AAG0292

TRAB
ED2004
S3

UNIVERSIDAD CATOLICA ANDRES BELLO
Urb. Montalbán - La Vega - Apartado 29068
Facultad de Humanidades y Educación
Escuela de Educación

UCAB Universidad Católica ANDRÉS BELLO
CENTRO CULTURAL PADRE CARLOS G. PLAZA S.J.
BIBLIOTECA CENTRAL - SALA P. PLAZA S.J.
PLANTA BAJA - REFERENCIA

TRABAJO DE ASCENSO PARA ACCEDER A LA CATEGORÍA DE PROFESOR ASISTENTE:

MÉTODOS DE PROYECCIÓN PARA SUBESPACIOS DE KRYLOV

Realizado por:
Lic. José Javier Salas González
Enero de 2004

A Sistente

Indice

Introducción: 1-2

Capítulo I Fundamentos de Álgebra Lineal: 3

Autovalores: 4

Tipos de Matrices: 4

Producto Interno y Normas Vectoriales: 5

Producto Interno Euclídeo: 5

Normas Vectoriales: 6

Normas Matriciales: 7

Subespacios Vectoriales: 8

Vectores Ortogonales: 8

Algoritmo de Gram - Schmidt: 9

Capítulo II Métodos de Proyección: 10

Algoritmo de Proyección: 11

Métodos para Subespacios de Krylov: 12

CG: 12

Iteración de Arnoldi: 15

La Matriz de Krylov y la Factorización QR: 16

Proyección sobre Subespacios de Krylov: 17

La Iteración de Arnoldi y la búsqueda de Autovalores: 17

Lemniscatas de Arnoldi: 18

GMRES: 20

Aspectos Computacionales del GMRES: 21

Polinomios pequeños en el espectro: 23

Técnicas de Precondicionamiento: 27

Iteraciones Precondicionadas: 28

BICG: 29

CGS: 30

TMRES: 30

Algorítmo del TMRES: 31

Comparación del GMRES vs TMRES: 32 - 37

Conclusiones: 38

Bibliografía: 39

Introducción

La resolución de Sistemas de Ecuaciones Lineales es sin duda una de las áreas de mayor importancia y auge investigativo en los últimos tiempos, tanto en la física como en la matemática, podemos citar algunas razones para esta afirmación:

1. el estudio de muchas situaciones terminan en un sistema de ecuaciones de grandes magnitudes $Ax = b$, con A la matriz de coeficientes del sistema, b el vector de soluciones y x el vector solución del sistema. Al ser el sistema muy grande no puede ser abordados por los métodos convencionales, como el cálculo de la inverta A^{-1}, el método de Gauss-Jordan entre otros
2. en muchos casos el modelaje y simulación numérica hecho por los ingenieros de situaciones físicas específicas, como por ejemplo: yacimientos petroleros, fluidos en medios porosos, conducción de calor en materiales, etc.termina en la resolución de un Sistema de Ecuaciones Lineales
3. muchos métodos para la resolución de *Sistemas de Ecuaciones No Lineales* implican la resolución de un sistema de ecuaciones lineales.
4. existe una estrecha relación entre la búsqueda de autovalores y los *Métodos de Proyección* que se implementan en la solución de grandes sistemas de ecuaciones lineales.

Los *Métodos de Proyección* han sido una mina de grandes descubrimientos que han permitido solucionar con mayor eficacia y eficiencia los Sistemas de Ecuaciones Lineales, este trabajo sentará las bases teóricas necesarias para entender su funcionamiento, ilustrando los ejemplos más importantes, nos pasearemos por métodos tan conocidos como el *Algoritmo de Arnoldi*, veremos su relación con otros métodos entre ellos el *GMRES* de *Saad y Schultz* 1981 y llegaremos a analizar métodos de última generación como el *TMRES* de *Hager* 1986.

La novedad de éstos descubrimientos es sin duda una de las características relevantes de este trabajo, así como su importancia en la resolución de los Sistemas de Ecuaciones Lineales y la búsqueda de autovalores. Ésta última es un área de vital importancia en el estudio de *Ecuaciones Diferenciales* y *Sistemas de Ecuaciones Diferenciales*.

La importancia de éstos avances ha obligado a que eventos tan relevantes como la *IV Escuela Matemática de América Latina y el Caribe* (*EMALCA*) y *XVII Escuela Venezolana de Matemáticas* (*EMV*) a realizarce conjuntamente entre el 11 de septiembre y el 16 de septiembre del 2004 hallan establecido como uno de los 4 temas a desarrollar durante la semana: *Métodos Iterativos para Sistemas Lineales*.

Además incursionaremos en un grupo selecto de *Métodos de Proyección* aquellos asociados a los *Sub-espacios de Krylov* y a las *Condiciones de Ortogonalidad de Petrov-Galerkin*. Así como también, analizaremos las *Técnicas de Precondicionamiento* más elementales, técnicas aplicables a cualquier Sistema de Ecuaiones Lineales con el fin de mejorar sus características espectrales.

Antes de desarrollar los métodos de proyección es necesario hacer un recuento de *Álgebra Lineal* para ello se creó el capítulo *Fundamentos de Álgebra Lineal* necesario para familiarizarnos con el lenguaje y las herramientas utilizadas por los métodos de proyección en su desempeño, dicho capítulo abarcará tópicos desde: *Matrices Cuadradas y Autovalores* hasta llegar a *Subespacios Vectoriales* pasando por *Normas Vectoriales y Matriciales*.

Este trabajo nos permitirá además familiarizarnos con unos de los programas más importantes en el área del Cálculo Numérico: *MatLab* 6.0.0.88 (*R*12) *Profesional para Windows* programa sobre el cual se crearon y ejecutaron todos los códigos y programas. Dado que el componente de Cálculo Numérico es crucial es nuestro trabajo tenemos que especificar las características de nuestro procesador: *Pentium III* 650 *Mhz con* 120 *Mb de memoria RAM*.

Para terminar esta introducción es importante resaltar como la resolución de un Sistema de Ecuaciones Lineales permite unificar tres grandes áreas del conocimiento científico: *Cálculo Numérico, las Matemáticas y las Ciencias de la Computación* trío de áreas que se conjugan para desarrollar los métodos de proyección que estudiaremos a continuación.

Capítulo I

Fundamentos de Álgebra Lineal

Matrices Cuadradas y Autovalores

Una matriz es cuadrada de orden k si el número de filas es igual al número de columnas y ambos a su vez iguales a k, es decir $m = n = k$, entre los ejemplos más útiles tenemos la matriz identidad:

$$I = \{\delta_{ij}\}_{i,j=1,\ldots,k} \tag{1.1}$$

donde δ_{ij} es el símbolo de Kronecker

$$\delta_{ij} = \begin{cases} 1 \text{ si } i = j \\ 0 \text{ si } i \neq j \end{cases} \tag{1.2}$$

La matriz identidad satisface la identidad $IA = AI = A$ para cualquier matriz A de tamaño k. La inversa de la matriz A, cuando existe es una matriz C tal que $CA = AC = I$. La inversa de A será denotada por A^{-1}.

El *determinante* de una matriz (cuadrada) puede ser definido por muchas vías; por simplicidad nosotros utilizaremos la definición recursiva . El *determinante* de una matriz $1 \times 1, A = [a]$ es el escalar a. Para el resto de los casos el *determinante* es definido por:

$$\det(A) = \sum_{j=1}^{k} (-1)^{j+1} a_{ij} \det(A_{ij}) \tag{1.3}$$

donde A_{ij} es una matriz cuadrada de orden $n-1$ obtenida eliminando la $i-$ *ésima* fila y la $j-$ *ésima* columna de A. Una matriz es singular si $\det(A) = 0$, en otro caso es llamada nosingular. Ahora reseñemos algunas propiedades de los determinantes:

- $\det(AB) = \det(BA)$
- $\det(A^T) = \det(A)$ donde A^T es la matriz traspuesta de A; $a_{ij}^T = a_{ji}$
- $\det(\gamma A) = \gamma^k \det(A)$
- $\det(\overline{A}) = \overline{\det(A)}$
- $\det(I) = 1$

Con la definición anterior de determinante podemos mostrar por inducción que la función $p_A(\lambda) = \det(A - \lambda I)$ para λ un número complejo dado, es un polinomio de grado k, al cual llamaremos polinomio característico de A.

Autovalores

Un número complejo λ es llamado *autovalor* de A si existe un vector $u \in C^k$ con $u \neq \vec{0}$ tal que $Au = \lambda u$. El vector u es llamado *autovector* de A asociado al *autovalor* λ. El conjunto de todos los autovalores es llamado *espectro de A* y es denotado por $\sigma(A)$.

Un escalar λ es un autovalor de A si y solo si $p_A(\lambda) = \det(A - \lambda I) = 0$, con lo cual se asegura que sólo pueden haber k autovalores diferentes, por ser λ raíz del polinomio característico p_A. También es claro que una matriz que admita algún autovalor $\lambda = 0$ es singular. Un importante resultado lo enunciamos sin demostración:

Una matriz A es nosingular si su determinante es diferente de cero

Se define *radio espectral* $\rho(A)$ como el máximo módulo de los autovalores de A

$$\rho(A) = \max_{\lambda \in \sigma(A)} |\lambda| \tag{1.4}$$

Se define por *traza* de una matriz cuadrada A la suma de todos los elementos de la diagonal principal

$$tr(A) = \sum_{i=1}^{k} a_{ii} \tag{1.5}$$

es fácil observar que

$$tr(A) = \sum \lambda \tag{1.6}$$

la suma de todos sus autovalores tomando en cuenta la multiplicidad de cada autovalor.

Tipos de Matrices

La elección de un método para la resolución de un sistema de ecuaciones dependerá en muchas ocasiones de la estructura de la matriz A. Una de la propiedades más importantes de las matrices es la simetría, debido a su impacto en la disposición de los autovalores. Citemos algunos ejemplos:

- Matrices Simétricas: $A^T = A$
- Matrices Hermitianas: $A^H = A$ donde $A^H = \overline{A}^T = \overline{A^T}$
- Matrices Normales: $A^H A = AA^H$
- Matrices No-negativas: $a_{ij} > 0 \quad i = 1, ..., m \;\; j = 1, ..., n$
- Matrices Unitarias: $Q^H Q = I$

  En este último ejemplo es necesario resaltar, que una matriz Q unitaria posee como inversa su traspuesta conjugada:

  $$Q^H = Q^{-1}$$

  un matriz Q tal que $Q^H Q$ es una matriz diagonal es llamada *Ortogonal*

Algunas matrices tienen estructuras particulares muy convenientes para fines computacionales. La siguiente lista, aunque incompleta, reseña alguna de éstas matrices, las mismas son ampliamente implementadas por los *Métodos de Resolución de Sistemas de*

Ecuaciones Lineales, en especial por los referentes a *Subespacios de Krylov*.

- Matriz Diagonal: $a_{ij} = 0 \ \ \forall i \neq j$
- Matriz Triangular Superior: $a_{ij} = 0 \ \ \forall i \geq j$
- Matriz Triangular Inferior: $a_{ij} = 0 \ \ \forall i \leq j$
- Matriz de Hessenber Superior: $a_{ij} = 0 \ \ \forall i \geq j+1$
- Matriz de Hessenber Inferior: $a_{ij} = 0 \ \ \forall i \leq j+1$

Producto Interno y Normas Vectoriales

Un producto interno en un espacio vectorial real o complejo X es una función s de $X \times X \rightarrow C$, siendo C los complejos

$$x \in X, y \in X \rightarrow s(x,y) \in C$$

que satisface las siguientes condiciones:

1. $s(x,y)$ es lineal con respecto a x:
$$s(\lambda_1 x_1 + \lambda_2 x_2, y) = \lambda_1 s(x_1,y) + \lambda_2 s(x_2,y) \quad \forall\, x_1, x_2 \in X, \ \ \forall\, \lambda_1, \lambda_2 \in C \tag{1.7}$$
2. $s(x,y)$ es Hermitiana:
$$s(x,y) = \overline{s(x,y)} \quad \forall\, x,y \in X \tag{1.8}$$
3. $s(x,y)$ es Positiva Definida:
$$s(x,x) > 0 \ \ \forall x \neq 0 \in X \tag{1.9}$$

Observe que (1.8) garantiza que $s(x,x)$ es un número real, (1.9) obliga a que también sea positiva definida para todo $x \neq 0$. Para cualquier x,y tenemos:

$$s(x,0) = s(x, 0 \bullet y) = 0 \bullet s(x,y) = 0$$

similarmente $s(0,y) = 0$ para cualquier y. Por lo que (9) puede reescribirse

$$s(x,x) \geq 0 \wedge s(x,x) = 0 \Leftrightarrow x = 0 \tag{1.10}$$

Una relación muy importante que se satisface para cualquier producto interior es la llamada desigualdad de Cauchy-Schwartz:

$$|s(x,y)|^2 \leq s(x,x)s(y,y) \tag{1.11}$$

Producto Interior Euclídeo

En el caso particular del espacio vectorial $X = C^n$, dados $x,y \in C^n$ $x = (x_i)_{i=1,\ldots,n}$ $y = (y_i)_{i=1,\ldots,n}$ el producto interno euclídeo se define como:

$$\langle x,y \rangle = \sum_{i=1}^{n} x_i \overline{y_i} \tag{1.12}$$

definición que podemos reescribir utilizando matrices como:

$$\langle x,y\rangle = y^H x \tag{1.13}$$

Una propiedad fundamental del producto interno Euclídeo es:

$$\langle Ax,y\rangle = \langle x,A^H y\rangle \tag{1.14}$$

como consecuencia de esto, podemos enunciar el siguiente resultado:

Las matrices unitarias preservan el producto interno euclídeo

$$\langle Qx,Qy\rangle = \langle x,y\rangle \;\; \forall Q \text{ unitaria}$$

Utilizando la definición de producto internno podemos definir:

Matrices Positiva Definidas

Sea A una matriz cuadrada de orden k se dice *Positiva Definida* si

$$\langle Au,u\rangle > 0 \; \forall \, u \in \Re^k \tag{1.15}$$

Normas Vectoriales

Una norma vectorial en un espacio vectorial X es una función a valores reales, que satisface las siguientes condiciones:

1. *Definida Positiva*: $\|x\| \geq 0, \;\; \|x\| = 0 \Leftrightarrow x = 0$
2. *Homogeneidad*: $\|\beta x\| = |\beta|\|x\|$
3. *Desigualdad Triangular*: $\|x+y\| \leq \|x\| + \|y\|$

Para el caso especial de $X = C^n$, nosotros podemos asociar con el producto interno (1.12) la *Norma Euclídea* del vector:

$$\|x\|_2 = \sqrt{\langle x,x\rangle} \tag{1.16}$$

Si también utilizamos el hecho de que las matrices unitarias preservan el producto interno Euclídeo tenemos:

$$\|Qx\|_2 = \|x\|_2 \tag{1.17}$$

lo que implica que las transformaciones lineales asociadas con matrices unitarias Q son *isometrías* (preservan distancias).

La norma de Hölder agrupa un número considerable de normas vectoriales como casos particulares de esta:

$$\|x\|_p = \left(\sum_{i=1}^{n} |x_i|^p\right)^{\frac{1}{p}} \quad p \geq 1 \tag{1.18}$$

con sólo variar el valor de p obtenemos algunas normas de gran importancia:

$$p = 1 \Rightarrow \|x\|_1 = |x_1| + |x_2| + \cdots + |x_n| \tag{19}$$
$$p = 2 \Rightarrow \|x\|_2 = \sqrt{|x_1|^2 + |x_2|^2 + \cdots + |x_n|^2} = \sqrt{\langle x,x\rangle}$$
$$p \to \infty \Rightarrow \lim_{p\to\infty}\|x\|_p = \|x\|_\infty = \max_i\{|x_i|\}$$

La Desigualdad de Cauchy-Schwartz queda:

$$|\langle x,y\rangle| \leq \|x\|_2\|y\|_2 \tag{1.20}$$

Normas Matriciales

Para una matriz cualquiera $A \in C^{n\times m}$ nosotros definimos el siguiente conjunto especial de normas

$$\|A\|_{pq} = \max_{x\in C^m, x\neq 0} \frac{\|Ax\|_p}{\|x\|_q} \tag{1.21}$$

La norma $\|.\|_{pq}$ es inducida a través de las normas $\|.\|_p$ y $\|.\|_q$; normas que satisfacen las propiedades usuales:

- $\|A\| \geq 0,\ \forall\, A \in C^{n\times m},$ y $\|A\| = 0 \Leftrightarrow A = 0$
- $\|\beta A\| = |\beta|\|A\|,\ \forall\, A \in C^{n\times m},\ \forall\, \beta \in C$
- $\|A + B\| \leq \|A\| + \|B\|,\ \forall\, A, B \in C^{n\times m}$

Los casos más importantes están asociados a $p, q = 1, 2, \infty$. El caso $p = q$ es particularmente interesante y la norma asociada $\|.\|_{pq}$ se denota simplemente por $\|.\|_p$ y es llamada *norma p*. Una propiedad fundamental de una *norma p* es:

$$\|AB\|_p \leq \|A\|_p\|B\|_p \tag{1.22}$$

una consecuencia inmediata de la definición (1.21). Normas matriciales que satisfacen la propiedad anterior son llamadas *consistentes*. Un resultado de *consistencia* es que para cualquier matriz A :

$$\|A^k\|_p \leq \|A\|_p^k \tag{1.23}$$

En particular la matriz A^k converge a cero si y solo si la *norma p* es menor que 1.

La norma de Frobenius de una matriz A es definida por:

$$\|A\|_F = \sqrt{\sum_{j=1}^{m}\sum_{i=1}^{n}|a_{ij}|^2} \tag{1.24}$$

aunque la norma de Frobenius no cumple alguna de las propiedades características de las normas matriciales, puede probarse que la misma es consistente.

Las siguientes expresiones se derivan de la definición (1.21) :

$$\|A\|_1 = \max_{j=1,\ldots,m} \sum_{i=1}^{n} |a_{ij}| \tag{1.25}$$

$$\|A\|_\infty = \max_{i=1,\ldots,n} \sum_{i=1}^{m} |a_{ij}| \tag{1.26}$$

$$\|A\|_2 = \sqrt{\rho(A^H A)} = \sqrt{\rho(AA^H)} \tag{1.27}$$

Se puede demostrar que los autovalores de $A^H A$ son no-negativos.

Subespacios Vectoriales

Un subespacio de C^n es un subconjunto de C^n que también es un espacio vectorial complejo. El conjunto de todas las combinaciones lineales de un conjunto de vectores $G = \{a_1, a_2, \ldots, a_q\}$ de C^n es un subespacio llamado *subespacio generado por* G:

$$gn\{G\} = gn\{a_1, a_2, \ldots, a_q\} = \left\{ z \in C^n \mid z = \sum_{i=1}^{q} \beta_i a_i \right\} \tag{1.28}$$

Si los a_i son linealmente independientes, entonces cada vector de G admite sólo una expresión como combinación lineal de los $\langle a_i \rangle$. En dado caso el conjunto G es llamado *base* del subespacio $gn\{G\}$

Dos subespacios importantes asociados a toda matriz $A \in C^{n \times m}$ son:

1. Rango:

$$Ran(A) = \{Ax \mid x \in C^m\}$$

2. Kernel

$$Ker(A) = \{x \in C^m \mid Ax = 0\}$$

Vectores Ortogonales y Subespacios

Un conjunto de vectores $G = \{a_1, a_2, \ldots, a_r\}$ es llamado *ortogonal* si

$$\langle a_i, a_j \rangle = 0 \quad \forall\, i \neq j \tag{1.29}$$

ademas se llama *ortonormal* si:

$$\|a_i\|_2 = 1 \quad \forall\, i \tag{1.30}$$

Un vector que es ortogonal a todos los vectores de un subespacio S es llamado ortogonal

a dicho subespacio. El conjunto de todos los vectores ortogonales a un subespacio S es llamado complemento ortogonal de S y se denota por $S^{\perp}$. El espacio $C^n = S \oplus S^{\perp}$ lo que asegura que cualquier vector de C^n es suma direta y única de dos elementos uno perteneciente a S y otro a $S^{\perp}$.

Cualquier subespacio admite una *base ortonormal* la cual es obtenida a partir de una base y "*ortonormalizándola*". La ortonormalización puede ser ejecuta a través del bien conocido algoritmo de *Gram-Schmidt* que describiremos brevemente: dado un conjunto de vectores linealmente independientes $\{x_1, x_2, \ldots, x_r\}$ primero normalizamos el vector x_1 lo cual es fácil simplemente dividimos el vector por su norma $\|x_1\|_2$ así obtenemos q_1 de norma 1: Para ortonormalizar x_2 hacemos:

$$\begin{aligned} x_2 &= x_2 - q_1\langle x_2, q_1\rangle \\ q_2 &= \frac{x_2}{\|x_2\|_2} \end{aligned} \tag{1.31}$$

el i-ésimo paso del proceso consiste en ortogonalizar x_i contra todos los q_j previos

Algoritmo de Gram-Schmidt

1. $$r_{11} = \|x_1\|_2 \text{ if } \|x_1\|_2 = 0 \text{ PARAR else } q_1 = \frac{x_1}{r_{11}}$$

2. for $j = 2 : r$
   a. $r_{ij} = \langle x_j, q_i\rangle$
   b. for $i = 1 : j-1$
      i. $\widehat{q} = x_j - \sum_{i=1}^{j-1} r_{ij} q_i$
      end
   c. $r_{jj} = \|\widehat{q}\|_2$
   d. if $r_{jj} = 0$
      i. PARAR
      else
      i. $q_j = \frac{\widehat{q}}{r_{jj}}$
      end
   end

Capítulo II

Métodos de Proyección

La mayoría de las técnicas iterativas que existen en la práctica para resolver grandes sistemas de ecuaciones lineales utilizan, de una u otra forma, *procesos de proyección.*

Sea $Ax = b$ con A una matriz de $(n \times n)$ $b, x \in \Re^n$ supondremos que existe A^{-1} de manera de asegurar que existe una solución única $\hat{x}$. Sea L y K dos subespacios $m - dimensional$ con $m \leq n$. Una técnica de proyección en el subespacio K y ortogonal a L es un proceso que busca una solución aproximada $\tilde{x}$ para $Ax = b$ imponiendo como condición que $\tilde{x} \in K$ y $r_0 = b - A\tilde{x} \perp L$. Si $L = K$ diremos que el *método de proyección es de Galerkin* si $L \neq K$ el *método de proyección será de Petrov-Galerkin.*

Si deseamos explotar las ventajas de conocer una aproximación inicial x_0, entoncés debemos redefinir nuestro problema de la siguiente manera:

$\tilde{x} \in x_0 + K$ de modo que se cumpla la condición de *Petrov – Galerkin* $r_0 = b - A\tilde{x} \perp L$.

Note que si $\tilde{x}$ es escrito como $\tilde{x} = x_0 + \delta$ y el vector residual inicial es definido como $r_0 = b - Ax_0$ tenemos

$$b - A\tilde{x} = b - Ax_0 - A\delta = r_0 - A\delta \perp L.$$

Es decir una solución aproximada se obtiene asi:

$$\begin{aligned} \tilde{x} &= x_0 + \delta \ \text{ con } \delta \in K \\ &< r_0 - A\delta, w >= 0 \ \ \forall w \in L \end{aligned} \tag{2.1}$$

La condición de ortogonalidad impuesta por (2.1) en el nuevo residual $r_1 = r_0 - A\delta$ es ilustrada en la siguiente figura:

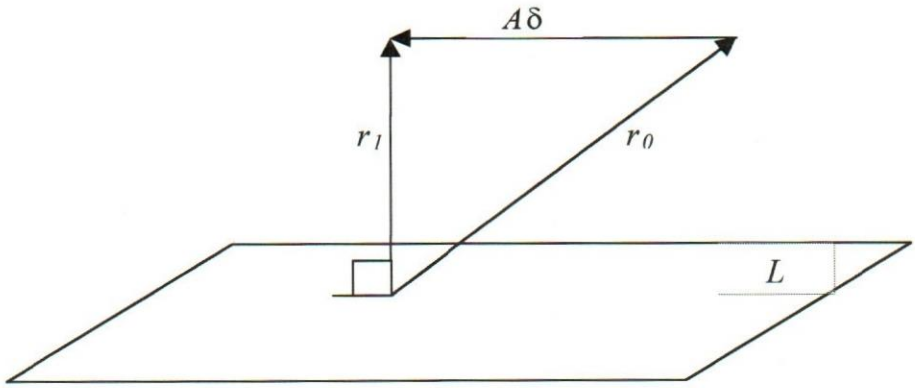

Figura 2.1: Interpretación Geométrica de la Condición de Ortogonalidad

En forma matricial será conveniente construir una base para K y L. Sea $V = [v_1|v_2|\ldots|v_{m-1}|v_m] \in \Re^{n\times m}$ una base para K y $W = [w_1|w_2|\ldots|w_{m-1}|w_m] \in \Re^{n\times m}$ una base para L, entonces queremos:

$$\tilde{x} = x_0 + Vy \text{ con } Vy = \delta \in K \tag{2.2}$$
$$A\delta = r_0 \Rightarrow AVy = r_0 \Rightarrow W^TAVy = W^Tr_0$$

Trasponiendo la base W para que pueda efectuarse el producto. Ahora

$$y = (W^TAV)^{-1}W^Tr_0 \text{ suponemos que } W^TAV \text{ no es singular} \tag{2.3}$$
$$x = x_0 + V\left((W^TAV)^{-1}W^Tr_0\right)$$

Nótese que

$$< r_0 - A\delta, w >= 0 \;\; \forall w \in L \Rightarrow W^T(r_0 - A\delta) = 0 \Rightarrow W^Tr_0 = W^TA\delta \tag{2.4}$$

Algoritmo de Proyección

Cuando hablamos de un método iterativo como el descrito estamos refiriéndonos a un proceso de refinamiento sucesivo de la solución, es decir dado x_0 hallaremos x_1 tal que la solución x_1 disminuya la norma del vector residual. Si calculamos: $r^* = A\hat{x} - b = \vec{0}$ con lo cual $\|r^*\| = 0$ en la medida en que nos aproximemos a $\hat{x}$ la norma de $\|r_k\| \to 0$. Esto nos permite olvidarnos de la solución exacta del problema y establecer una *tolerancia*, en este trabajo la tolerancia fue tomada como 10^{-7}, es decir, los métodos se detienen cuando han encontrado una solución $\tilde{x}_k$ que difiere de la solución verdadera en $10^{-7} = 0.0000001$

1. Dado x
2. while $norm(Ax - b) > tolerancia$
   Formar las bases V y W para K y L respectivamente
   $r = Ax - b$
   $y = (W^TAV)^{-1}W^Tr_0$
   $x = x + Vy$
   end

¿Cuándo W^TAV es invertible?

Hay muchas situaciones para las cuales esta matriz es invertible, cada método de proyección desarrollado garantiza de forma diferente está condición, sólo para ilustrar tomaremos el caso:

- Cuando A es positiva definida y $L = K$

Demostración

Si A es positiva definida, entonces: $x^TAx > 0 \; \forall x \neq \vec{0}$
Como $L = K \Rightarrow W = VG$ con G invertible $\therefore \; B = W^TAV = G^TV^TAV$

Métodos para subespacios de Krylov

Vimos que para resolver $Ax = b$ necesitamos buscar una aproximación $x_m \in x_0 + K_m$ de dimensión m imponiendo la condición de Petrov-Galerkin

$$b - Ax_m \perp L_m \tag{2.5}$$

con L_m un subespacio de dimensión m. Sabemos que x_0 representa una aproximación inicial arbitraria. Un *método basado en subespacios de Krylov* genera una base de la siguiente forma:

$$K_m(A, r_0) = gen\{r_0, Ar_0, A^2r_0, A^3r_0, \ldots, A^{m-1}r_0\} \tag{2.6}$$

con $r_0 = b - Ax_0$. La diversidad de métodos proviene de la forma en que se defina L_m. Entre ellos tenemos: *CG* método de *Gradiente Conjugado*, el *GMRES* Método Generalizado de Minimización de Residuales (Generalized Minimal Residual), el *BICGSTAB* método Gradiente Bi-Conjugado Estabilizado.

Desde el punto de vista de Teoría de la Aproximación buscamos una solución de la forma:

$$\hat{x} = A^{-1}b \approx x_0 + q_{m-1}(A)r_0 \tag{2.7}$$

donde q es un polinomio de orden $m - 1$ en A.

CG

El Método de *Gradiente Conjugado* es la iteración originaria para un subespacio de Krylov, el más famoso de los métodos conocidos y uno de los soportes del cálculo numérico. Descubierto por *Hestenes y Stiefel* en 1952, para resolver *sistemas simétricos y positivos definidos*. Se conoce como el *método de descenso más rápido* considera:

$$K_m = gen\{b, Ab, A^2b, \ldots, A^{m-1}b\} \tag{2.8}$$

de forma que minimiza en cada paso m

$$f(x_m) = (x_m - \hat{x})^T A(x_m - \hat{x}) = \|x_m - \hat{x}\|_A^2 \tag{2.9}$$

donde $\hat{x}$ es la solución exacta de $Ax = b$.

El vector cuya norma-A consideraremos es $e_m = \hat{x} - x_m$ como el error al paso m. La iteración del Gradiente Conjugado puede ser descrita como sigue:

"un sistema de fórmulas recurrentes que generan una sucesión única de iterados $\{x_m \in K_m\}$ con la propiedad tal que en el paso m $\|e_m\|_A$ es mínima"

Trefethen L. y Bau D

Nosotros presentaremos el algoritmo sin prestar atención a la motivación inicial, luego enunciaremos algunas propiedades de ortogonalidad y a partir de éstas demostraremos la minimalidad de $\|e_m\|_A$ en el Teorema 2.1

Algoritmo CG Gradiente Conjugado de *Hestenes y Stiefel*

1. $x_0 = 0,\ r_0 = b,\ p_0 = r_0$
2. for $n = 1,2,3...$

$$\beta_n = \frac{< r_{n-1}, r_{n-1} >}{< Ap_{n-1}, p_{n-1} >}$$

$$x_n = x_{n-1} + \beta_n p_{n-1}$$

$$r_n = r_{n-1} - \beta_n A p_{n-1}$$

$$\omega_n = \frac{< r_n, r_n >}{< r_{n-1}, r_{n-1} >}$$

$$p_n = r_n + \beta_n p_{n-1}$$

end

Del algoritmo anterior se desprenden las siguientes propiedades que enunciamos sin demostración, la misma puede ser realizada por inducción matemática sobre el paso m de la iteración.

Las siguientes identidades de subespacios:

$$K_m = gen\{b, Ab, A^2b, \ldots, A^{m-1}b\} = gen\{r_0, r_1, r_2, \ldots, r_{m-1}\} \qquad (2.10)$$
$$= gen\{p_0, p_1, p_2, \ldots, p_{m-1}\} = gen\{x_1, x_2, x_3, \ldots, x_m\}$$

La ortogonalidad de los residuales:

$$r_m^T r_j = 0 \quad j < m \qquad (2.11)$$

Las direcciones de búsqueda son A-conjugadas:

$$p_m^T A p_j = 0 \quad j < m \qquad (2.12)$$

A partir de (2.11) y (2.12) podemos demostrar rápidamente el siguiente teorema:

Teorema 2.1: Sea la iteración de Gradiente Conjugado aplicada a una matriz simétrica y positiva definida A tal que la solución del sistema $Ax = b$ es $\hat{x}$. Si la iteración aún no converge en el paso m (es decir, $r_{m-1} \neq \vec{0}$), entonces x_m es el único punto en K_m que minimiza $\|e_m\|_A$. Además la convergencia es monótona

$$\|e_m\|_A \leq \|e_{m-1}\|_A \qquad (2.13)$$

y $\|e_m\|_A = 0$ para algún $m \leq n$

Demostración:

Sea $x_m \in K_m$ mostraremos que es el único punto de K_m que minimiza $\|e_m\|_A$. Sea ahora $x = x_m - \Delta x \in K_m$ con $e = \hat{x} - x = e_m + \Delta x$. Si calculamos:

$$\|e_m\|_A^2 = (e_m + \Delta x)^T A(e_m + \Delta x) = (e_m)^T A(e_m) + (\Delta x)^T A(\Delta x) + 2(e_m)^T A(\Delta x)$$

donde el último término es igual a $2(r_m)^T(\Delta x)$, osea el producto interno de r_m con otro vector de K_m lo que por (2.11) dicho producto es igual a cero. Esta es la propiedad de ortogonalidad que hace al CG poderoso. Además tenemos que:

$$\|e_m\|_A^2 = (e_m)^T A(e_m) + (\Delta x)^T A(\Delta x)$$

donde $\|e_m\|_A^2 = (e_m)^T A(e_m) \Rightarrow (\Delta x)^T A(\Delta x) = 0$. Dado que A es positivo definida $(\Delta x)^T A(\Delta x) \geq 0 \Leftrightarrow \Delta x = \vec{0} \Rightarrow x = x_m$. Ahora $\|e_m\|_A$ es mínima $\Leftrightarrow x = x_m$.

El resto de las implicaciones siguen rápidamente, la propiedad de monotonía (2.13) es una consecuencia de la inclusión de $K_m \subseteq K_{m+1}$ con K_m un subespacio de $\Re^n$. Luego la convergencia es alcanzada en no más de n pasos.

Ejemplo: Sea $A_{500\times500}$ como sigue:

$$A = \begin{cases} a_{ii} = 1 \\ a_{ij} = a_{ji} \in [-1, 1] \text{ aleatorio} \end{cases}$$

a esta matriz vamos a sustituir sus entradas $|a_{ij}| > p$ por cero.

La matriz resultante es simétrica, mal condicionada y para p cercanos a cero A es positivo definida. La siguiente gráfica muestra el comportamiento del CG para diferentes valores del parámetro p

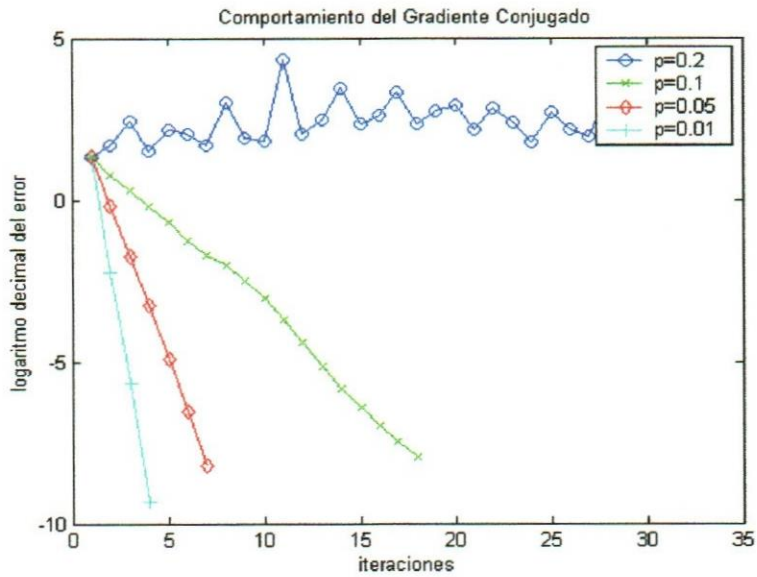

Figura 2.2: Gradiente Conjugado

A medida que p se acerca a cero la matriz A se hace positivo definida y la iteración de Gradiente Conjugado es cada vez más eficiente. Para $p = 0.2$ la iteración no llega a converger.

La Iteración de Arnoldi

Es un método de proyección sobre K_m para matrices no hermitianas. Este proceso fue introducido en 1951 con la idea de reducir una matriz densa A a su forma de Hessenberg. Arnoldi presento este método de esta manera, pero, además de conseguir llevar la matriz A a su forma de Hessenberg encontro que los *autovalores* de esta matriz (de Hessenberg) son una buena aproximación a los *autovalores de la matriz original*, por esta razón se considera como un método eficiente para el cálculo de autovalores de *matrices grandes y esparcidas*.

El algoritmo básico

El proceso de Arnoldi es un algoritmo útil para construir un base ortogonal del subespacio de Krylov K_m.

1. Elegir un vector v_1 con $\|v_1\|_2 = 1$
   for $j = 1 : m$
1. for $i = 1 : j$
2. $h_{ij} =< Av_j, v_i >$
3. $w_j = Av_j - \sum_{i=1}^{j} h_{ij} v_i$
4. $h_{j+1,j} = \| w_j \|_2$
5. if $h_{j+1,j} == 0$
6. break
7. end
8. end
9. $v_{j+1} = \frac{w_j}{h_{j+1,j}}$
10. end

En cada paso el algoritmo multiplica el previo vector de Arnoldi por la matriz A y ortonormaliza el vector resultante w_j al igual que a todos los vectores previos v_i por el proceso estandar de *Gram-Schmidt*.

Arnoldi y la construcción de bases para los subespacios de Krylov

Sea $V_m = [v_1|v_2|\ldots|v_m]$ y $\widetilde{H_m}$ la matriz de Hessenberg de tamaño $(m+1) \times m$ con elementos $h_{ij} \neq 0$ definidos por el algoritmo de Arnoldi. También definimos H_m la matriz matriz cuadrada más grande dentro de $\widetilde{H_m}$ de tamaño $m \times m$ (es la misma matriz $\widetilde{H_m}$ sólo que se eliminó la última fila). Entonces se mantienen las siguientes relaciones:

$$AV_m = V_m H_m + w_m e_m^T \tag{2.10}$$
$$AV_m = V_{m+1} \widetilde{H_m}$$

Esto quiere decir que:

$$\begin{bmatrix} & & \\ & A & \\ & & \end{bmatrix}\begin{bmatrix} v_1 & \cdots & v_m \end{bmatrix} = \begin{bmatrix} v_1 & \cdots & v_{m+1} \end{bmatrix}\begin{bmatrix} h_{11} & \cdots & h_{1m} \\ h_{21} & & \vdots \\ & \ddots & \vdots \\ & & h_{m+1,m} \end{bmatrix}$$

Donde la $m-ésima$ columna de esta ecuación puede escribirse como:

$$Av_m = h_{1m}v_1 + \cdots + h_{m,m}v_m + h_{m+1,m}v_{m+1} \tag{2.11}$$

en pocas palabras v_{m+1} satisface una relación de recurrencia de $(m+1)$ términos envuelto con los demás vectores previos de Krylov. Lo cual evidencia que los vectores $\{v_1, v_2, \ldots, v_m\}$ forman una base de los sucesivos subespacios de Krylov generados por A y b definidos como:

$$K_m(A,b) = gen\{b, Ab, A^2b, A^3b, \ldots, A^{m-1}b\} = gen\{v_1, v_2, \ldots, v_m\} \tag{2.12}$$

más aún si los v_i son ortonormales la base es ortonormal.

La Matriz de Krylov y la factorización QR

Si expresamos la observación anterior en forma matricial, definimos la matriz de Krylov de tamaño $n \times m$ como:

$$K_m = \begin{bmatrix} b & Ab & \cdots & A^{m-1}b \end{bmatrix}$$

matriz que puede ser reducida por la *Factorización QR a*: $K_m = Q_m R_m$ vista de forma matricial como:

$$K_m = \begin{bmatrix} & & \\ q_1 & \cdots & q_m \\ & & \end{bmatrix}\begin{bmatrix} r_{11} & \cdots & r_{1m} \\ & \ddots & \vdots \\ & & r_{m,m} \end{bmatrix}$$

donde Q_m es V_m. Como en el proceso de Arnoldi no se calculan K_m ni R_m explicítamente, el algoritmo se hace inestable para matrices muy mal condicionadas en general. Aunque las columnas de K_m tienden todas a aproximar el mismo autovector dominante de A. Las dos ecuaciones anteriores dan una explicación intuitiva de porque el proceso de Arnoldi lidera los métodos para determinar autovalores.Claramente K_m contiene muy buena información sobre los autovalores de A con norma suficientemente grandes, y la factorización QR podría revelar esta información sacando fuera una aproximación al autovector luego otra, arrancando con el autovector dominante.

Proyección sobre subespacios de Krylov

Otra forma de ver el algoritmo de Arnoldi es como un cálculo de proyecciones sobre sucesivos espacios de Krylov. Esto se ve, notando que el producto $V_m^* V_{m+1}$ es la Matriz Identidad de tamaño $n \times (n+1)$, es decir, una matriz que tiene 1 en la diagonal principal y ceros en el resto. Por ello, $V_m^* V_{m+1} \widetilde{H_m} = H_m$ la matriz de Hessenberg.

Esta matriz puede interpretarse como la representación en la base $\{v_1, v_2, \ldots, v_m\}$ de la proyección ortogonal de A en K_m. Profundicemos al respecto, considere el operador lineal $K_m \longrightarrow K_m$ definido como: dado $z \in K_m$, aplicar A sobre él, entonces Az es proyectado ortogonalmente sobre K_m. Como el proyector ortogonal de C^m sobre K_m es $V_m V_m^*$, este operador puede escribirse $V_m V_m^* A$ con respecto a la base estandar de C^m. Con respecto a la base de columnas de V_m podemos escribir: $V_m^* A V_m$

El tipo de proyección descrito viene a través de aplicacion es y del análisis numérico. En otro contexto es conocido como el proceso de Rayleigh-Ritz, no coincidencialmente, ya que los elementos de la diagonal de H_m se conocen como las cantidades de Rayleig de A con respecto a los vectores v_j. Este proceso de proyección es también una de las ideas subyacente a la solución de ecuaciones diferenciales parciales por el método de elementos finitos, así como sus familiares más cercanos son conocidos como métodos espectrales.

Ya que H_m es una proyección de A, uno podría imaginarse que sus autovalores estan relacionados con los de A en una forma muy conveniente. Éstos n números

$$\{\theta_j\} = \left\{\text{autovalores de } H_m\right\} \tag{2.13}$$

son llamados *estimaciones de Arnoldi* para autovalores (para el paso m) o *valores de Ritz* de A (con respecto de K_m). Más adelante veremos como éstos números pueden ser extraordinariamente aproximados a alguno de los autovalores de A, incluso para $m \ll n$.

La Iteración de Arnoldi para la búsqueda de autovalores

La búsqueda de autovalores mediante la iteración de Arnoldi se hace buscando los autovalores de la matriz de Hessenberg H_m creada en cada paso por el algoritmo de Arnoldi, a éstos autovalores se les conoce como las estimaciones de Arnoldi o valores de Ritz, la búsqueda de éstos autovalores en las matrices de Hessenberg se hace por los métodos estandar entre ellos la factorización QR.

Si $m \ll n$ por razones de cálculo numérico, uno no puede esperar calcular todos el autovalores de A por este proceso, ¿Cuáles autovalores podemos encontrar a través de Arnoldi?. Tipicamente, se pueden hallar los autovalores extremos, aquellos que están más cerca del radio espectral de la matriz A. Afortunadamente éstos son de interés principal en la mayoría de las aplicaciones.

Arnoldi y la Aproximacion Polinomial

Sea x un vector en el subespacio de Krylov K_m. Todo x puede escribirse como una combinación lineal de potencias de A para algún b.

$$x = c_0 b + c_1 Ab + c_2 A^2 b + \cdots + c_{m-1} A^{m-1} b \tag{2.14}$$

Esto puede expresarse de forma compacta, si definimos $q(z) = c_0 + c_1 z + c_2 z^2 + \cdots + c_{m-1} z^{m-1}$ obteniendo

$$x = q(A)b \tag{2.15}$$

Un análisis de las iteraciones de Arnoldi toma la siguiente forma. Definimos:

$$P^m = \{\text{Polinomio de grado } m \text{ tal que } c_m = 1\}$$

Ahora consideremos el siguiente problema de aproximación:

Arnoldi - Lanczos Problema de Aproximación: Buscar $p^m \in P^m$ tal que

$$\|p^m(A)b\| = \textbf{mínima} \tag{2.16}$$

¿Qué relación tienen las aproximaciones polinomiales con algún autovalor de A?

Si nuestro objetivo es encontrar un polinomio p^m con la propiedad de que $p^m(A)$ es pequeño, un medio eficaz para este fin sería escoger p^m para tener ceros cerca de los autovalores de A.

Consideremos un caso extremo, supongamos que A es diagonalizable y que sólo posee $m \ll n$ autovalores distintos, con p^m un polinomio minimal de grado m. Entonces, después de m pasos, todos los demás autovalores serán calculados de forma exacta, por lo menos el vector b contiene componentes en direcciones asociadas a todos los autovalores. Después de éstos m pasos el polinomio minimal de A es calculado exactamente.

Ahora el convenio entre los valores de Ritz y los autovalores es aproximado en lugar de exacto, y en lugar de tener un polinomio minimal, el resultado es un *pseudo-polinomio minimal*, entiéndase, un polinomio p^m con la propiedad de que $\|p^m(A)\|$ es pequeño

Lemniscatas de Arnoldi

El proceso de convergencia puede ilustrarse visualmente, por el gráfico de la lemníscata en el plano complejo. Una *lemníscata* es una curva o colección de curvas

$$\{z \in C : |p(z)| = k\} \tag{2.17}$$

donde p es un polinomio y k una constante. Si p es el polinomio de Arnoldi p^m para una iteración con la matriz A en el paso m, entonces k toma el valor

$$k = \frac{\|p^m(A)b\|}{\|b\|} \tag{2.18}$$

donde la curva definida es llamada *Lemniscatas de Arnoldi*. Como el número de iteraciones m aumenta, componentes de éstas lemníscatas aparecen típicamente rodeando los autovalores extremos de A y se encogen rapidamente alrededor de un punto.

Para ilustrar esta idea, sea $A = \text{rand}(100)$ una matriz de tamaño 100×100 con todas sus entradas números que provienen de la distribución normal con media 0 y desviación $\sqrt{100}$. A continuación se presentan los autovalores de A y las aproximaciones de Arnoldi para los autovalores para $m = 5, 7$ y 10

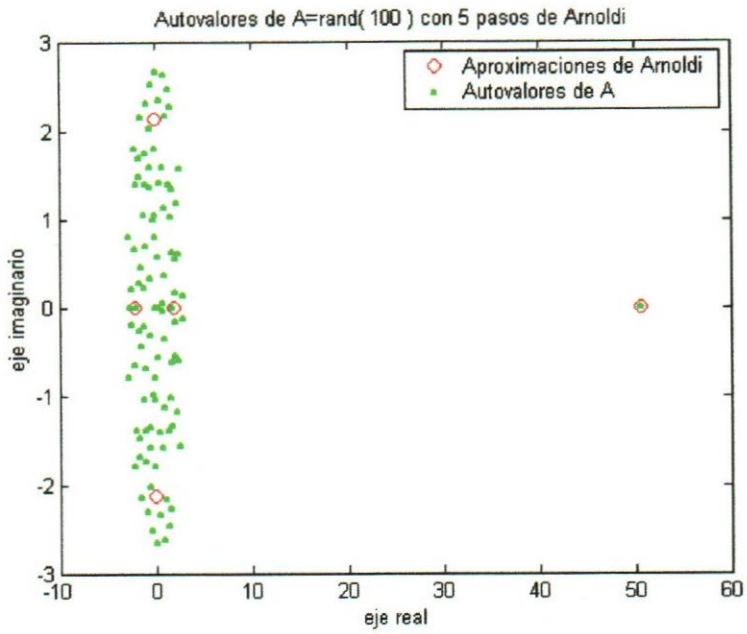

Figura 2.3: Método de Arnoldi y Autovalores

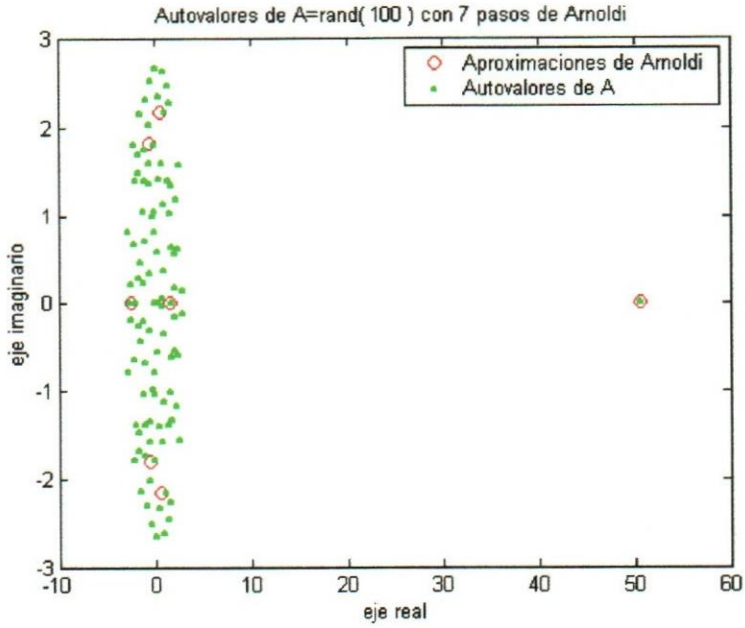

Figura 2.4: Método de Arnoldi y Autovalores

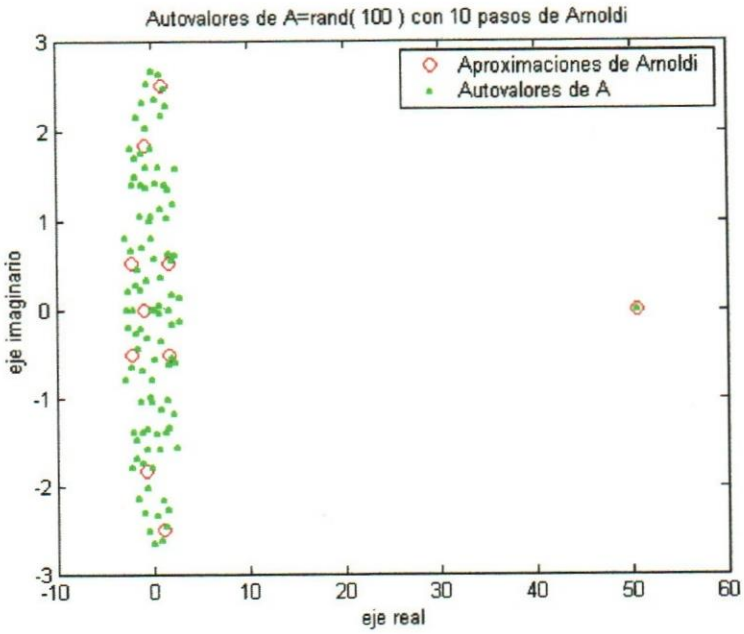

Figura 2.5: Método de Arnoldi y Autovalores

Es importante destacar como las aproximaciones de Arnoldi o Ritz en muy pocas iteraciones se posicionan alrededor de los autovalores dominantes.

GMRES

El Método Generalizado de Minimización de Residuales (*Generalizad Minimum Residual Method GMRES*) es un método de proyección basado en tomar: $K = K_m$ y $L = AK_m$, con K_m el subespacio de Krylov con $v_1 = r_0/\|r_0\|$. El GMRES está pensado para resolver matrices no simétricas. Es una extensión de otro método de proyección el MINRES, para sistemas no simétricos. Este método genera una secuencia de vectores ortogonales que son almacenados para ser usados en iteraciones sucesivas. La forma más popular del *GMRES* se basa en la ortogonalización de Gram-Schmidt.

Cualquier vector x en $x_0 + K_m$ se puede escribir como:

$$x = x_0 + V_m y \tag{2.19}$$

donde y es el m-vector definido por:

$$J(y) = \|b - Ax\| = \|b - A(x_0 + V_m y)\| \tag{2.20}$$

de la relación (2.20) resulta

$$\begin{aligned} b - Ax &= b - A(x_0 + V_m y) \\ &= r_0 - AV_m y \\ &= \beta v_1 - V_{m+1}\widetilde{H_m} y \\ &= V_{m+1}\left(\beta e_1 - \widetilde{H_m} y\right) \end{aligned}$$

Visto que los vectores columnas de V_{m+1} son ortonormales, tenemos

$$J(y) = \|b - Ax\| = \|b - A(x_0 + V_m y)\| = \left\| \beta e_1 - \widetilde{H_m} y \right\| \tag{2.21}$$

La aproximación GMRES es el único vector de $x_0 + K_m$ que minimiza $J(y)$. Por (2.19) y (2.20) éstas aproximaciones pueden ser obtenidas fácilmente con:

$$\begin{aligned} x_m &= x_0 + V_m y_m \\ y_m &= \arg\min_y \left\| \beta e_1 - \widetilde{H_m} y \right\| \end{aligned} \tag{2.22}$$

El calculo de y_m no requiere de elevados computos, sólo de la solución de un sistema casi-cuadrado donde m es típicamente pequeño.

El Algoritmo GMRES

1. $r_0 = b - Ax_0,\ \beta = \|r_0\|$ y $v_1 = r_0/\beta$
2. Arnoldi para hallar V_m y $\widetilde{H_m}$
3. $y_m = \arg\min_y \left\| \beta e_1 - \widetilde{H_m} y \right\|$
4. $x_m = x_0 + V_m y_m$

Aspectos Computacionales del GMRES

Para resolver $\arg\min_y \left\| \beta e_1 - \widetilde{H_m} y \right\|$ conviene transformar la matriz de Hessenberg en una matriz triangular superior. Usando *Transformaciones de Givens*

$$G_i = \begin{array}{c} \begin{bmatrix} 1 & & & & & 0 \\ & \ddots & & & & \\ & & c_i & s_i & & \\ & & -s_i & c_i & & \\ & & & & \ddots & \\ 0 & & & & & 1 \end{bmatrix} \end{array} \begin{array}{l} \\ \\ \leftarrow \text{fila } i \\ \leftarrow \text{fila } i+1 \\ \\ \\ \end{array}$$

Si hacemos esto m veces obtenemos

$$Q_m = G_m \bullet G_{m-1} \bullet \cdots \bullet G_2 \bullet G_1$$
$$\widetilde{R_m} = Q_m \widetilde{H_m}$$
$$\widetilde{g_m} = Q_m(\beta e_1) = [r_1, r_2, \ldots, r_{m+1}]^T$$

Como Q_m es unitaria, $\arg\min_y \left\| \beta e_1 - \widetilde{H_m} y \right\| = \arg\min_y \left\| \widetilde{g_m} - \widetilde{R_m} y \right\|$

Nótese que:

$$\widetilde{R_m} = \begin{bmatrix} \widehat{R_m} \\ 0 \end{bmatrix}$$

$$\therefore \left\| \begin{bmatrix} \widehat{g_m} \\ r_m \end{bmatrix} - \begin{bmatrix} \widehat{R_m} \\ 0 \end{bmatrix} y \right\|^2 = \|r_m\|^2 + \left\| \widehat{g_m} - \widehat{R_m} y \right\|^2 = \|r_m\|^2$$

para y la solución.

De lo anterior se desprende que:

1. El vector que minimiza $\left\| \beta e_1 - \widetilde{H_m} y \right\|$ está dado por $y_m = \widehat{R_m}^{-1} \widehat{g_m}$
2. El vector residual en el paso m satisface:
$$b - Ax_m = V_{m+1}\left(\beta e_1 - \widetilde{H_m} y_m\right) = V_{m+1} Q_m^T (r_{m+1} e_{m+1}) \tag{2.23}$$
y por lo tanto: $\|b - Ax_m\| = |r_{m+1}|$

¿Cuán rapido converge el GMRES?

Comenzaremos con 2 observaciones:

1. La convergencia del GMRES es monótona
$$\|r_{m+1}\| \leq \|r_m\| \tag{2.24}$$
2. La razón es que $\|r_m\|$ es el más pequeño residual posible para el subespacio K_m, al aumentar K_m a K_{m+1} nosotros sólo podemos decrecer la norma del residual, o en el peor de los casos mantenerla igual. Note que $P^m \subseteq P^{m+1}$ y considerando que $P^m \not\subseteq P^{m+1}$.
3. Después de n pasos el proceso debe converger, con la ausencia de errores de redondeo: $\|r_m\| = 0$

Para obtener más infomación sobre la convergencia, nosotros debemos volver al problema de aproximación polinomial. De allí sabemos que:

$$\|r_m\| = \|p_m(A)b\| \leq \|p_m(A)\| \|b\| \tag{2.25}$$

y por ende la convergencia del GMRES esta dado por la desigualdad

$$\frac{\|r_m\|}{\|b\|} \leq \inf_{p^m \in P^m} \|p_m(A)\| \tag{2.26}$$

Esto nos trae a una pregunta matemáticamente elegante: Dada una matriz A y una número m de iteraciones, ¿Cuán pequeño puede ser $\|p_m(A)\|$? Esta pregunta es la base de casi todos los análisis de convergencia de iteraciones sobre subespacios de Krylov para resolver sistemas de ecuaciones.

Polinomios pequeños en el espectro

Dada una matriz A y una número m de iteraciones, ¿Cuán pequeño puede ser $\|p_m(A)\|$?

El camino estandar para obtener estimaciones es buscar a los polinomios $p(z)$ que son lo más pequeños posibles en el espectro $\Lambda(A)$, mientras $p(0) = 1$. Si p es un polinomio y S es un conjunto en el plano complejo, definimos el escalar $\|p\|_S$ como:

$$\|p\|_S = \sup_{z \in S} |p(z)| \tag{2.27}$$

Suponga que A es diagonalizable, satisfaciendo $A = V\Lambda V^{-1}$ para alguna matriz invertible V y una matriz Λ diagonal donde sus entradas son los autovalores. Entonces tenemos:

$$\|p_m(A)\| \leq \|V\| \|p(\Lambda)\| \|V^{-1}\| = \kappa(V) \|p\|_{\Lambda(A)} \tag{2.28}$$

Combinando este resultado con (27) podemos formular el siguiente teorema:

Teorema: Al m – *ésimo* paso del GMRES, el residual r_m satisface

$$\frac{\|r_m\|}{\|b\|} \leq \inf_{p^m \in P^m} \|p_m(A)\| \leq \kappa(V) \inf_{p^m \in P^m} \|p\|_{\Lambda(A)} \tag{2.29}$$

donde $\Lambda(A)$ es el conjunto de autovalores de A, V es una matriz invertible de autovectores (al asumir que A es diagonalizable) y $\|p\|_{\Lambda(A)}$ es como se definió en (2.28)

Este teorema puede ser resumido en palabras como sigue. Si A no esta lejos de lo normal en el sentido de que $\kappa(V)$ es pequeño y si la norma del polinomio de grado m decrece en el espectro $\Lambda(A)$ decrece rápidamente con m. Entonces el *GMRES converge rápidamente*.

Ejemplo: Sea $A = 2I_m + 0,5\frac{\text{rand}(m)}{\sqrt{m}}$ para $m = 200$.

Mostremos primero la distribución de los autovalores de A:

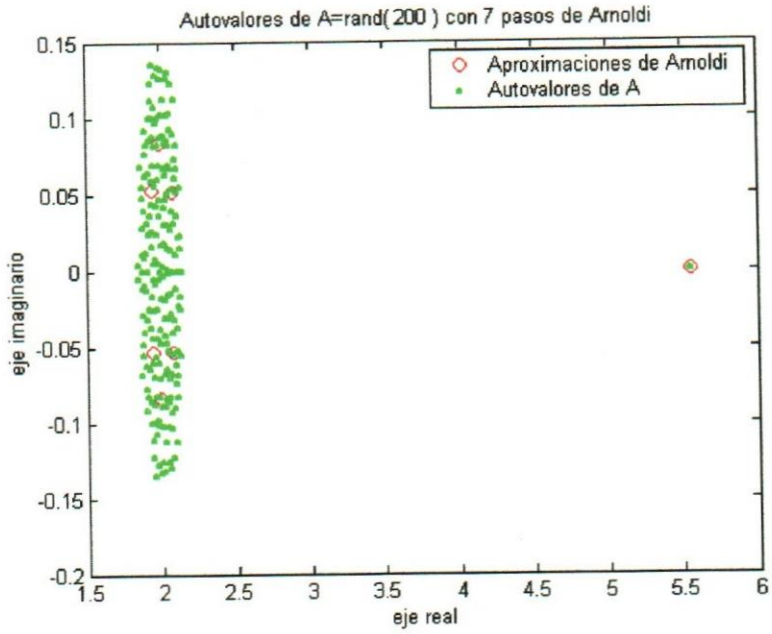

Figura 2.6: Método de Arnoldi y Autovalores

Ahora si definimos el sistema lineal $Ax = \vec{1}$ y aplicamos *GMRES* obtenemos que la norma del error hasta hallar la convergencia es:

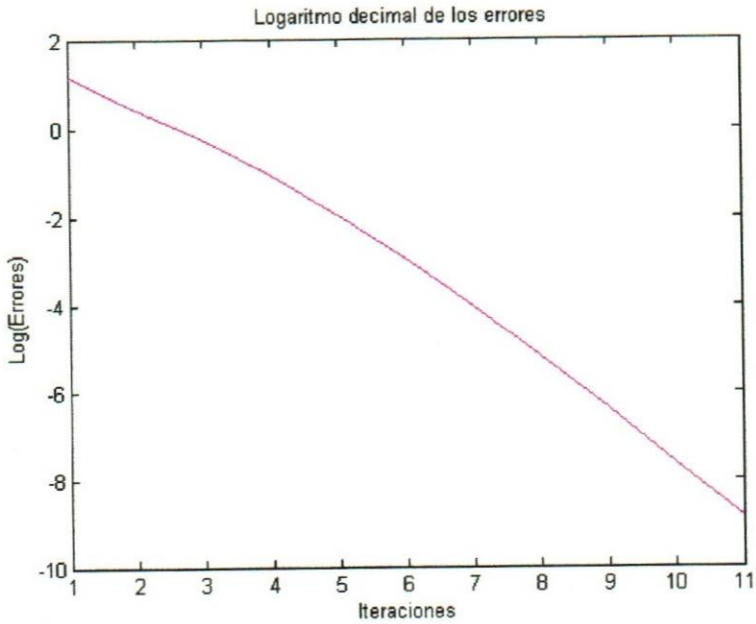

Figura 2.7: Convergencia del GMRES

donde en sólo 11 iteraciones alcanzó un error de 10^{-9}. Esto es debido a la proximidad entre los autovalores de A. La convergencia en este caso es extraordinariamente rápida con una velocidad aproximada de 4^{-n}. La razón para esto no es díficil de divisar, como el espectro

de A esta casí llenando el disco de centro $z = 2$ con radio $r = \frac{1}{2}$, $\|p(A)\|$ es aproximadamente minimizada por el cambio $p(z) = \left(1 - \frac{z}{2}\right)^n$. Ahora $I - A/2$ es una matriz random escalada tal que su espectro llena el disco de radio $\frac{1}{4}$ y centro 0, nosotros sabemos que

$$\|p(A)\| = \left\| \left(I - \frac{A}{2}\right)^n \right\| \approx 4^{-n}$$

Esta matriz A está bien condicionada con un número de condición $\kappa(A) \approx 2.0\dot{3}$. La desviación de la normalidad es modesta con $\kappa(V) \approx 141$

Otro ejemplo: $A^* = A + D$ con A como antes y D una matriz diagonal con entradas complejas $d_k = (-2 + 2\sin\theta_k) + i\cos\theta_k \quad \theta_k = \frac{k\pi}{m-1}, \quad 0 \leq k \leq m-1$ m la dimensión de A.

Los autovalores de A^* están distribuidos:

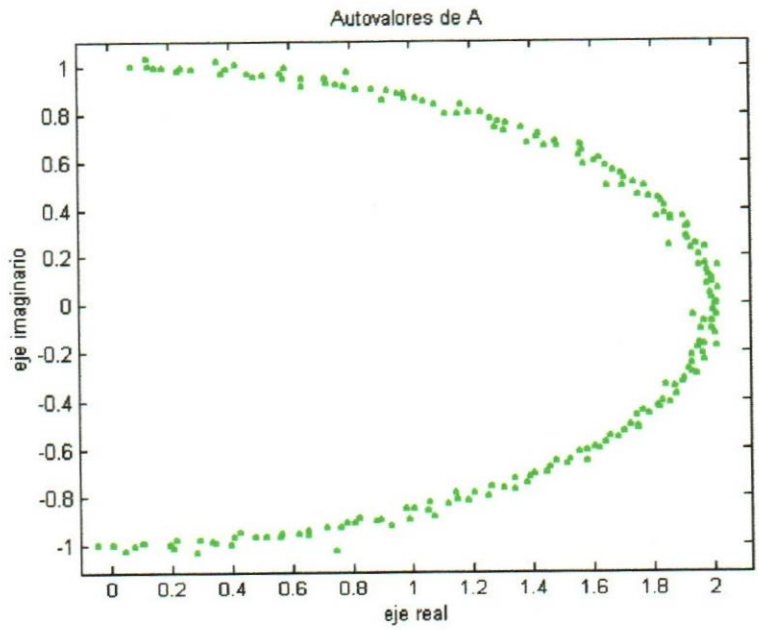

Figura 2.8: Autovalores

Arnoldi después de 30 iteraciones no ha logrado acercarse lo suficiente a ningún autovalor retrasando así la convergencia del GMRES

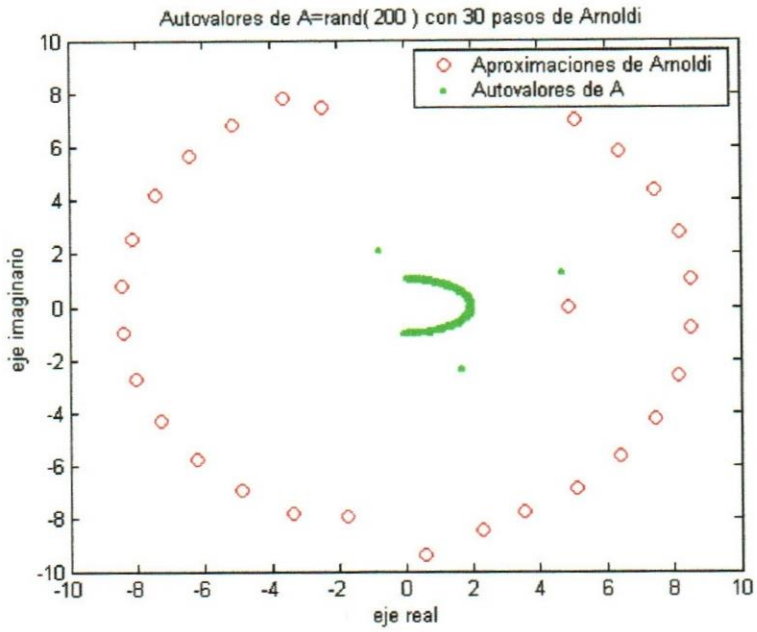

Figura 2.9: Método de Arnoldi y Autovalores

La norma del error del GMRES de puede visualizar después de 70 iteraciones:

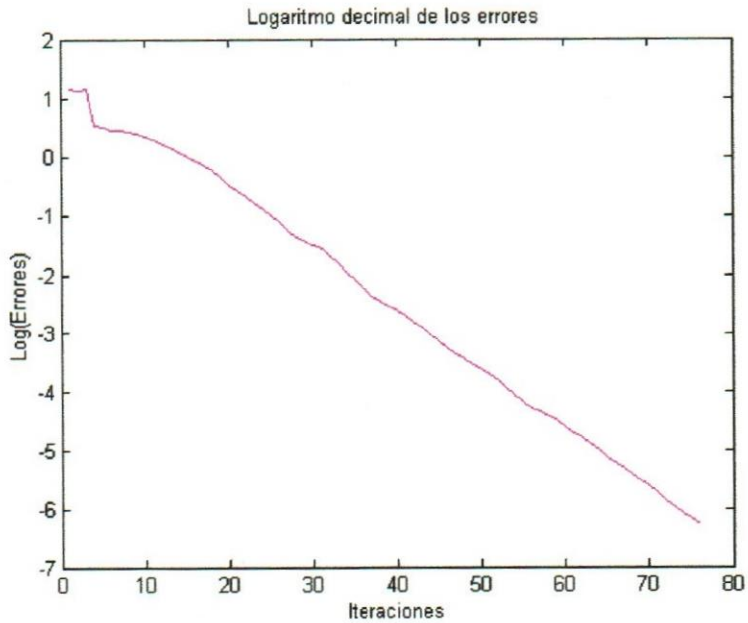

Figura 2.10: Convergencia del GMRES

La convergencia comparada con el problema anterior es mucho más lenta 75 iteraciones para disminuir el error a 6.64×10^{-7}, el número de condición $\kappa(A) \approx 4.32$. y $\kappa(V) \approx 54.0$

por lo que el deterioro de la convergencia no es explicado por los números de condición anteriores sino por la ubicación de los autovalores.

Técnicas de Precondicionamiento

Un precondicionador es cualquier forma de modificación implícita o explícita de un sistema lineal original que hace que sea más fácil de resilber un método iterativo. Por ejemplo, escalar todos las filas del sistema lineal para hacer que los elementos diagonales sean $1'nos$ es una forma explícita de precondicionar. El sistema resultante se puede resolver por métodos iterativos y probablemente requiera menos pasos para alcanzar la convergencia que el sistema original. (aunque esto no esta garantizado a priori).

Como hemos visto anteriormente, la convergencia de los métodos iterativos depende de las características espectrales de la matriz de coeficientes. Por lo tanto, de puede intentar una transformación del sistema lineal en otro equivalente de forma que posea la misma solución, pero que tenga propiedades espectrales más favorables. En general la idea de precondicionamiento es muy simple, dado el sistema $Ax = b$ para alugna matriz M invertible de igual tamaño que A nosotros podemos hacer

$$M^{-1}Ax = M^{-1}b \tag{2.30}$$

sistema que posee la misma solución que el inicial, ahora la convergencia de los métodos iterativos dependen de las características de la matriz $M^{-1}A$ en vez de las de A, que de acuerdo a la elección de M serán mejores o peores que las de A.

En la figura se muestra un ejemplo de la distribución de autovalores para la matriz A^* y para la matriz $M^{-1}A^*$ resultado de precondicionarla tomando como $A^* = A + D$ como en el ejemplo N° 2. La matriz que se usó para precondicionar fue la matriz triangular inferior de A^* a continuación mostramos los autovalores del sistema precondicionado:

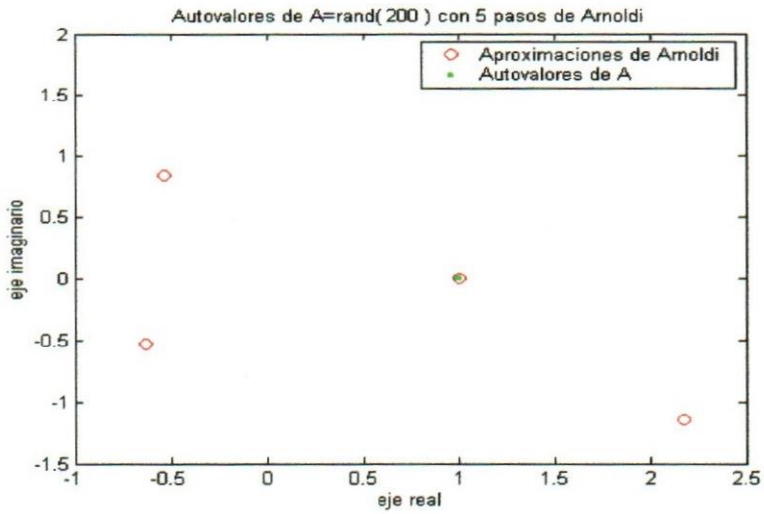

Figura 2.11: Algoritmo de Arnoldi y Autovalores

El sistema precondicionado *sólo tiene un autovalor* $\lambda_1 = \lambda_2 = \cdots \lambda_{200} = 1 + 0i$. En cuanto a la convergencia del *GMRES* para el mismo problema anterior tenemos

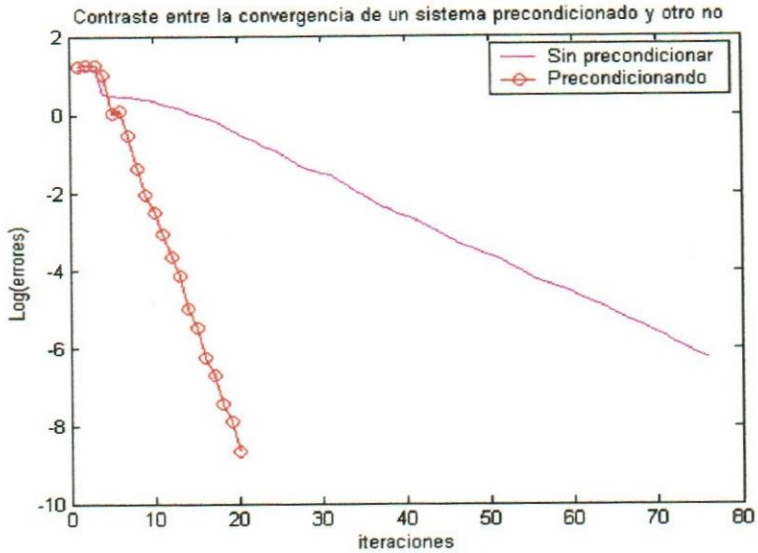

Figura 2.12: Mejorías del Sistema Precondicionado

Como vemos la convergencia es mucho mejor gracias a las características del sistema precondicionado, obteniendo un error inferior a 10^{-9} en sólo 20 iteraciones.

Iteraciones precondicionadas

Desde el punto de vista práctico, el único requerimiento de M es que el sistema lineal $Mz = r$ sea fácil de resolver. esto se debe a que los métodos iterativos precondicionados requeriran la solución de un sistema lineal con la matri M en cada iteración. Así reemplazamos el sistema original por el sistema precondicionado:

$$M^{-1}Ax = M^{-1}b \tag{2.31}$$

Sin embargo, la transformaión anterior del sistema lineal: $A = M^{-1}A$ no es la más común. Una fo rma más correcta de introducir el precondicionador es dividir el mismo como $M = M_1M_2$ y transformar el sistema a

$$M_1^{-1}AM_2^{-1}(M_2x) = M_1^{-1}b \tag{2.32}$$

Las matrices M_1 y M_2 se llaman precondicionador izquierdo y derecho respectivamente.

Un método iterativo puede precondicionarse de acuerdo al siguiente esquema:

1. Se transforma el vector término independiente $b = M_1^{-1}b$
2. Se aplica el método iterativo sin precondicionar, pero reemplazando la matriz de coeficientes A por $M_1^{-1}AM_2^{-1}$ y la solución de ese sistema la llamamos y
3. Se calcula $x = M_2^{-1}y$

El punto teórico importante es que la matriz de coeficientes transformada $M_1^{-1}AM_2^{-1}$ preserva algunas propiedades de la matriz A, por ejemplo, si A es positiva definida y $M_1 = M_2^T$; entonces la matriz de coeficientes transformada es igual simétrica y positiva definida.

Mostraremos brevemente las técnicas más populares usadas para precondicionar un sitema general. Muchos invetsigadores se han concentrado en el desarrollo de buenos precondicionadores. Existen 3 criterios a la hora de buscar un buen precondicionador M para la matriz A:

1. Usando el precondicionador M se debe reducir el número de iteraciones necesarias para la convergencia
2. M debe ser fácil de calcular. Es decir, el costo de construir el precondicionador debe ser pequeño en comparación con el costo total de resolución del problema.
3. El sistema $Mz = r$ debe ser fácil de resolver,

Una de las múltiples clasificaciones de los precondicionadores es aquella que los divide en: *algebraicos y multinivel*. Los algebraicos dependen unicamente de la estructura algenraica de la matriz de coeficientes A, se pueden calificar como a posteriori, ya que sólo dependen de la matriz de coeficientes y no de los detalles del proceso de construcción del sistema lineal. El escalado diagonal y las factorizaciones incompletas como las de *Cholesky* y LU son ejemplos típicos de precondicionadores algebraicos. Los precondicionadores multinivel son menos generales ya que dependen del conocimiento de la ecuación diferencial o el proceso de discretización que da lugar al sistema. Estos precondicionadores se califican como a priori, ya que dependen de la información sobre la construcción de la matriz de coeficientes antes que de la matriz en sí misma.

El mayor rendimiento se consigue con el precondicionador multinivel, ya que muchos de ellos consiguen una tasa de convergencia óptima. Además muchos precondionadores multinivel son fácilmente paralelizable. Sin embargo, requieren de un conocimiento a priori sobre la formación del sistema lineal, y esta información no siempre se conoce.

BICG y *CGS*

El GMRES retiene la ortogonalidad de los residuos usando recurrencia a costa de un al to requerimiento de memoria. Hay otros métodos como *BICG* (Bi-Conjugate Gradient), el *CGS* (Conjugate Gradient Squared) o el *BICGSTAB* (Bi-Conjugate Gradient Stabilized) que siguen aproximaciones distintas.

BICG

Las iteraciones del *BICG* construyen aproximaciones en el mismo subespacio de Krylov

que el *GMRES* pero más baratas en cada paso. La diferencia con el *GMRES* es que ahora la condición de orotgonalidad es ahora:

$$r_m \perp \left\{\widetilde{r_0}, A^T\widetilde{r_0}, (A^T)^2\widetilde{r_0}, \ldots, (A^T)^{m-1}\widetilde{r_0}\right\} \tag{2.33}$$

donde $\widetilde{r_0} \in C^n$ es un vector que normalmente se toma igual a r_0.

Basado en la notación de los métodos de proyección el *BICG* es un proceso de proyección sobre

$$K_m = gen\{v_1, Av_1, \ldots, A^{m-1}v_1\} \tag{2.34}$$

ortogonalmente sobre

$$L_m = gen\left\{w_1, A^T w_1, \ldots, (A^T)^{m-1} w_1\right\} \tag{2.35}$$

tomando usualmente $v_1 = r_0/\|r_0\|$. El vector w_1 es arbitrario, provisto de la propiedad de que $< v_1, w_1 > \neq 0$.

El método es derivado de una extensión para matrices no simétricas del algoritmo de Lanczos para matrices simétricas. Calcula las aproximación x_i usando $\bigcirc(1)$ operaciones vectoriales frente a las $\bigcirc(n)$ que necesita el *GMRES*. Además la demanda de memoria no aumenta con el número de iteraciones. El hecho de necesitar el uso de A^T puede resultar una desventaja para ciertas estructuras de datos dispersas, debido al indireccionamiento de las mismas.

CGS

Es una modificación del *BICG* que sólo requiere multiplicar por la matriz A . La idea detrás de este método es la búsqueda de una secuencia de iteraciones de tal forma que la norma del residuo en cada iteración satisfaga:

$$r_j = p_j^2(A)r_0 \tag{2.36}$$

en lugar de $r_j = p_j(A)r_0$ condición que cumple el *BIC*.

Una de las desventajas fundamentales del *CGS* es que su comportamiento a nivel de convergencia es bastante irregular, lo que puede dar lugar a una pérdida de presición al actualizar el residuo. Además tiende a diverger cuando la elección de la solución inicial esta próxima a la solución real

TMRES

En el GMRES de Saad y Schultz el residual $r = b - Ax$ es minimizado sobre un espacio de Krylov generado por la matriz A. La aproximación para residuales transformados TMRES (Transformed minimal residual approach) esta basada en la partición $A = S - T$, donde S es una matriz invertible y el radio espectral de $S^{-1}T$ es menor o igual que 1 ($\rho(S^{-1}T) \leq 1$). El residual transformado $S^{-1}r$ es minimizado sobre $x = x_0 + z$ con z en el

espacio de Krylov

$$K_m(S^{-1}T,g) = gen\left(g,(S^{-1}T)g,(S^{-1}T)^2g,\ldots,(S^{-1}T)^{m-1}g\right) \tag{2.37}$$

generado por $S^{-1}T$ y arrancando con $g = S^{-1}(b - Ax_0)$.

Aunque el esquema iterativo asociado $Sx_{m+1} = Tx_m + b$ converge lentamente cuando A es casi singular, nosotros observamos que $K_m(S^{-1}T,g)$ a menudo rinden una buena aproximación a la solución de $Ax = b$ para m relativamente pequeño. Teoricamente el espacio de Krylov generado por $S^{-1}T$ es el mismo que genera el GMRES tomando como precondicionador la matriz $S^{-1}A$. Numéricamente éstos espacios difieren porque las matrices $S^{-1}T$ y $S^{-1}A$ son diferentes.

El Algoritmo *TMRES*

1. $A = S - T$, S invertible, $(\rho(S^{-1}T) \leq 1)$
2. $g = S^{-1}(b - Ax_0)$, $v_1 = \frac{g}{\|g\|}$, $e = \vec{0}$, $e_1 = \|g\|$
3. for $j = 1$: hallar convergencia
4. $\quad s = S^{-1}(Tv_j)$
5. $\quad$ for $i = 1 : j$
6. $\quad\quad h_{ij} =< s, v_i >$
7. $\quad\quad s = s - v_i h_{ij}$
8. $\quad$ end
9. $\quad h_{j+1,j} = \|s\|$
10. $\quad v_{j+1} = \frac{s}{h_{j+1,j}}$
11. $\quad h_{ii} = h_{ii} - 1$
12. $\quad$ for $i = 1 : j-1$
13. $\quad\quad H(i : i+1,j) = Q_i H(i : i+1,j)$
14. $\quad$ end
15. $\quad Q_j = Givens(H(j : j+1,j))$
16. $\quad H(j : j+1,j) = Q_j H(j : j+1,j)$
17. $\quad e(j : j+1) = Q_j e(j : j+1)$
18. end
19. $x_j = x_0 + V_j(H(1 : j, 1 : j))^{-1} e(1 : j)$

Posibles cambios para S con la propiedad de $\rho(S^{-1}T) \leq 1$

1. Si A es simétrica y positiva definida, entonces $S = L$ donde L es la matriz triangular inferior de A (cambio de Gauss-Seidel)
2. Si A es diagonal dominante, entonces $S = L$ (L por Gauss-Seidel) o $S = D$ donde D es la matriz diagonal de A.

Comparación entre el GMRES, TMRES

Para tener una idea aún más clara de las ventajas de cada uno de los métodos de proyección en la resolución de Sistemas de Ecuaciones Lineales presentaremos 6 situaciones y mostraremos el comportamiento de cada uno de éstos métodos. Para ellos observaremos el comportamiento de los residuales hasta alcanzar la tolerancia exigida $tol = 10^{-7}$. Se tomaron éstos dos métodos por ser el TMRES el más novedoso y el GMRES uno de los métodos de mayor eficiencia y de comprobada eficacia.

Primer Caso $A = \mathbf{rand(100)} + 2I_{100}$

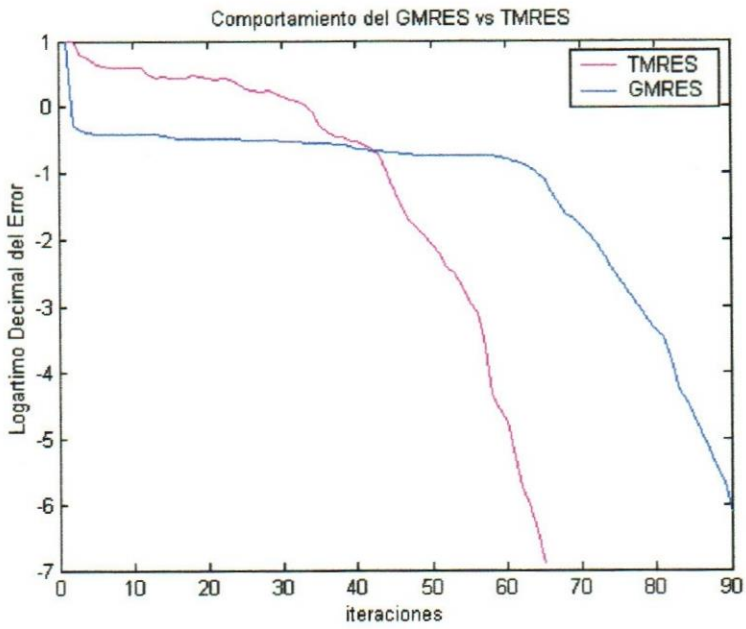

Figura 2.13

La distribución de los autovalores de este sistema ya fue analizada, puede observarse como el GMRES es menos eficiente y necesita de muchas más iteraciones para alcanzar la solución para este sistema de ecuaciones. Recordemos que la gráfica muestra el logarítmo decimal de la norma del vector residual a medida que los métodos desarrollan más iteraciones. El TMRES es más eficiente porque la matriz en cuestión es diagonal dominante y la elección de la matriz $S = diag(A)$ garantiza que se cumpla la condición $\rho(S^{-1}T) \leq 1$ y por ende el mejor performance de éste método sobre el otro.

Segundo Caso: $A = A^* + D$ **con** $A^* = 2I_{100} + 0,5\frac{\text{rand}(100)}{\sqrt{100}}$ **y** D **una matriz diagonal con entradas complejas** $d_k = (-2 + 2\sin\theta_k) + i\cos\theta_k \quad \theta_k = \frac{k\pi}{99}$, $0 \leq k \leq 99$

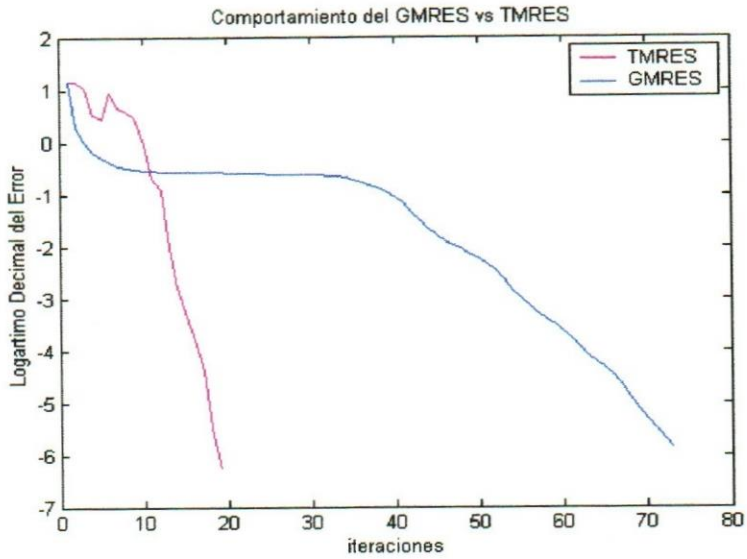

Figura 2.14: Matriz Diagonal para el TMRES

En este caso se presentan los dos comportamientos del TMRES tomando $S = lower(A)$ la matriz triangular inferior de A y $A = diag(A)$ la matriz cuya diagonal son los elementos de la diagonal principal de A. Dado que el sistema es diagonal dominante la elección de la matriz diagonal como partición para el TMRES es más eficiente que la matriz triangular inferior. Esto se evidencia en el comportamiento de uno u otro planteamiento.

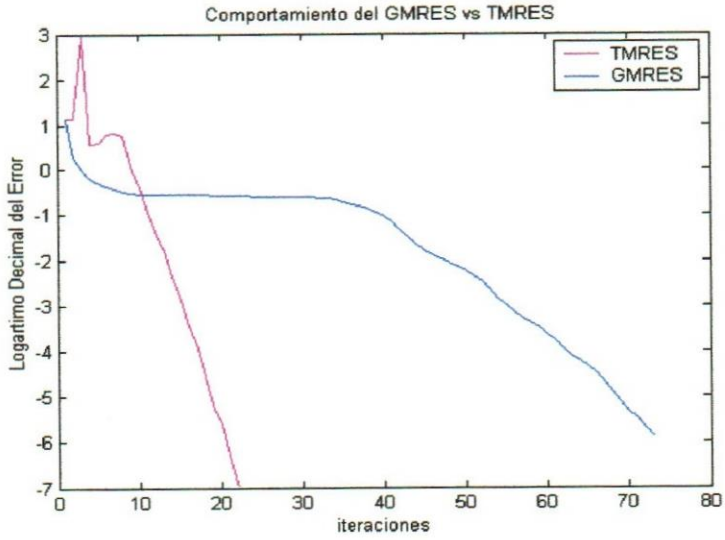

Figura 2.15: Matriz Triangular Inferior para el TMRES

Aunque el TMRES termina siendo más eficiente es importante notar como las primeras iteraciones del método no mejoran la aproximación a la solución por el contrario se alejan de ella mientras que el GMRES aunque tarda más todas sus iteraciones fueron de pendiente negativa, es decir, siempre hubo mejoría en las soluciones calculadas

Tercer Caso $A = \mathbf{rand}(100) - 2I_{100}$

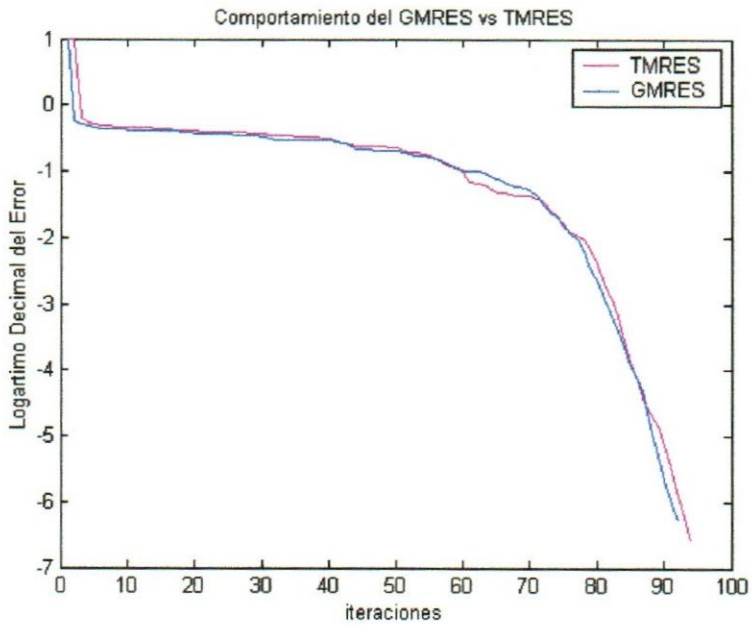

Figura 2.16

En esta situación el TMRES no demuestra su superioridad sobre el GMRES, recordemos que el TMRES necesita de una partición tal que se cumpla la condición $\rho(S^{-1}T) \leq 1$ los cambios elementales, no funcionan y su comportamiento es similar al GMRES. El siguiente caso evidencia que la búsqueda de esta partición no es tarea fácil y que puede no apartarnos de la solución del sistema

Cuarto Caso $A = \mathbf{rand}(100)$

Como se puede apreciar el TMRES no converge a la solución mientrás el GMRES necesita de casi todos los elementos de la base del espacio de Krylov necesarios para hallar la convergencia. Recordemos que el GMRES converge a lo más en $m = \dim(A)$ osea si el espacio donde pertenece la solución tiene dimensión 100 el GMRES converge en un máximo de 100 iteraciones

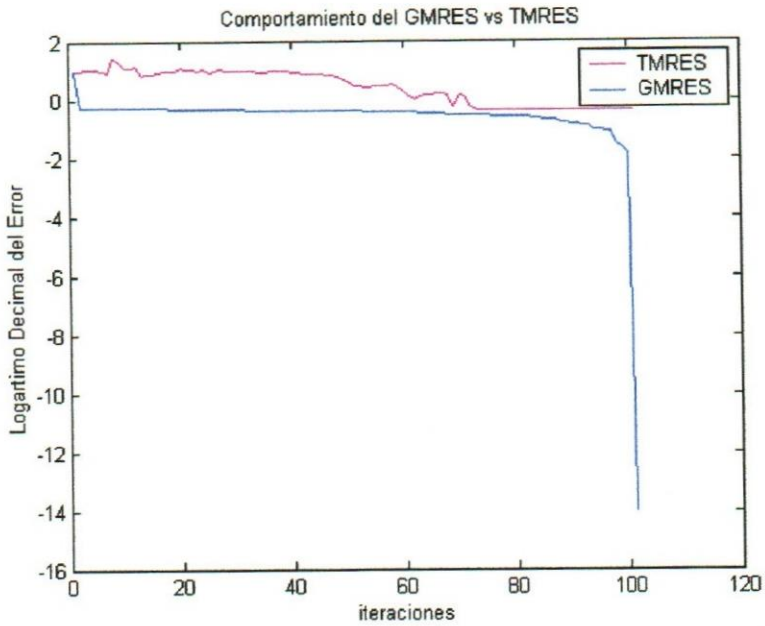

Figura 2.17

Quinto Caso A de la forma siguiente:

$$A = \begin{cases} a_{i,i} = 2 \\ a_{i,i+1} = -1 \\ a_{i,i-1} = -1 \\ \textbf{0 en otro caso} \end{cases}$$

Este ejemplo es particularmente importante ya que al resolver *numéricamente sistemas de ecuaciones diferenciales* las matrices que normalmente se encuentran son matrices simétricas, lo que induce a pensar en el método de Gradiente Conjugado como una alternativa, hay que recordar que además de simetría este método exige que la matriz sea Positivo Definida, aunque esta matriz es simétrica no cumple la segunda condición es por ello que el método tiene un comportamiento tan irregular sin embargo converge a la solución en el mismo número de pasos del GMRES, el TMRES simplemente necesita de muhcas más iteraciones para poder acceder a la solución buscada.

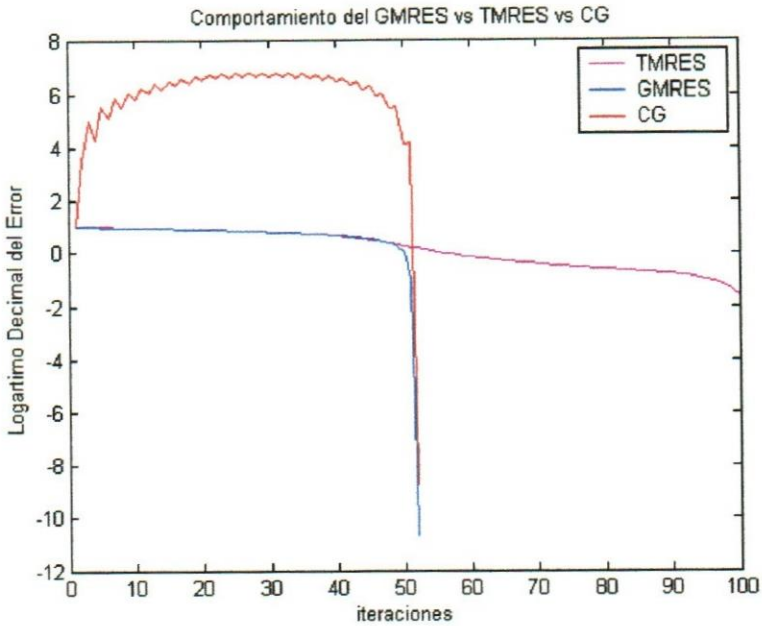

Figura 2.18

Sexto y Último Caso A de la siguiente forma:

$$A = \begin{cases} a_{i,i} = 4 - k \\ a_{i,i+1} = -1 \\ a_{i,i+4} = 1 \\ a_{i,i-1} = -1 \\ a_{i,i-4} = 1 \\ 0 \text{ en otro caso} \end{cases}$$

$k = 2$

La matriz vuelve a ser simétrica más no positivo definida, recordemos que esta condición es fuerte:

$$\langle Au, u \rangle > 0 \ \forall \, u \in \Re^k$$

aunque no cumple con serlo el método termina por converger de una forma muy irregular el GMRES mantiene su regular convergencia.

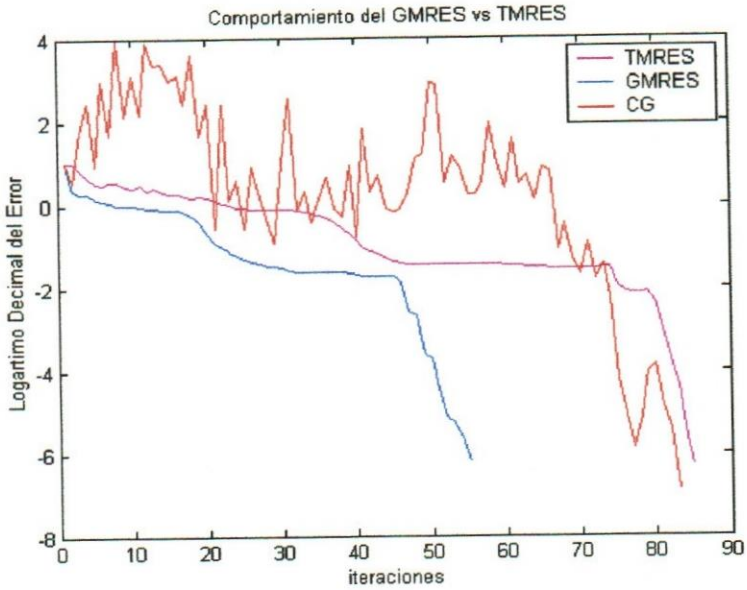

Figura 2.19

$k = -2$

Aunque la matriz es similar a la anterior, simétrica y no positiva definida, sólo hemos cambiado la diagonal principal, el método de Gradiente Conjugado tiene un mejor comportamiento que el GMRES y el TMRES. Con lo cual se evidencia una de las características de los métodos de proyección: la pertinencia de uno u otro depende únicamente de las características del sistema y aunque teóricamente un método tenga ventajas sobre otro en la práctica pueden ser igualmente competitivos, como en este caso.

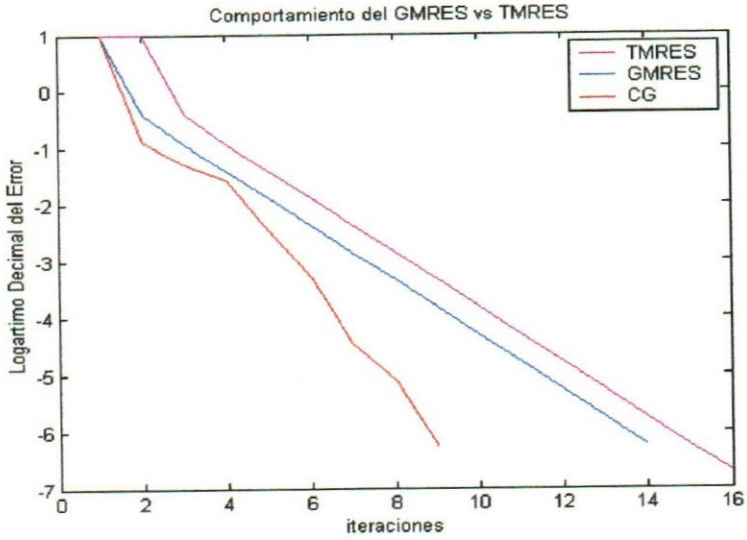

Figura 2.20

Conclusiones

1. La necesidad de los métodos de proyección es evidente, resolver un sistema de ecuaciones lineales de 100 o más variables es una tarea muy compleja para los métodos convencionales de resolución de sistemas de ecuaciones.

2. Es necesario estudiar previamente las condiciones espectrales de la matriz de coeficientes del sistema de manera de poder implementar particiones adecuadas para implementar el TMRES y acceder a soluciones más eficientes.

3. El GMRES garantiza la convergencia del método en $m = \dim(A)$ característica de la cual adolece el TMRES al no encontrar una partición que garantice $\rho(S^{-1}T) \leq 1$ esta condición es muy fuerte. Sin embargo los cambios elementales: Gauss - Seidel y Jacobi funcionan bastanate bien para muchos ejemplos.

4. En todos los ejemplos desarrollados el GMRES demostró ser monótonamente convergente $\|r_{m+1}\| \leq \|r_m\|$ aunque el número de iteraciones necesarias puede ser inferior utilizando otro método éste puede no garantizar la regularidad demostrada por este método.

5. Las Técnicas de Precondicionamiento son sin lugar a duda un área de gran importancia a nivel investigativo ya que las características espectrales (la distribución de los autovalores) puede ser radicalemente mejor facilitando así la implementación de los métodos de proyección.

6. El CG impone condiciones muy fuertes para ser aplicado, sin embargo obviar la condición de Positivo Definida puede llevar a resultados satisfactorios, sobre todo si el sistema es Diagonal Dominante. Además chequear la propiedad de ser positiva definida para una matriz no es sencillo, a veces imposible. Sin emabrgo, no podemos olvidar que así como el GMRES garantiza convergencia monótona el CG también lo hace bajo los supuestos ya estudiados.

7. Hemos conjugado el Álgebra Lineal, el Cálculo Numérico y la Computación para desarrollar métodos eficientes y necesarios para resolver grandes Sistemas de Ecuaciones Lineales.

8. El vínculo de los autovalores y el Algortimo de Arnoldi es sin duda uno de los vínculos más importantes entre los métodos de proyección, el estudio de las Ecuaciones Diferenciales y la predicción del comportamiento de éstos métodos en la resolución de los sistemas de Ecuaciones.

9. Próximos trabajos se concentrarán en el estudio de Precondicionadores y en su relación con los planteamientos establecidos en la Simulación de Yacimientos Petroleros para resolver los Sistemas de Ecuaciones Diferenciales y sus consecuentes Sistemas Lineales.

Bibliografía

1. *Coughran W. y Freund R.*, **Recent Advance in Krylov-Subespace Solver for Linear Systems and Application in Device Simulation**, Bell Labs Lucent Tecnologies

2. *Grossman S.*, **Álgebra Lineal**, Mc Graw Hill, pp. 536 - 556 Año 1977.

3. *Hager W.*, **Iterative Methods for Nearly Singular Linear Systems**, Society for Industrial and Applied Mathematics, Vol 2, N° 2, pp. 747 - 766, Año 2000.

4. *Nieves A. y Domínguez F.*, **Métodos Numéricos Aplicados a la Ingeniería**, Editorial CECSA, pp. 125 - 216, Año 1995.

5. *Saad Y.*, **Iterative Methods for Sparse Linear Systems**, An International Thomson Publishing Company, pp. 122 - 200, Año 1997.

6. *Scheid F.*, **Análisis Numérico Teoría y 755 problemas resueltos**, Mc Graw Hill, pp. 334 - 356, Año 1972.

7. *Sturler E.*, **Truncation Strategies for optimal Krylov Subespaces Methods**, Society for Industrial and Applied Mathematics, Vol. 36 pp. 864 -889, Año 1999.

8. *Trefethen L. y Bau D.*, **Numerical Linear Algebra**, Society for Industrial and Applied Mathematics, pp. 179 - 275, Año 1989.