$, $j<i$. Si $\alpha_i = 0$, entonces $\|e_{i+1}\| = \|e_i\|$. Si, por otro lado, $V_{i+1}=V_i$ ya hemos visto en la sección 3.1 que se alcanza la solución y por tanto $e_{i+1}=0$. Sin embargo, si $V_{i+1}\neq V_i$ hay un único (excepto múltiplos) vector $p_i \in V_{i+1}$ tal que $[p_i,z]=0$ para todo $z\in V_i$.

Resumiendo, si cada $x_{i+1}=x_0+y_i$ es elegido de tal forma que $\|e_{i+1}\|$ sea óptimal entre todos los $y_i \in V_{i+1}$, entonces $x_{i+1}-x_i=\alpha_i p_i$ donde p_i es el único vector (excepto múltiplos) tal que $[p_i,z]=0$ para todo $z\in V_i$ y $\alpha_i=[e_0,p_i]/[p_i,p_i]$. Notemos que también

$$\alpha_i = \frac{[e_i,p_i]}{[p_1,p_i]}. \tag{3.5}$$

Puesto que los p_i son conjugados dos a dos con respecto al producto interno es por lo que llamamos a este método un método de gradientes conjugados. Para que la fórmula en (3.5) sea computable será requisito elegir de forma adecuada el producto interno.

3.3. Cálculo recursivo de los p_i's

De la sección anterior, sabemos que puesto que $p_i \in V_{i+1}$ y p_i es ortogonal a V_i, $p_i = q_i(A)p_0$ donde $q_i(z)$ es un polinomio de grado exactamente i. Por tanto Ap_i es un polinomio de grado exacto $i+1$ en A por p_0, y por tanto,

$$V_{i+2} = \text{span}\{Ap_i, p_0, ..., p_i\}$$

El vector p_{i+1} puede ser calculado como

$$p_{i+1} = Ap_i - \sum_{j=0}^{i} \beta_{ij} p_j$$

Los β_{ij} quedan unívocamente determinados imponiendo las condiciones de ortogonalidad

$$[p_{i+1}, p_j] = 0, \quad j = 0, ..., i$$

que origina

$$\begin{aligned}[p_{i+1}, p_j] &= [Ap_i - \sum_{j=0}^{i} \beta_{ij} p_j, p_j] = [Ap_i, p_j] - \sum_{j=0}^{i} \beta_{ij} [p_j, p_j] \\ &= [Ap_i, p_j] - \beta_{ij} [p_j, p_j] = 0\end{aligned}$$

y, por tanto

$$\beta_{ij} = \frac{[Ap_i, p_j]}{[p_j, p_j]}, \quad j = 0, .., i$$

Si la matriz A es autoadjunta con respecto al producto interno (generalización de $A = A^*$), tenemos

$$\beta_{ij} = \frac{[p_i, A^* p_j]}{[p_j, p_j]} = \frac{[p_i, Ap_j]}{[p_j, p_j]}$$

Puesto que $p_j \in V_{j+1}$, $Ap_j \in V_{j+2}$. Como $p_i \in V_{i+1}$ es ortogonal a V_i, tenemos $\beta_{ij} = 0$ para $j < i - 1$ (para $j = i - 1 \quad Ap_{i-1} \in V_{i+1}$). Por tanto

$$p_{i+1} = Ap_i - \beta_{ii} p_i - \beta_{i,i-1} p_{i-1} \tag{3.6}$$

Reconocemos en esta expresión una relación de recurrencia de 3-términos para calcular los sucesivos p_i.

De forma similar, podemos considerar recurrencias de s términos, para

$$p_{i+1} = Ap_i - \sum_{j=i-s+2}^{i} \beta_{ij} p_j, \tag{3.7}$$

y preguntarnos para qué clase de matrices obtenemos una relación de recurrencia como ésta cualquiera que sea p_0.

Denotemos con $d(p)$ el grado del polinomio mínimo del vector p con respecto a A. Esta es la dimensión del espacio de Krylov generado por p y A. Sea $d(A)$ el grado del polinomio mínimo de A. Puesto que V_{i+1} es de dimensión $i + 1$, si $d(p_0) = i + 1$ entonces no es necesario computar p_{i+1} porque V_{i+1} contiene a la solución. Por tanto β_{ij} no necesita ser calculado, y podemos introducir la siguiente definición.

Definición 1. *Existe una iteración gradiente conjugado de s-términos para la matriz A si para todo* p_0, $[Ap_i, p_j] = 0$ *para todo* i, j *tales que* $j + s - 1 \leq i \leq d(p_0) - 2$. *Denotaremos a esta clase de matrices* $CG(s)$.

La forma más comunmente usada del método del gradiente conjugado de 3-términos aplicada a una matriz simétrica definida positiva A tiene como iteraciones:

$$x_{i+1} = x_i + \alpha_i p_i \quad r_{i+1} = r_i - \alpha_i A p_i \quad p_{i+1} = r_{i+1} + \beta_i p_i$$

Observamos que $r_i = p_i - \beta_{i-1} p_{i-1}$ y sustituyendo en la segunda ecuación:

$$r_{i+1} = p_i - \beta_{i-1} p_{i-1} - \alpha_i A p_i$$

Sustituyendo ahora r_{i+1} en la tercera ecuación obtenemos:

$$p_{i+1} = -\alpha_i A p_i + (1 + \beta_i) p_i - \beta_{i-1} p_{i-1}$$

que es la misma expresión que (3.6) a excepción de un factor escalar.

3.4. Caracterización

El producto interno $[\cdot, \cdot]$ determina la clase de bases ortogonales en el espacio. La caracterización que presentamos es con respecto a este producto interno. Consideremos A^*. Existe una matriz no singular C tal que

$$[x, y] = \langle Cx, Cy \rangle$$

donde $\langle \cdot, \cdot \rangle$ es el producto interno estándar,

$$\langle x, y \rangle = \sum_{j=1}^{N} x_i \overline{y}_i$$

Ahora,

$$[Ax, y] = \langle CAC^{-1}Cx, Cy \rangle = \langle Cx, \bar{C}^{-T} \bar{A}^T \bar{C}^T C y \rangle$$
$$\langle Cx, C(\bar{C}^T C)^{-1} \bar{A}^T \bar{C}^T C y \rangle = [x, A^* y].$$

Por tanto, $A^* = (\bar{C}^T C)^{-1} \bar{A}^T (\bar{C}^T C)$. Ya que $\bar{C}^T C$ es una matriz hermitica definida positiva, existe una matriz hermitica definida positiva B tal que $B^2 = \bar{C}^T C$. Hagamos un cambio de base tal que

$$\hat{A} = BAB^{-1}, \quad \hat{x} = Bx, \quad \text{y} \quad \hat{y} = By.$$

Entonces,

$$[Ax, y] = \langle \hat{A}\hat{x}, \hat{y} \rangle.$$

La adjunta de $\hat{A}$ con respecto a $\langle \cdot, \cdot \rangle$ es $\hat{A}^* = \bar{A}^T$. En la nueva base el problema original pasa a ser $\hat{A}\hat{x} = \hat{b}$. Una iteración del gradiente conjugado en este nuevo sistema que optimice con respecto al producto interno estándar producirá iteraciones que se corresponden con las producidas usando $[\cdot, \cdot]$. Sin embargo podría ser computacionalmente factible optimizar con respecto a $[\cdot, \cdot]$ (por ejemplo cuando C sea dispersa), pero impracticable calcular la matriz de cambio de base B, y mucho menos su inversa B^{-1}. Por conveniencia en lo que queda de capítulo usaremos sólo $\langle \cdot, \cdot \rangle$ y la definición $A^* = \bar{A}^T$, entendiendo que el cambio de base se ha llevado a cabo.

Es fácil ver que si $d(A) \leq s$, entonces $A \in CG(s)$. La condición en la definición de $CG(s)$ es obviamente cierta ya que para todo p_0, $s - 2 \geq d(p_0) - 2$. El método converge en s o menos iteraciones. El siguiente lema da otra condición suficiente para que una matriz esté en la clase $CG(s)$.

Lema 2. *Si A es tal que para todo p,*

$$A^* p \in span\{p, Ap, ..., A^{s-2}p\},$$

entonces $A \in CG(s)$.

Demostración. Consideramos

$$\langle Ap_i, p_j \rangle = \langle p_i, A^* p_j \rangle.$$

Ya que $A^* p_j = q(A)p_j$ para algún polinomio $q(z)$ de grado a lo sumo $s - 2$. Notemos que el polinomio $q(z)$ es función de p_0. Si $j + s - 2 < i$, entonces $q(A)p_j \in V_{j+s-1} \subseteq V_i$. Puesto que p_i es ortogonal a V_i entonces $\langle Ap_i, p_j \rangle = 0$ □

El siguiente lema caracteriza las matrices que satisfacen la hipótesis del lema 2. Antes de ello, recordamos algunos resultados sobre matrices normales.

Una matriz A es normal si y solo si $A^*A = AA^*$. También sabemos que A es normal si y solo si admite una base de autovectores ortonormales. Vamos a ver que también se cumple que A es normal si y sólo si $A^* = q(A)$ para algún polinomio $q(z)$.

Demostración. Primero supongamos que A es normal, sabemos que esto implica que $A = U\Lambda U^*$, donde U es unitaria y $\Lambda = diag(\lambda_1, ..., \lambda_N)$. Usando

la interpolación de Lagrange podemos construir un polinomio $q(z)$ tal que $q(\lambda_i) = \bar{\lambda_i}, \quad i = 1, ..., N$. Entonces,

$$q(A) = Uq(\Lambda)U^* = U\bar{\Lambda}U^* = A^*$$

En el otro sentido, suponiendo que existe un polinomio $q(z) = \sum_{i=0}^{N-1} c_i z^i$ tal que $q(A) = A^*$. Entonces,

$$A^*A = q(A)A = (\sum_{i=0}^{N-1} c_i A^i)A = \sum_{i=0}^{N-1} c_i A^{i+1} = Aq(A) = AA^*$$

□

Denotaremos por $n(A)$ al grado del polinomio de menor grado que satisface $A^* = q(A)$.

Lema 3. *A es tal que para todo p*

$$A^*p \in span\{p, Ap, ..., A^{s-2}p\}$$

si y sólo si A es normal y $n(A) \leq s - 2$.

Demostración. La implicación en la segunda dirección es prácticamente trivial ya que si A es normal con $n(A) \leq s - 2$ entonces

$$A^*p = q(A)p \in \text{span}\{p, Ap, ..., A^{s-2}p\}$$

Ahora en la otra dirección, supongamos que $A^*p \in \text{span}\{p, Ap, ..., A^{s-2}p\}$ para todo p. Sea v_i un autovector de A tal que $Av_i = \lambda_i v_i$. Entonces $A^*v_i \in \text{Span}\{v_i, Av_i, ..., A^{s-2}v_i\} = \text{span}\{v_i\}$, así que $A^*v_i = \mu v_i$ para algún μ. Ahora

$$\mu\langle v_i, v_i\rangle = \langle \mu v_i, v_i\rangle = \langle A^*v_i, v_i\rangle = \langle v_i, Av_i\rangle = \bar{\lambda_i}\langle v_i, v_i\rangle.$$

Por tanto, $\mu = \bar{\lambda_i}$. Ahora suponemos que A tiene un divisor elemental no lineal asociado con λ_i (es decir, un autovalor con multiplicidad geométrica mayor que 1). Entonces, hay un vector v_j tal que $(A - \lambda_i I)v_j = v_i$. Por tanto

$$\langle Av_j, v_i\rangle = \lambda_i\langle v_j, v_i\rangle + \langle v_i, v_i\rangle,$$

$$\langle Av_j, v_i\rangle = \langle v_j, A^*v_i\rangle = \langle v_j, \bar{\lambda_i}v_i\rangle = \lambda_i\langle v_j, v_i\rangle$$

Por tanto $\langle v_i, v_i\rangle = 0$, lo cual es una contradicción. Por tanto A tienen un conjunto completo de autovectores. Ahora veamos que son ortogonales dos a dos. Sean dos autovalores $\lambda_i \neq \lambda_j$, entonces

$$\lambda_i\langle v_i, v_j\rangle = \langle \lambda_i v_i, v_j\rangle = \langle Av_i, v_j\rangle = \langle v_i, A^*v_j\rangle = \lambda_j\langle v_i, v_j\rangle$$

Por tanto $\langle v_i, v_j \rangle = 0$. Ya que A tiene un conjunto completo de autovectores ortonormales, A es normal.

Veamos ahora que $n(A) \leq s-2$. Como A tiene exactamente $d(A)$ autovalores distintos, existe un polinomio interpolante $q(z)$, de grado a lo sumo $d(A)-1$ tal que $q(\lambda_i) = \bar{\lambda_i}, \quad i = 1, ..., d(A)$. Por tanto $n(A) \leq d(A) - 1$. Si $d(A) = s - 1$, entonces $n(A) \leq s - 2$.

Supongamos que $d(A) \geq s$. Tomamos p tal que $d = d(p) = d(A)$. Los vectores $\{p, Ap, ..., A^{d-1}p\}$ son linealmente independientes. Por hipótesis

$$A^*p = q(A)p \in \text{span}\{p, Ap, ..., A^{s-2}p\}$$

Por tanto $q(x)$ debe tener grado $\leq s - 2$. □

Los lemas 4 y 5 caracterizan las matrices para las que $n(A)$ es pequeño.

Lema 4. *Si A es normal y $n(A) \leq 1$, entonces $d(A) = 1$, o $A = A^*$, o*

$$A = e^{i\theta}(\frac{r}{2}I + B)$$

donde r es real y $B = -B^*$

Demostración. Si A tiene todos los autovalores reales, entonces $A^* = A$.

Veámoslo, escribiendo $A = U\Lambda U^*$, donde U es unitaria y $\Lambda = diag(\lambda_1, ..., \lambda_N)$.

Entonces $A^* = U\Lambda^* U^*$ donde $\Lambda^* = diag(\bar{\lambda_1}, ..., \bar{\lambda_N}) = diag(\lambda_1, ..., \lambda_N) = \Lambda$. Por tanto $A = A^*$.

Supongamos ahora que A tiene un autovalor complejo , digamos λ. Entonces el polinomio lineal $q(z) = az + b$ debe satisfacer

$$a\lambda + b = \bar{\lambda} \quad \text{y} \quad \bar{a}\bar{\lambda} + \bar{b} = \lambda$$

lo cuál implica $\bar{a}(a\lambda + b) + \bar{b} = \lambda$, es decir, $(\bar{a}a - 1)\lambda + (\bar{a}b + \bar{b}) = 0$. En general, solo es posible una raíz λ, ya que la ecuación es lineal, lo cuál significa que $d(A) = 1$. Si hubiera más de una raíz λ necesariamente debemos tener simultáneamente

$$\bar{a}a = 1, \quad \bar{a}b + \bar{b} = 0$$

De la segunda obtenemos equivalentemente $a\bar{b} + b = 0$ y por tanto, $a = -b/\bar{b}$. Pongamos $b = re^{i\theta}$, entonces

$$q(z) = -e^{-2i\theta}z - re^{-i\theta} = -e^{-i\theta}(ze^{-i\theta} - r)$$

Si $q(z) = \bar{z}$, entonces $q(z) = -e^{-i\theta}(ze^{-i\theta} - r) = \bar{z}$, y por tanto

$$-(ze^{-i\theta} - r) = \bar{z}e^{i\theta} = (\overline{ze^{-i\theta}}),$$

lo cuál implica

$$r = ze^{-i\theta} + (\overline{ze^{-i\theta}}).$$

Recordamos que $Re(z) = \frac{z+\bar{z}}{2}$, por tanto, si λ es un autovalor complejo de A, sustituyendo $z = \lambda$ en la ecuación anterior obtenemos que la parte real de $\lambda e^{-i\theta}$ es r/2. Esto implica que

$$B = \left(e^{-i\theta}A - \frac{r}{2}I\right)$$

tiene solo autovalores complejos imaginarios puros. Como A es normal, B es antihermítica

$$B^* = \left(e^{-i\theta}A - \frac{r}{2}I\right)^* = (e^{-i\theta}A)^* - \frac{r}{2}I = e^{i\theta}A^* + \frac{r}{2}I = -B$$

sin más que reemplazar A^* por $q(A) = -e^{-2i\theta}A + re^{-i\theta}I$. □

Lema 5. *Si A es normal entonces $n(A) \leq d(A) - 1$. Si además $n(A) > 1$, entonces $d(A) \leq n(A)^2$.*

Demostración. Ya que A es normal, tiene exactamente $d(A)$ autovalores distintos. Cómo ya hemos justificado anteriormente, usando los polinomios interpolantes de Lagrange podemos construir un polinomio de grado a lo sumo $d(A) - 1$ tal que

$$q(\lambda_i) = \bar{\lambda_i} \quad i = 1, ..., d(A).$$

Por tanto $n(A) \leq d(A) - 1$.

Ahora supongamos que $n(A) > 1$, ¿Cuántos números complejos distintos satisfacen $q(\lambda) = \bar{\lambda}$? Notemos que

$$\bar{q}(\bar{\lambda}) = \lambda, \quad \text{es decir} \quad \bar{q}(q(\lambda)) = \lambda.$$

Aplicando el teorema fundamental del álgebra por el que sabemos que un polinomio de grado n tiene, contando multiplicidades, exactamente n raíces complejas. Ya que q no es lineal, $\bar{q}(q(\lambda)) - \lambda = 0$ tiene como mucho $n(A)^2$ raíces. Por tanto, $d(A) \leq n(A)^2$. □

Hemos demostrado que $d(A) \leq s$ o A normal y $n(A) \leq s - 2$ son condiciones suficientes para que $A \in CG(s)$. Lo que queda por establecer es que estas

condiciones son también necesrias para $A \in CG(s)$, que es la parte técnicamente más difícil de la prueba del teorema de Faber-Manteuffel. Primero estableceremos algunos resultados que ayudarán a simplificar la prueba.

Definimos el conjunto de vectores:

$$D = \{p : d(p) < d(A)\}$$

Este conjunto es una unión de subespacios de dimensión menor o igual a $N-1$, en que cada subespacio está generado por un divisor del polinomio minimal de A. Por tanto, hay un numero finito de tales subespacios. El conjunto D es por tanto un conjunto de medida cero bajo la topología derivada del producto interno.

Consideramos ahora el cálculo de la base ortogonal del espacio de Krylov descrita en el apartado 3.3. Dado un vector p, pongamos

$$\begin{aligned} &p_0 = p, \\ &p_1 = Ap_0 - \beta_{00}p_0, \\ &... \\ &p_{i+1} = Ap_i - \sum_{j=0}^{i} \beta_{ij}p_j \quad \text{donde} \quad \beta_{ij} = \frac{\langle Ap_i, p_j\rangle}{\langle p_j, p_j\rangle} \end{aligned}$$

Si $d(p) > i$, entonces p_i puede ser considerado como una función de p. Vamos a extender esta definición a todo p poniendo $p_i = 0$ cuando p tiene grado $d(p) \leq i$. Demostraremos que para todo $i \leq d(A)$, p_i es una función continua de p.

Lema 6. *Con la definición dada de p_i, entonces para todo $i \leq d(A)$, p_i es una función continua de p y $\|p_i\| \leq \|A\| \cdot \|p_{i-1}\|$.*

Demostración. Lo probamos por inducción. Claramente, $p_0 = p$ es continua. Ahora consideramos $p_1 = Ap_0 - \beta_{00}p_0$. Para $p_0 \neq 0$, tenemos

$$|\beta_{00}| = \left|\frac{\langle Ap_0, p_0\rangle}{\langle p_0, p_0\rangle}\right| \leq \|A\|$$

Por tanto, para $p_0 \neq 0$, β_{00} es continua y acotada. Puesto que el espacio sobre el que $p_0 = 0$ es cerrado y de menor dimensión que el espacio total, podemos extender con continuidad el producto $\beta_{00}p_0$ a todo el espacio. Por último, es

claro que

$$\begin{aligned}\|p_1\|^2 = \langle p_1, p_1\rangle &= \langle Ap_0 - \beta_{00}p_0, Ap_0 - \beta_{00}p_0\rangle \\ &= \|Ap_0\|^2 - 2\frac{|\langle Ap_0, p_0\rangle|^2}{\langle p_0, p_0\rangle} + \frac{\langle |Ap_0, p_0\rangle|^2}{\langle p_0, p_0\rangle^2}\langle p_0, p_0\rangle \\ &= \|Ap_0\|^2 - \frac{|\langle Ap_0, p_0\rangle|^2}{\langle p_0, p_0\rangle} \leq \|A\|^2\|p_0\|^2\end{aligned}$$

Supongamos ahora que para $j \leq i < d(A)$, p_j es una función continua de p y $\|p_j\| \leq \|A\| \cdot \|p_{j-1}\|$. Tenemos entonces

$$p_{i+1} = Ap_i - \sum_{j=0}^{i} \beta_{ij}p_j$$

Consideramos

$$\beta_{ij} = \frac{\langle Ap_i, p_j\rangle}{\langle p_j, p_j\rangle}$$

Entonces primero aplicando la desigualdad de Cauchy-Schwarz y después la hipótesis de inducción, para $p_j \neq 0$:

$$\begin{aligned}|\beta_{ij}| &\leq \frac{\|Ap_i\|\|p_j\|}{\|p_j\|^2} \leq \frac{\|A\|\|p_i\|\|p_j\|}{\|p_j\|^2} \leq \frac{\|A\|\|p_i\|}{\|p_j\|} \\ &\leq \frac{\|A\|\|p_{i-1}\|}{\|p_{j-1}\|} \leq ... \leq \frac{\|A\|\|p_{i-j}\|}{\|p_0\|} \leq \|A\|^{i-j}\end{aligned}$$

Para $p_j \neq 0$, β_{ij} es una función continua y acotada de p_j, y por tanto una función continua y acotada de p. Puesto que $j < d(A)$, el conjunto para el cuál $p_j = 0$ es cerrado y de menor dimensión que la del espacio total, esto último concretamente recordando que $p_j = q(A)p_0$ con $q(z)$ un polinomio de grado j. Como $j < d(A)$ con $d(A)$ el grado del polinomio mínimo entonces el espacio $\{p(A) = 0\}$ es de menor dimensión que el espacio total ya que sino el polinomio mínimo tendría dimensión menor o igual que j lo cuál es absurdo por hipótesis.

Por tanto nosotros podemos extender por densidad el producto $\beta_{ij}p_j$ al espacio total. Ya que p_{i+1} es una suma de funciones continuas de p, es continua. Ahora

$$\|p_{i+1}\|^2 = \|Ap_i\|^2 - \sum_{j=0}^{i} \frac{|\langle Ap_i, p_j\rangle^2}{\langle p_j, p_j\rangle} \leq \|A\|^2\|p_i\|^2$$

Esto completa la prueba. □

Llegamos ahora al resultado principal:

Teorema. *$A \in CG(s)$ si y sólo si $d(A) \leq s$ o A es normal y $n(A) \leq s-2$.*

Demostración. La suficiencia ya ha sido demostrada anteriormente. Supongamos entonces que $A \in CG(s)$ y que $d(A) > s$ y veamos que A es normal y $n(A) \leq s-2$. Sea p cualquier vector tal que $d(p) > s$. Por la definición de $CG(s)$ sabemos que

$$\langle Ap_i, p\rangle = \langle p_i, A^*p\rangle = \langle p_i, A^*p_0\rangle = 0, \quad s-1 \leq i \leq d(p)-2$$

En particular, $\langle p_{s-1}, A^*p\rangle = 0$. Sea $F(p) = \langle p_{s-1}, A^*p\rangle$ considerada como una función de p. Cómo vimos en el lema anterior p_{s-1} es una función continua de p, y entonces $F(p)$ es una función continua de p. Tenemos que $F(p) = 0$ para todo p tal que $d(p) > s$. Puesto que $d(A) > s$, el conjunto para el cual $d(p) \leq s$ es cerrado y de menor dimensión que la del espacio total. Por tanto $F(p) = 0$ para todo p. Veámoslo: sólo nos queda ver que es cierto para $d(p) \leq s$. Es trivial para $d(p) < s$ ya que $p_{s-1} = 0$ para estos p. Sin embargo, si $d(p) = s$, $p_{s-1} \neq 0$.

Sea p tal que $d(p) = s$ y sea

$$V_s = \text{span}\{p, Ap, ..., A^{s-1}p\}$$

el subespacio invariante generado por p. Sea $\tilde{A}$ la restricción de A a V_s. Si Q es una proyección ortogonal sobre V_s, entonces

$$\tilde{A} = AQ \quad \text{y} \quad \tilde{A}^* = QA^*.$$

Demostraremos ahora que $\tilde{A}$ es normal sobre V_s. Tenemos que $d(\tilde{A}) = s$. Sea p cualquier generador de V_s, es decir, $p \in V_s$ tal que $d(p) = s$. Ya hemos visto que

$$F(p) = \langle p_{s-1}, A^*p\rangle = \langle p_{s-1}, QA^*p\rangle = \langle p_{s-1}, \tilde{A}^*p\rangle = 0.$$

Puesto que $\tilde{A}^*p \in V_s$, debemos tener

$$\tilde{A}^*p \in \text{span}\{p_0, ..., p_{s-2}\} = \text{span}\{p, Ap, .., A^{s-2}p\},$$

para todo $p \in V_s$ tal que $d(p) = s$. Ahora consideramos

$$W(p) = p \wedge Ap \wedge ... \wedge A^{s-2}p \wedge \tilde{A}^*p,$$

donde $\wedge$ es el producto exterior. Dado que el producto exterior es una función multilineal de un espacio vectorial a otro, $W(p)$ es una función continua de p. Para todo $p \in V_s$ tal que $d(p) = s$, $W(p) = 0$ debido a que los vectores son

linealmente dependientes. El conjunto $\{p \in V_s : d(p) < s\}$ es un subconjunto cerrado de menor dimensión, por tanto $W(p) = 0$ para todo $p \in V_s$. Esto es trivial para aquellos p con $d(p) < s-1$ ya que los primeros $s-1$ vectores son linealmente dependientes. Sin embargo, si $p \in V_s$ con $d(p) = s-1$, los primeros $s-1$ vectores son independientes. Esto implica que si $d(p) = s-1$,

$$\tilde{A}^* p \in \text{span}\{p, Ap, .., A^{s-2}p\}.$$

Sea ahora $V_{s-1}^{(1)} \subseteq V_s$ un subespacio invariante de dimensión s-1. Como $d(\tilde{A}) = s$, $V_{s-1}^{(1)}$ debe ser generado por un vector p con $d(p) = s-1$. Esto implica que $V_{s-1}^{(1)}$ tiene una base, llamémosla $b_1, ..., b_{s-1}$ tal que $d(b_i) = s-1, i = 1..., s-1$. De lo anterior sabemos que

$$\tilde{A}^* b_i \in V_{s-1}^{(1)}, \quad i = 1, ..., s-1.$$

Por tanto, $V_{s-1}^{(1)}$ es también un subespacio invariante de $\tilde{A}^*$.

Sea $q_1 \in V_s$ el vector único ortogonal a $V_{s-1}^{(1)}$. Tenemos entonces que

$$\langle Aq_1, y\rangle = \langle q_1, \tilde{A}^* y\rangle = 0$$

para todo $y \in V_{s-1}^{(1)}$. Por tanto $Aq_1 = \lambda_1 q_1$ para algún λ_1. Ahora sea $V_{s-1}^{(2)} \subseteq V_s$ cualquier subespacio invariante de dimensión $s-1$ tal que $q_1 \in V_{s-1}^{(2)}$. Sea $q_2 \in V_s$ el único vector ortogonal a $V_{s-1}^{(2)}$. Usando el mismo argumento que antes llegamos a que $Aq_2 = \lambda_2 q_2$ para algún λ_2 y $\langle q_1, q_2\rangle = 0$. Si continuamos con esta dinámica, vemos que V_s tiene una base formada por autovectores ortonormales de A. Mostramos ahora que de hecho A tiene un conjunto completo de autovectores ortonormales. Supongamos que existe v tal que para algún λ_i

$$(A - \lambda_i I)v \neq 0, \quad (A - \lambda_i I)^2 v = 0.$$

Sean v y $(A-\lambda_i I)v$ dentro de un subespacio invariante, V, con $s-2$ autovectores distintos asociados. Sabemos del argumento anterior que hay al menos s de tales autovectores. V es un subespacio invariante de dimensión s. Sea $\tilde{A}$ la restricción de A a V. Tenemos que $d(\tilde{A}) = s$. Repitiendo el argumento anterior vemos que V contiene s autovectores ortogonales de A, lo cual es una contradicción.

Similarmente, si asumimos que q_i y q_j son autovectores de A asociados a autovalores distintos, podemos incluirlos en un subespacio invariante de dimensión s sobre el cual $d(\tilde{A}) = s$. El argumento anterior permitiría concluir que q_i es ortogonal a q_j.

Queda pues establecido que A es normal. Esto implica que existe un polinomio $q(z)$ tal que $A^* = q(A)$, y recordemos que $n(A)$ es el grado del polinomio de grado mínimo que satisface esto. Del lema 5 sabemos que $n(A) \leq d(A)-1$. Veamos ahora que de hecho $n(A) \leq s-2$. De la definición de $CG(s)$ sabemos que si $d(p) = d(A)$ entonces

$$F_i(p) = \langle p_i, A^*p \rangle = 0, \quad s-1 \leq i \leq d(A)-2 \tag{3.8}$$

Ahora, $F_i(p)$ es una función continua de p y por tanto $F_i(p) = 0, s-2 \leq i \leq d(A)-2$ para todo p. Si $d(p) < d(A)$, entonces de (3.8)

$$A^*p \in \text{span}\{p_0, p_1, ..., p_{s-2}\} = \text{span}\{p, ..., A^{s-2}p\}.$$

Ahora sea $d(p) = d(A)-1$. El polinomio que anula p es el producto de todos a excepción de un factor. Sea $d = d(A)$,

$$p_k(z) = \prod_{i \neq k}(z - \lambda_i) = z^{d-1} - \left(\sum_{j \neq k} \lambda_i\right) z^{d-1} + ... + ...,$$

y pongamos

$$q(z) = \gamma_{d-1}z^{d-1} + \gamma_{d-2}z^{d-2} + ... + ...$$

Sabemos que si $d(p) = d(A) - 1$, entonces los vectores $\{p, ..., A^{d-2}p\}$ son linealmente independientes. Ahora $A^*p = q(A)p = \gamma_{d-1}A^{d-1}p + \gamma_{d-2}A^{d-2}p + ...$ Como $p_k(A)p = 0$, sabemos

$$A^{d-1}p = \left(\sum_{j \neq k} \lambda_i\right) A^{d-2}p + ... + ...$$

Por tanto,

$$A^*p = \left(\gamma_{d-1}\left(\sum_{j \neq k} \lambda_i\right) + \gamma_{d-2}\right) A^{d-2}p + ... + ...,$$

pero $A^*p \in \text{span}\{p, ..., A^{s-2}p\}$ implica

$$\left(\gamma_{d-1}\left(\sum_{j \neq k} \lambda_i\right) + \gamma_{d-2}\right) = 0.$$

Puesto que esto es verdad para todo k, debemos tener $\gamma_{d-1} = \gamma_{d-2} = 0$. Si $\gamma_{d-1} = 0$, entonces debemos tener $\gamma_i = 0$, para todo $i > s-2$ ya que $\{p, ..., A^{d-2}p\}$ son linealmente independientes y $A^*p \in \{p, ..., A^{s-2}p\}$. Por tanto $q(z)$ es de grado menor o igual a $s-2$. □

Estos resultados dependen de la elección de un producto interno, el cual implica un cambio de base. El teorema principal implica que existe un producto interno para el cual, por ejemplo, $A \in CG(3)$ si y solo si $d(A) \leq 3$ o si A tiene un conjunto completo de autovectores y autovalores que se hallan en alguna línea recta del plano complejo. Sin embargo la determinación del producto interno puede ser muy difícil.

Capítulo 4

Aplicación de dichos algoritmos en matrices test

En este capítulo discutiremos los resultados de comparaciones experimentales llevadas a cabo sobre las formas básicas de los algoritmos CG, GMRES(m) y MINRES. Nuestro objetivo no es identificar el mejor algoritmo ya que cada uno de ellos es preferible para matrices con unas ciertas propiedades, pero si intentaremos, si es posible, determinar las características tanto de algoritmos como de matrices que dan lugar a comportamientos particulares.

Hemos escrito nuestro propio software, usando MATLAB, para no añadir diferencias que puedan venir de diferentes implementaciones para algunas operaciones auxiliares, como por ejemplo reducir una matriz Hessenberg superior a triangular superior. Por tanto cualquier diferencia en el rendimiento es debida únicamente a las propiedades básicas de los métodos de Krylov a menos que se indique lo contrario. Además hemos testeado nuestras implementaciones contra aquellas funciones públicamente disponibles por MATLAB para verificar así el correcto funcionamiento de nuestros programas. Comparamos así nuestras versiones de CG, GMRES(m) y MINRES con las de MATLAB para varias matrices test y podemos afirmar que concuerdan. La efectividad de estos métodos, incluído el gradiente conjugado, mejora cuando la matriz del sistema se preprocesa para mejorar su número de condición. Este proceso es comúnmente llamado de preacondicionamiento, y en la siguiente sección abordamos unas ideas generales sobre el mismo y su implementación basada en factorizaciones LU, o LL^T, incompletas.

4.1. Precondicionamiento

4.1.1. Ideas generales

Como hemos visto en nuestra discusión sobre los distintos métodos basados en subespacios de Krylov, en ocasiones estos no son robustos en el sentido de poder garantizar una solución aproximada aceptable dentro de un tiempo de computación y almacenamiento modestos. Para algunos métodos como GMRES es obvio que, en aritmética exacta, llegan a la solución en un máximo de n iteraciones, pero esto podría no ser muy práctico en la realidad si n es excesivamente grande. Otros métodos están restringidos a una clase especifica de problemas, como serían CG y MINRES, o sufren de problemas como estancamiento (si la norma del residuo no decrece suficientemente rápido a cero) o *breakdown* (cuando resulta una división por cero). En general, la convergencia lenta a la solución depende en una manera muy compleja de las propiedades espectrales de A: la distribución de los autovalores de A, el campo de valores, etc. En muchas situaciones reales no tenemos información disponible de estas propiedades.

Es en esta situación donde surge la idea del precondicionamiento, que se basa en intentar encontrar una matriz K tal que $K^{-1}A$ tenga mejores propiedades espectrales de forma que se tenga convergencia rápida de estos métodos cuando se aplican al sistema modificado $K^{-1}A = K^{-1}b$. Esto se apoya en la observación de que para $K = A$, tendríamos el sistema ideal $K^{-1}Ax = Ix = K^{-1}b$ y todos los métodos alcanzarían la solución exacta en un único paso. Por tanto, buscamos una aproximación K de A de manera que el correspondiente método de Krylov aplicado a $K^{-1}Ax = K^{-1}b$ sólo necesite unas pocas iteraciones para alcanzar una buena aproximación de la solución del sistema. La matriz K en este contexto recibe el nombre de precondicionador para la matriz A.

El problema de encontrar un precondicionador eficiente, es aquel que trata de identificar una matriz K con las siguientes propiedades:

- K es una buena aproximación de A
- El costo por construir K no es restrictivo
- El sistema $Ky = z$ es mucho más fácil de resolver que el sistema original $Ax = b$.

No existe una teoría general para la búsqueda de un precondicionador con la cual nosotros podamos asegurarnos que hemos hecho una elección eficiente. La principal dificultad es que el precondicionamiento está basado en una

aproximación y en ausencia de información precisa del comportamiento de la solución del sistema $Ax = b$ y de las propiedades espectrales de A, la convergencia podría depender criticamente de la información que hemos descartado en el proceso de aproximación.

Excepto para algunas soluciones triviales (matriz diagonal), la matrix $K^{-1}A$ no es formada explícitamente. En muchos casos esto nos dejaría una matriz densa y destrozaría toda la eficiencia que podría ser obtenida para las matrices dispersas A que utilizamos frecuentemente en estos métodos. Incluso para una matriz A densa podría ser demasiado caro formar la matriz precondicionada explícitamente. En vez de eso procedemos de la siguiente forma: para cada aplicación de $K^{-1}A$ sobre algún vector y, primero computamos w como el resultado del operador A aplicado a y ($w = Ay$), para seguidamente determinar el resultado z del operador K^{-1} aplicado a w mediante la resolución del sistema $Kz = w$. Otras estrategias construyen una aproximación M de A^{-1} y entonces basta aplicar el operador M a w para obtener z.

El mismo precondicionador puede ser implementado de diferentes formas. Esto no cambia los autovalores de la matriz precondicionada pero sí los autovectores, lo cual puede repercutir en el comportamiento de la convergencia. Existen tres implementaciones diferentes:

1. Precondicionamiento por la izquierda.

   Aplicamos el método iterativo a

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

   Notemos que la simetría de A y K no implica la simetría de $K^{-1}A$. Sin embargo si K es simétrica definida positiva entonces $[x, y] \equiv (x, Ky)$ define un producto interno adecuado. Es fácil verificar que $K^{-1}A$ es simétrica respecto a este nuevo producto interno, y por tanto podemos usar métodos como MINRES y CG.

   Si usamos un método que minimice la norma del residuo (como GMRES o MINRES) debemos tener en cuenta que estamos minimizando el residuo precondicionado $K^{-1}(b - Ax_k)$, el cual podría ser bastante diferente del residuo $b - Ax_k$. Esto podría tener consecuencias si el criterio de parada está basado en la norma del residuo.

2. Precondicionamiento por la derecha. Aplicamos el método iterativo a

   $$AK^{-1}y = b \quad \text{con} \quad x = K^{-1}y$$

   Esta forma de precondicionamiento tampoco deja un producto simétrico aunque A y K sean simétricas. En este caso debemos tener cuidado

con el criterio de parada basandonos en que el error $\|y - y_k\|_2$ podría ser mucho más pequeño que el error $\|x - x_k\|_2 = \|K^{-1}(y - y_k)\|_2$, siendo este último el que nosotros estamos interesados. Este precondicionamiento tiene la ventaja de que sólo afecta al operador y no al lado derecho del sistema, lo cuál podría ser atractivo en diseño de software para aplicaciones especificas.

3. Precondicionamiento por ambos lados.

   Para un precondicionador K con $K = K_1K_2$, el método iterativo puede ser aplicado a

   $$K_1^{-1}AK_2^{-1}z = K_1^{-1}b \quad \text{con} \quad x = K_2^{-1}z$$

   Esta forma de precondicionamiento podría ser útil para aquellos precondiconadores que vienen en forma factorizada o para obtener un operador (casi) simétrico cuando K no pueda ser usada para la definición de producto interno (es decir, K no es simétrica definida positiva).

4.1.2. Factorización LU incompleta (ILU)

Consideramos el sistema lineal disperso $Ax = b$, con A una matriz $n \times n$. Con fecuencia este es resuelto utilizando eliminación Gaussiana, que es equivalente a factorizar la matriz A como $A = LU$ donde L es una matriz triangular inferior con unos en la diagonal y U una matriz triangular superior. El principal problema de estos cálculos en matrices dispersas, cuando n es grande, es que durante el proceso de eliminación se destruyen ceros de la matriz A causando que los factores L y U de la factorización frecuentemente sean mucho menos dispersos, o incluso densos, aumentando el costo computacional de la solución, e incluso las necesidades de memoria. La idea básica detrás del precondicionador ILU es modificar la eliminación Gaussiana permitiendo estas perdidas de ceros (fill-in) sólo en un conjunto de posiciones previamente seleccionado.

Sean estas posiciones de relleno dadas *a priori* por el conjunto de indices $\mathcal{S}$, es decir,

$$\begin{aligned} \mathcal{S} \subset & \mathcal{S}_n \equiv \{(i,j), 1 \leq i \leq n, 1 \leq j \leq n\} \\ & l_{i,j} = 0 \; si \; j > i \; o \; (i,j) \notin \mathcal{S} \\ & u_{i,j} = 0 \; si \; i > j \; o \; (i,j) \notin \mathcal{S} \end{aligned} \tag{4.1}$$

Una estrategia comúnmente usada es definir $\mathcal{S}$ como:

$$\mathcal{S} = \{(i,j) \mid a_{i,j} \neq 0\} \tag{4.2}$$

que corresponde a modificar en el proceso de la eliminación Gaussiana únicamente los elementos que son inicialmente no nulos, manteniendo las propiedades de dispersión de A. Notemos que tener a priori el conjunto de los elementos que vamos a ignorar durante el proceso no es una gran restricción, ya que durante el mismo podríamos ignorar los elementos en función de un criterio que definiría dichas posiciones implícitamente.

Definición 2. *Sea $A = (a_{i,j})$ una matriz $n \times n$, $\mathcal{S} \subset \mathcal{S}_n$ el patrón de dispersión escogido, entonces definimos la descomposición incompleta LU como $A = \tilde{L}\tilde{U} - N$ donde $\tilde{L} = (\ell_{i,j})$ con $\ell_{i,i} = 1$ es una matriz triangular superior, $\tilde{U} = (u_{i,j})$ una matriz triangular superior y $N = (N_{i,j})$ tales que*

- $\ell_{i,j} = 0$,$u_{i,j} = 0$, *si* $(i,j) \notin \mathcal{S}$
- $n_{i,j} = 0$ *si* $(i,j) \in \mathcal{S}$

Puede demostrase la estabilidad de esta factorización incompleta si la matriz A es una M-matriz. Una matriz real $n \times n$, $A = (a_{ij})$, con $a_{ij} \leq 0$ para todo $i \neq j$ es una M-matriz si y sólo si A es no singular $A^{-1} \geq 0$. Este tipo de matrices son comunes en la discretización de ecuaciones en derivadas parciales. Básicamente podemos identificarlas como aquellas que al aproximar el precondicionador K con una factorización obtenemos un método iterativo convergente. Para ver más sobre la teoría sobre las M-matrices para métodos iterativos remitimos [7].

Algoritmo ILU:

```
1  Sea A una matriz n × n
2  for k = 1, 2, ...n − 1
3     d = 1/a_{k,k}
4     for i = k + 1, k + 2, ..., n
5        if (i, k) ∈ S
6           e = d a_{i,k}; a_{i,k} = e
7           for j = k + 1, ..., n
8              if (i, j) ∈ S y (k, j) ∈ S
9                 a_{i,j} = a_{i,j} − e a_{k,j}
10             end
11          end
12       end
13    end
14 end
```

4.1.3. Incomplete Cholesky

Cuando la matriz A es simétrica definida positiva entonces obviamente seleccionamos $\mathcal{S}$ de manera que defina un patrón de dispersión simétrico y

así las diagonales de L y U sean iguales. Así obtenemos la factorización incompleta de Cholesky, que no es más que una aproximación dispersa de la factorización de Cholesky. Ciertamente ésta es usada con frecuencia como precondicionador en algoritmos como el del gradiente conjugado.

La factorización de Cholesky de una matriz simétrica definida positiva A es $A = LL^T$ donde L es una matriz triangular inferior. Una factorización incompleta de Cholesky esta dada por matrices triangulares inferiores dispersas K que son en cierto modo aproximaciones de L: el correspondiente precondicionador es KK^T. De igual forma que para ILU, una manera popularmente usada para encontrar tal matriz K es usar el mismo algoritmo que para la descomposición de Cholesky, excepto para aquellos elementos en A que ya sean cero. Esto nos da una factorización incompleta de Cholesky, que es dispersa como la matriz A.

Un algoritmo equivalente al 4.1.2 puede ser construido, a rasgos generales es lo siguiente:

Algoritmo ICHOL:

```
1  for  i = 1, 2, ...n
2     L_ii = (a_ii − Σ_{k=1}^{i−1} L_ik^2)^{1/2}
3     for  j = i + 1, ..., N
4           L_ji = (1/L_ii)(a_ji − Σ_{k=1}^{i−1} L_ik L_jk)  if  (j, i) ∈ S
5     end
6  end
```

4.2. Presentación de resultados

Los test que hemos llevado a cabo consisten en lo siguiente. Primero hemos adjuntado las imágenes del patrón de dispersión de las matrices utilizadas. Por otro lado, hemos mostrado los resultados gráficamente pintando el residuo relativo $\|b - Ax_i\|_2/\|b\|_2$ en el eje de ordenadas frente al número de iteración en el eje de abscisas. El total de iteraciones está determinado por la tolerancia de parada sobre el residuo relativo, 10^{-14}. Con esto pretendemos mostrar las diferencias en el comportamiento de particulares algoritmos con respecto a la convergencia. Además también presentamos en forma tabular los resultados obtenidos por los algoritmos al cambiar diferentes parámetros de entrada sobre un conjunto de matrices test. En dichas tablas indicamos la dimensión de la matriz por n, el número de elementos distintos de zero como Nz, el número de condición de la matriz por κ, la tolerancia exigida a la solución aproximada y mostramos, para cada algoritmo y matriz, el número

de iteraciones necesarias para alcanzar esta tolerancia.

Las matrices test han sido escogidas en función del algoritmo, ya que cómo es lógico para el Gradiente Conjugado hemos utilizado matrices simétricas definidas positivas, para GMRES matrices no simétricas y para MINRES simétricas indefinidas.

Además de las propiedades de simetría y del caracter definido positivo, la elección de las matrices también ha considerado el numero de condición de las mismas, y en el caso de GMRES las propiedades de normalidad y simetría. Más adelante en la sección dedicada a este algoritmo explicaremos los criterios usados para medir estas propiedades. Todas las matrices provienen de la colección Harwell-Boeing basadas en problemas reales y obtenidas a través de Matrix Market, un depósito de matrices test para la comparación de algoritmos del álgebra lineal numérica (`https://math.nist.gov/MatrixMarket/`). Los números de condición varian desde 8.8 (matriz `bcsstm02`) a 2.8 $\times 10^{11}$ (matriz `bcsstk19`) y forman parte de los datos proporcionados por la colección.

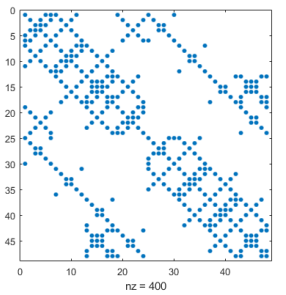
Aunque es posible aplicar los algoritmos a las matrices de los problemas originales, veremos que se obtienen mejores resultados precondicionando las matrices. Para el caso del Gradiente Conjugado usamos la factorización incompleta de Cholesky y para GMRES simplemente escalamos tanto la matriz de coeficientes como el lado derecho del sistema de ecuaciones (podría considerarse como un precondicionamiento diagonal).

Hemos encontrado dos tipos de fallo al evaluar el comportamiento de los algoritmos. El primero es la inestabilidad que conduce a over- o underflow [1], generalmente esto se produce al dividir por cero o por una cantidad muy pequeña. En este caso las normas de los residuos relativos oscilan violentamente. Ejemplos de inestabilidad pueden ser obervados en la primera gráfica de la figura 4.2. En las tablas denotaremos este fallo por I. El segundo tipo de error, denotado por S, es el estancamiento. Las normas residuales se mantienen constantes indefinidamente a partir de cierta iteración, en nuestro caso hasta alcanzar el número máximo de iteraciones permitido o hasta que el programa se detiene por overflow o underflow. Si un algoritmo alterna periodos de estancamiento e inestabilidad en las tablas lo denotamos por I/S.

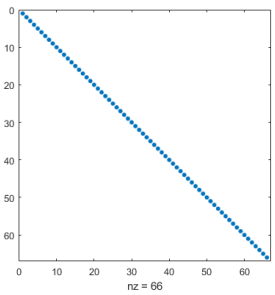
[1] Overflow y underflow son errores producidos por la limitación de bits en la mantisa del sistema coma flotante.

4.3. Gradiente Conjugado

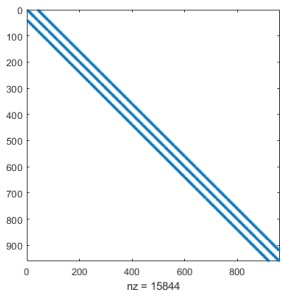
Para el algoritmo del gradiente conjugado todas las matrices son simétricas definidas positivas como ya hemos indicado anteriormente. En este caso los números de condición oscilan desde 8.8 (matriz `bcsstm02`) a 6.1×10^9 (matriz `bcsstm25`). Las matrices seleccionadas y sus diagramas de dispersión son mostrados a continuación:



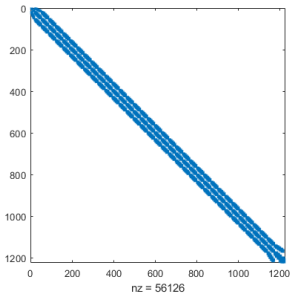
(a) bcsstk01



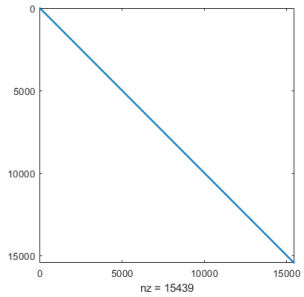
(b) bcsstm02



(c) nos3



(d) bcsstk27



(e) bcsstm25

Figura 4.1: Matrices test para el gradiente conjugado.

Primero aplicamos los algoritmos a las matrices originales y después precondicionamos utilizando la factorización incompleta de Cholesky. Se ha implementado también en MATLAB el algoritmo del gradiente conjugado precondicionado y ha sido comparado con la versión que implementa MATLAB en la función `pcg()`. En nuestra versión obtenemos el precondicionador utilizando la función de matlab `ichol()`, que dada la matriz A proporciona la factorización incompleta de Cholesky de la misma.

Problema	bcsstk01	bcsstm02	nos3	bcsstk27	bcsstm25
n	48	66	960	1224	15439
Nz	400	66	15844	56126	15439
κ	1.6×10^{6}	8.8	7.3×10^{4}	7.7×10^{4}	6.1×10^{9}
Tolerancia	10^{-14}	10^{-14}	10^{-14}	10^{-14}	10^{-14}
GC sin prec	162 (I)	12	323	1700 (I)	I/S
GC con prec	21	1	61	32	1

Cuadro 4.1: Matrices y resultados para el gradiente conjugado sin y con precondicionamiento

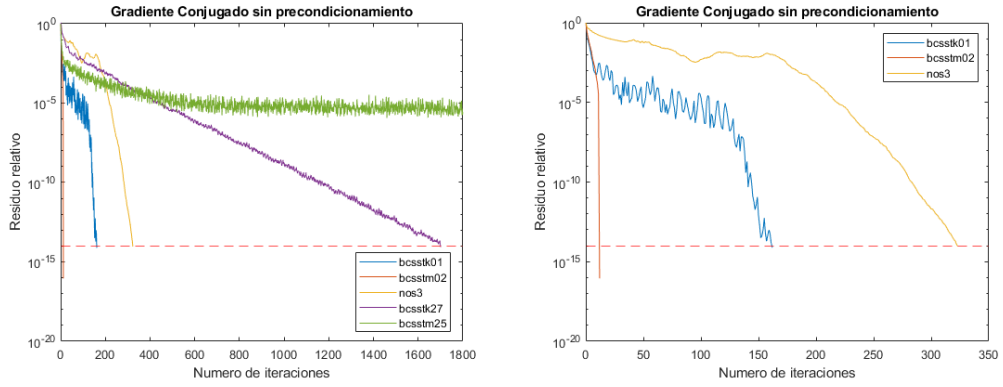


Figura 4.2: Gradiente conjugado sin precondicionamiento con una tolerancia de 10^{-14}. En la segunda imagen hemos ampliado para poder visualizar mejor el comportamiento de aquellas matrices que convergen en menor número de iteraciones. En este caso para las gráficas hemos utilizado sólo líneas y no líneas y puntos para mejorar la visibilidad del fenómeno de inestabilidad.

Analizando los resultados sin precondicionamiento observamos que el sistema con matriz de coeficientes `bcsstm02` es el que mejor resultados obtiene alcanzando la tolerancia deseada en sólo 12 iteraciones. De hecho podemos observar el fenómeno de convergencia superlineal mencionado en el apartado 2.2.3. Esta es la matriz con menor número de condición con diferencia. Aunque con la matriz `bcsstk01` se converge en 163 iteraciones se observa el fenómeno de inestabilidad produciéndose oscilaciones en la norma residual de hasta 10^{2}. Con la matriz `bcsstk27` también observamos cierta inestabilidad que afecta también a la velocidad de convergencia siendo el segundo resultado

más lento. Vemos que en el único caso en el que se alcanza el número máximo de iteraciones es para `bcsstm25`, la matriz con peor número de condición: se producen ambos fenómenos, tanto inestabilidad como estancamiento.

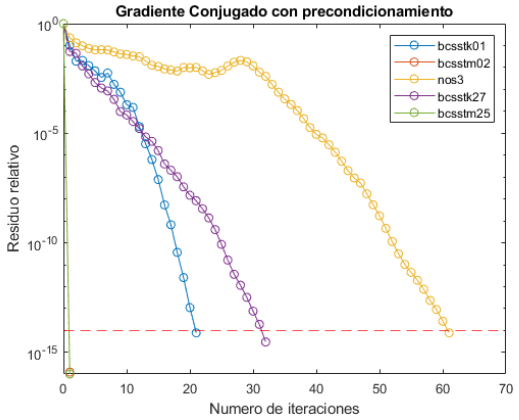
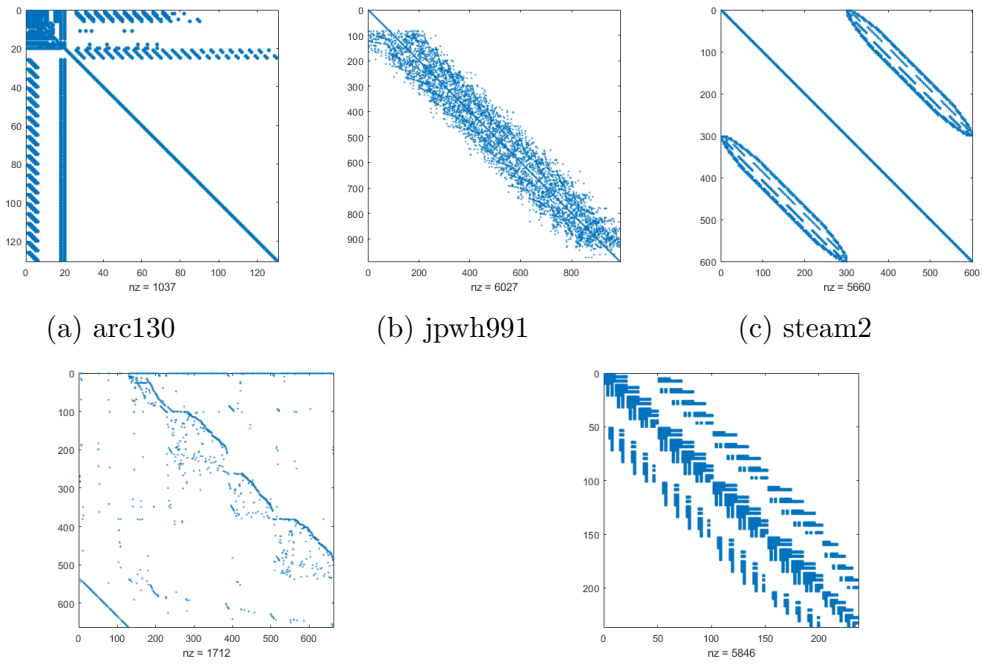


Figura 4.3: Gradiente conjugado con precondicionamiento con una tolerancia de 10^{-14}.

Observamos que al aplicar precondicionamiento a la matriz mejoran mucho los resultados en líneas generales. Aumenta la velocidad de convergencia en todos los algoritmos disminuyendo así el número de iteraciones necesarias para alcanzar la tolerancia deseada, consiguiéndolo en una sola iteración en el caso de `bcsstm02` y `bcsstm25`. Desaparece el fenómeno de inestabilidad en las matrices en las que antes se producía. También podemos observar el fenómeno de convergencia lineal mencionado en un subapartado de la sección 2.2.3.

4.4. GMRES

Para este algoritmo tenemos que seleccionar matrices no simétricas y aparte de basarnos en el número de condición publicado en el Matrix Market para su elección, también nos hemos guiado por su normalidad y simetría. Las matrices escogidas y sus diagramas de dispersión son los siguientes:



(a) arc130 (b) jpwh991 (c) steam2

(d) shl400 (e) e05r0500

Figura 4.4: Matrices test para GMRES

No ha sido sencillo escoger las matrices de acuerdo a su normalidad ya que esta no está publicada en los datos de Harwell-Boeing, quizás porque no existe un acuerdo general para medir esta característica. La medida que usamos en nuestro proceso de selección fue

$$\nu = \frac{\sqrt{\sum_{i=1}^{n} |\lambda_i|^2}}{\|A\|_F}.$$

Esta variable toma valores entre cero y uno, el primero implica que la matriz es nilpotente mientras que uno caracteriza las matrices normales. Usando esta medida la normalidad de las matrices seleccionada varía desde 2.6×10^{-5} hasta 0,999998.

En el caso de la simetría, como hemos explicado nos interesa medir la no simetría de dichas matrices y aunque parezca sorprendente tampoco hay una medida sobre esto en la colección Harwell-Boeing. Definimos la simetría de la matriz A como

$$\sigma = \frac{\|S\|_F}{\|A\|_F}$$

donde $S = \frac{1}{2}(A + A^T)$ y el subíndice F se refiere a la norma matricial de Frobenius. Los valores de σ oscilan entre 0 y 1 de nuevo, significando los valores extremos que la matriz es antisimétrica y simétrica respectivamente. Usando esta definición las matrices seleccionadas para GMRES oscilan entre 0.4708 (matriz `e05r0500`) y 0.999999 (matriz `steam2`).

Aunque es posible aplicar los algoritmos a las matrices de los problemas originales, obtenemos mejores resultados escalando las filas de la matriz de coeficientes y el correspondiente vector del lado derecho de la ecuación. Esto es equivalente a premultiplicar las ecuaciones originales $\bar{A}x = \bar{b}$ por una matriz diagonal D obteniendo $Ax = b$, que de hecho es una forma de precondicionamiento diagonal. Hay muchas posibles estrategias para realizar esto, nos decantamos por escalar $\bar{A}$ tal que $\sum_{j=1}^{n} |a_{ij}| = 1$ para todo i, la cual suele denominarse escala euclidiana.

Problema	`arc130`	`jpwh991`	`steam2`	`shl400`	`e05r0500`
n	130	991	600	663	236
Nz	1037	6027	5660	1712	5846
Simetría	0.7071	0.9979	0.999999	0.7071	0.4708
Normalidad	2.6×10^{-5}	0.9957	0.999998	0.0030	0.9052
κ	1.1×10^{10}	730	3.5×10^{6}	1.9×10^{7}	4.8×10^{6}
Tolerancia	10^{-14}	10^{-14}	10^{-14}	10^{-14}	10^{-14}
GMRES(10)	19	237	S	S	S
GMRES(20)	15	156	13860	S	S
GMRES(30)	15	123	384	S	S

Cuadro 4.2: Matrices no escaladas y resultados para GMRES.

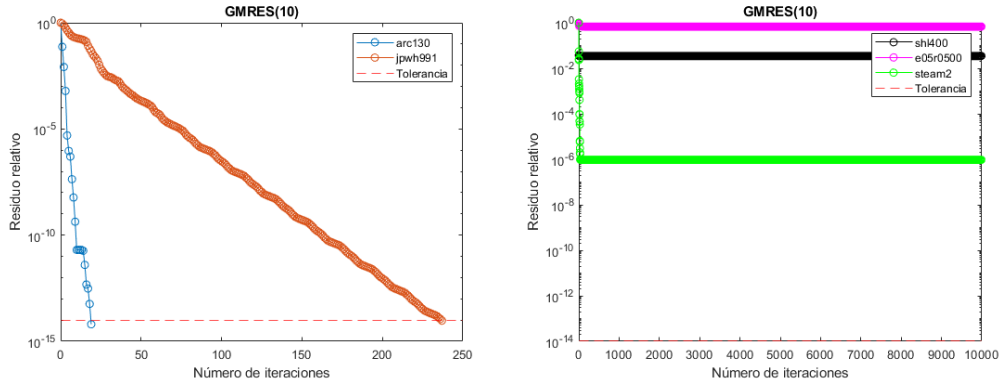


Figura 4.5: GMRES con rearranque 10 con una tolerancia de 10^{-14}

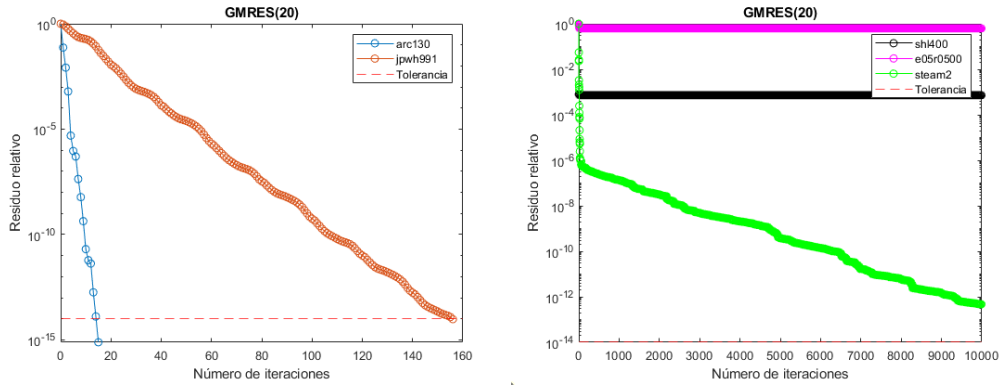


Figura 4.6: GMRES con rearranque 20 con una tolerancia de 10^{-14}

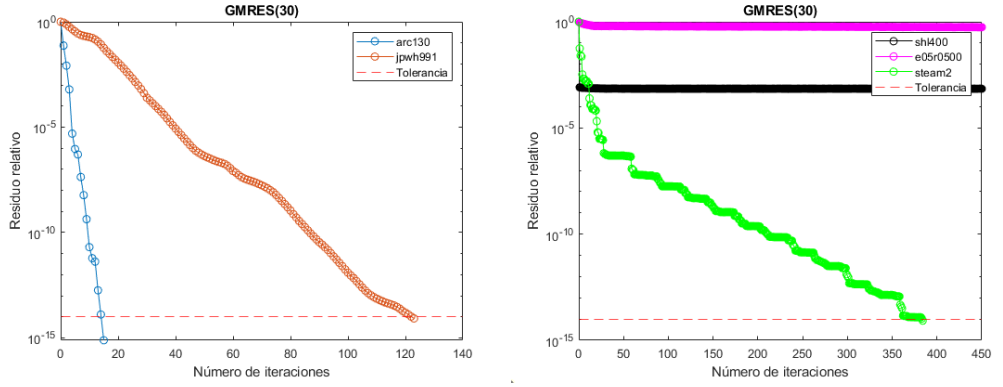


Figura 4.7: GMRES con rearranque 30 con una tolerancia de 10^{-14}.

Primero veamos los resultados de las matrices sin escalar. Observamos que

para los sistemas con matriz de coeficientes `arc130` y `jpwh991` obtenemos convergencia siempre con el algoritmo de GMRES, aunque mejoran los resultados al aumentar el número de reinicio m. En cuanto al sistema con matriz `steam2` con reinicio 10 se estanca con una tolerancia entorno a 9.48×10^{-7} alcanzando así el número máximo de iteraciones. Sin embargo al aumentar el número de rearranque vemos que para $m = 20$ ya empieza a converger pero demasiado lentamente quedándose en la iteración 10000 con una tolerancia de 4.66×10^{-13} y alcanzaría la tolerancia deseada en la iteración 13860. Finalmente con $m = 30$ conseguimos converger en la iteración 384. Para las matrices `shl400` y `e05r0500` observamos unos resultados mucho peores, en el primer caso se estanca con una tolerancia entorno a 10^{-3} y en el segundo alrededor de 0,68 y en este caso la mejora al aumentar el número de reinicio m es prácticamente imperceptible.

Problema	`arc130`	`jpwh991`	`steam2`	`shl400`	`e05r0500`
n	130	991	600	663	236
Nz	1037	6027	5660	1712	5846
Simetría	0.9838	0.9958	0.9354	0.7076	0.6079
Normalidad	0.9675	0.9917	0.8660	0.8254	0.7477
κ	6.2×10^5	88	4986	2.3×10^5	1.3×10^4
Tolerancia	10^{-14}	10^{-14}	10^{-14}	10^{-14}	10^{-14}
GMRES(10)	2040	158	14	S	S
GMRES(20)	99	110	14	S	S
GMRES(30)	56	90	14	S	S
GMRES(40)	37	85	14	S	S

Cuadro 4.3: Matrices escaladas y resultados para GMRES

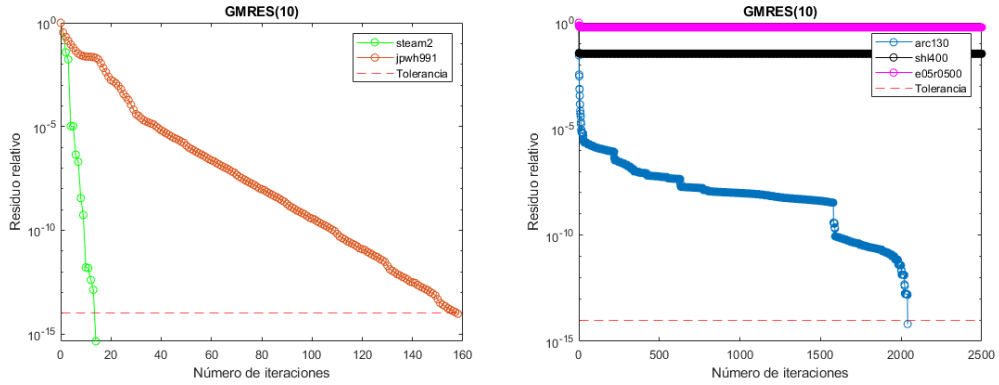


Figura 4.8: GMRES con rearranque 10 con una tolerancia de 10^{-14}

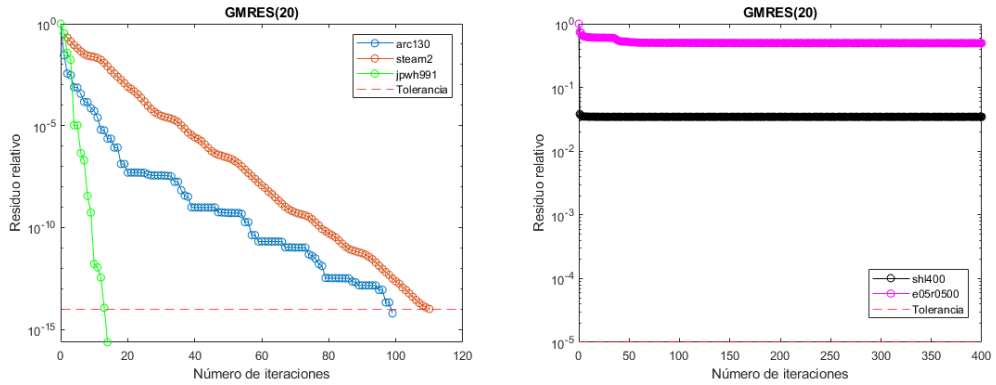


Figura 4.9: GMRES con rearranque 20 con una tolerancia de 10^{-14} para `arc130`, `jpwh991` y `steam2`. En el caso de `shl400` y `e05r0500` hemos decidido bajar la tolerancia a 10^{-5} para que se vean un poco mejor los resultados que no varían respecto a GMRES(10).

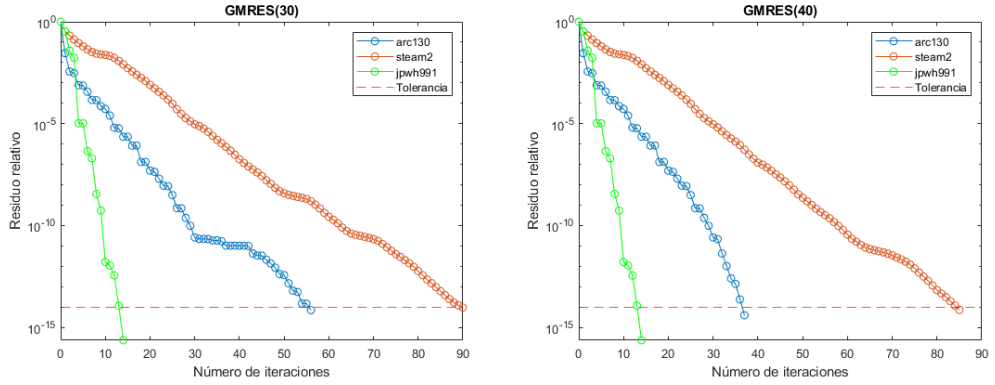


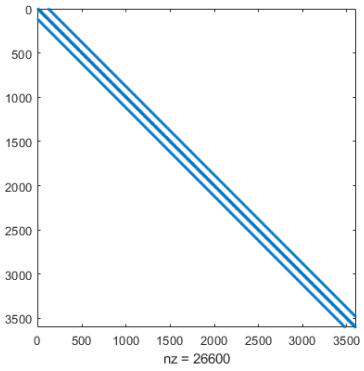
Figura 4.10: GMRES con rearranque 30 y 40 con una tolerancia de 10^{-14}, el caso `shl400` y `e05r0500` no se muestra ya que no hay ninguna variación respecto a GMRES(10) y GMRES(20).

Vemos que escalando las matrices el número de condición disminuye y la simetría y normalidad aumentan asemejándose más los valores para poder obtener los test menos correlacionados posibles. Comparando estos resultados con los anteriores de las matrices sin escalar observamos que la mayor mejora es observada en `steam2` que pasa a ser el sistema con convergencia en el menor número de iteraciones. Los sistemas `arc130` y `jpwh991` siguen convergiendo en todos los casos más o menos en un número similar de iteraciones. Sin embargo para las matrices `shl400` y `e05r0500` seguimos sin obtener convergencia incluso empeorando la tolerancia a la que se estanca el algoritmo en el caso de `shl400`.

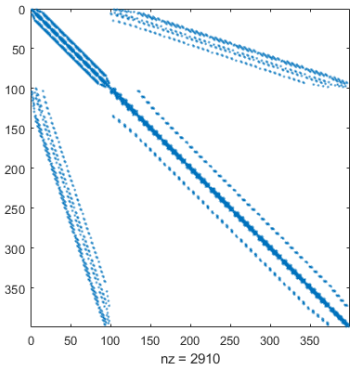
A rasgos generales vemos que el comportamiento de GMRES es bastante satisfactorio. El fallo que se produce en este algoritmo es el estancamiento prolongado, que puede darse cuando la matriz A no es positiva real y en ninguno de los 2 casos difíciles lo eran. En particular, cuando m es grande, GMRES(m) tiene que realizar una gran cantidad de operaciones aritméticas por iteración, esto afecta en el tiempo de ejecución de los algoritmos. Esta sobrecarga es algo característico de GMRES(m) pero se reduce cuando m es pequeño y n/Nz es grande, es decir, el número de elementos distintos de cero por fila en la matriz es grande.

4.5. MINRES

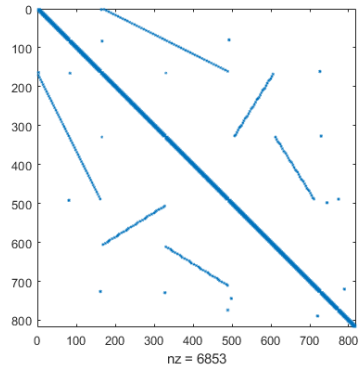
Ya que MINRES no es más que la versión simplificada de GMRES cuando la matriz A es simétrica, en este caso hemos seleccionado matrices reales simétricas indefinidas ya que en el caso definido positivo es mejor el método gradiente conjugado.



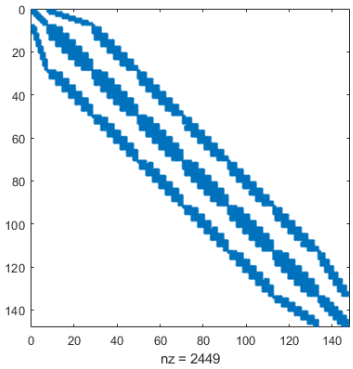
(a) bcsstk21



(b) bfw398b



(c) bcsstk19

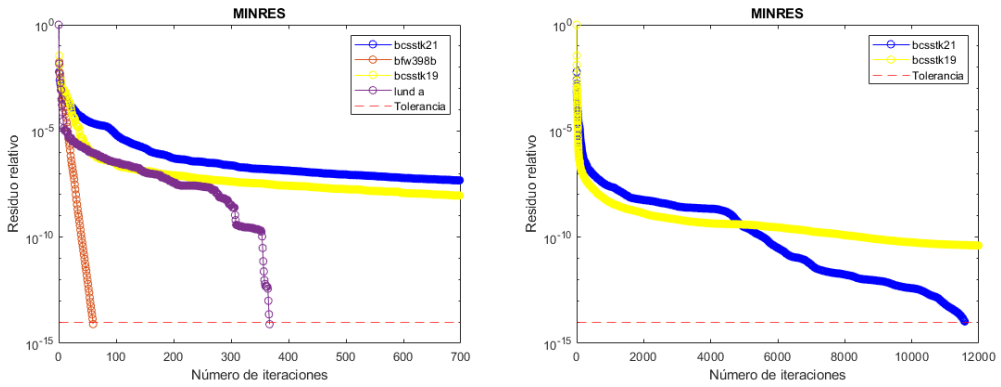


(d) lund a

Figura 4.11: Matrices test para MINRES

Problema	`bcsstk21`	`bfw398b`	`bcsstk19`	`lund a`
n	3600	398	817	147
Nz	26600	2910	6853	2449
κ	4.5×10^{7}	36	2.8×10^{11}	5.4×10^{6}
Tolerancia	10^{-14}	10^{-14}	10^{-14}	10^{-14}
MINRES	11596	60	S	367

Cuadro 4.4: Matrices sin precondicionar y resultados para MINRES



(a) Nº máximo iteraciones = 700 (b) Nº máximo iteraciones = 12000

Figura 4.12: MINRES aplicado a `bcsstk21`, `bfw398b`, `bcsstk19` y `lund a` con una tolerancia de 10^{-14}. En la segunda imagen aumentamos el número máximo de iteraciones para comprobar si `bcsstk21` y `bcsstk19` son así capaces de converger.

Observamos que los resultados para MINRES sin precondicionar varían mucho en función de la matriz de coeficientes. Vemos que tanto `lund a` como `bfw398b`, convergen en un número razonable de iteraciones. Mientras que para `bcsstk21` y `bcsstk19` vemos que se produce un estancamiento, en el caso de `bcsstk19` es intolerablemente lento. Sin embargo con `bcsstk21` si aumentamos el número máximo de iteraciones vemos que alcanzamos la tolerancia deseada en la iteración 11596.

Si los autovalores de A forman distintas agrupaciones se puede ver [8] que tenemos una convergencia bastante razonable pero aún así existen algunos contraejemplos en los que la convergencia es intolerablemente lenta.

Como ya hemos visto una posible solución para obtener mejores resultados del método MINRES es precondicionar el sistema pero en esta memoria no vamos abordar este problema.

Cómo conclusión podríamos decir que aproximar la solución del sistema $Ax = b$ donde A es simétrica indefinida es mucho más díficil que cuando A es definida, y más aún que si A es positiva real, aunque en todos los casos la convergencia depende en gran medida de las propiedades espectrales de la matriz de coeficientes, principalmente del número de condición y de la distribución de los autovalores.

Bibliografía

[1] E. W. Cheney, 1966. *Intoduction to Approximation Theory*, McGraw-Hill, New York.

[2] V. Faber and T. Manteuffel, 1984. Necessary and Sufficient Conditions for the Existence of a Conjugate Gradient Method, SIAM J. Numer. Anal. 21(2), pp. 352–362.

[3] A. Greenbaum, 1997. *Iterative methods for solving linear systems*, SIAM, Philadelphia.

[4] E. F. Kaasschieter, 1988. *A practical termination criterion for the Conjugate Gradient method*, BIT, 28: pp 308-322.

[5] Y. Saad, 1992. *Numerical Methods for Large Eigenvalue Problems*, Halstead Press, New York.

[6] Y. Saad, 2000. *Iterative Methods for Sparse Linear Systems*, Halstead Press, New York.

[7] R. S. Varga, 1962. *Matrix Iterative Analysis*, Prentice-Hall, Englewood Cliffs NJ.

[8] Broyden Ch. G., Vespucci M. T., 2004. *Krylov Solvers for Linear Algebraic Systems*, Elsevier.

[9] Henk A. Van der Vorst, 2009. *Iterative Krylov Methods for Large Linear Systems*, Cambridge University Press.