



*Facultad
de
Ciencias*

**ALGUNOS ALGORITMOS ITERATIVOS EN
ÁLGEBRA LINEAL NUMÉRICA**

**(SOME ALGORITHMS IN NUMERICAL LINEAR
ALGEBRA)**

GRADO EN MATEMÁTICAS

Autor: Laura Uslé Laza

Director: Carlos Beltrán Álvarez

Junio, 2016

Agradecimientos: al profesorado del Grado en Matemáticas, Administración del Departamento y a la Facultad por su ayuda en la premura para facilitar los trámites que han permitido presentar este trabajo a tiempo. Al director del trabajo y a todas aquellas personas que han confiado en la realización de este trabajo y me han ayudado, porque sin ellas no lo hubiese logrado.

RESUMEN

En este trabajo estudiamos algunos de los grandes problemas a los que se enfrenta el Álgebra Lineal Numérica, como el cálculo de los valores propios de una matriz, así como la resolución de sistemas de ecuaciones mediante distintos algoritmos. Se realiza un estudio sobre diversos algoritmos, implementándolos en Matlab. Para el análisis de resultados tomamos matrices aleatorias de tamaño 100, 500 y 1000 tanto para matrices reales como complejas, simétricas/hermíticas o no. Concluimos con una serie de resultados que consolidan lo estudiado en la teoría.

SUMMARY

In this work we deal with some of the most important problems in Numerical Linear Algebra, such as the computation of eigenvalues and the solution of linear systems of equations, using different approaches. We describe different algorithms, and program them in Matlab. For the analysis of the results we use collections of random matrices of sizes 100, 500 and 1000, and we experiment with real and complex matrices, both in the case that they are symmetric (hermitian) or non-symmetric (non-hermitian). Our conclusions support the theoretical properties of the algorithms.

ÍNDICE

	Págs
1. INTRODUCCIÓN.....	5-6
2. CONCEPTOS PREVIOS	7-11
3. RESULTADOS TEÓRICOS RELEVANTES.....	12-16
4. FACTORIZACION QR.....	17-21
5. MÉTODO DE LA POTENCIA.....	22-23
6. MÉTODO DE LA POTENCIA INVERSA.....	24
7. MÉTODO DEL COCIENTE RAYLEIGH.....	25-26
8. MÉTODO DE RESIDUOS MÍNIMOS GENERALIZADO (GMRES)	27-29
9. MÉTODO DE LANCZOS.....	30
10. ANÁLISIS DE RESULTADOS.....	31-39
11. NOTA FINAL.....	40
12. BIBLIOGRAFÍA.....	41

1. INTRODUCCIÓN

El presente trabajo se centra en el área del Álgebra Lineal Numérica. En esta área de las Matemáticas se estudian diversos algoritmos para realizar cálculos de Álgebra Lineal. Además, esta rama intenta explicar principios fundamentales y simplificar cálculos en matemáticas, física, economía, estadística... algunas de estas aplicaciones aparecerán a lo largo del estudio realizado en el presente trabajo.

El estudio que hemos realizado en este trabajo incluyen los problemas más frecuentes a los que se enfrenta esta área, como es el caso del cálculo de valores propios, factorización QR y la resolución de sistemas de ecuaciones.

El principal objetivo en nuestro trabajo es analizar, estudiar y programar diversos métodos que dividiremos en secciones, las cuales se mostrarán más adelante. Además la sección de conclusiones estará formada por tablas comparativas con la finalidad de analizar los resultados y poder sacar conclusiones que muestren y corroboren los resultados teóricos que hemos expuesto con anterioridad.

A continuación, pasaremos a mencionar y mostrar una pequeña reseña bibliográfica sobre algunos de los principales autores que han tenido especial relevancia sobre los métodos y resultados que veremos durante el desarrollo del trabajo.

Fue David Hilbert (1862-1943), matemático-alemán, reconocido como uno de los más influyentes del siglo XIX y principios del XX, quien utilizó por primera vez la vez la palabra 'eigen' (propio) en el contexto de resaltar la importancia del cálculo de los valores propios.

Destaca William Strutt Rayleigh (1842-1919), físico – matemático procedente de Reino Unido conocido principalmente por su aportación al dar el primer valor numérico al número de Avogadro y sus importantes aportaciones matemáticas, a pesar de que su trayectoria se enfocó más hacia el ámbito físico. Una de sus aportaciones matemáticas fue el algoritmo del Cociente de Rayleigh. Es por esto, que el algoritmo lleva su nombre.

El algoritmo de Gram-Schmidt recibe este nombre en honor a Jorgen Pedersen Gram y Erhard Schmidt (1876-1959), siendo este último alumno de David Hilbert.

Respecto al cálculo de los valores propios que obtiene el método de Arnoldi, se denominan también valores de Ritz, en honor a su descubrimiento Walther Ritz (1878-1909), físico suizo que falleció debido a la enfermedad de tuberculosis.

Actualmente y como desarrollaremos a lo largo del trabajo, sabemos que el método de Lanczos es una modificación del método de Arnoldi para el caso en que las matrices sean hermíticas. Fue el húngaro Cornelius Lanczos (1893-1974) físico teórico,

analista numérico e inventor de varios métodos numéricos quien elaboró este algoritmo que lleva su nombre en su honor. Además, cabe destacar que Lanczos fue asistente de Albert Einstein en Alemania.

El trabajo ha requerido un gran esfuerzo de comprensión, análisis y programación de varios métodos. La parte más extensa se corresponde con entender y analizar en qué consiste cada método y las diferencias que presentan unos métodos frente a otros. Sin embargo, la parte más laboriosa del presente trabajo ha sido la programación en Matlab, el estudio de los algoritmos que aparecen el libro que se menciona en la bibliografía, puesto que teníamos que entender muy bien cómo funciona el método para poder transcribir el algoritmo a código Matlab, así como el análisis de resultados. Cabe mencionar, que a la hora de realizar el análisis de resultados nos hemos apoyado en algunos consejos que aparecen en el libro en temas referentes a convergencia en algunos métodos, por ejemplo. No obstante, aunque el trabajo haya sido laborioso, hemos sentido una gran satisfacción al obtener los resultados deseados que son coherentes con la teoría que habíamos estudiado y planteado con anterioridad.

El desarrollo del trabajo nos ha servido para evolucionar en el proceso de hacer estudios más detallados y sobre todo para enfrentarnos a errores en la programación y poder solucionarlos con más o menos facilidad.

2. CONCEPTOS PREVIOS

Comenzamos el desarrollo del trabajo con una serie de definiciones básicas que, si bien el lector con seguridad conocerá, nos permitan establecer unas notaciones comunes a todo el trabajo. El conocimiento de estos conceptos es imprescindible para poder entender el resto del trabajo.

En este pequeño glosario, aparecen tanto conceptos básicos, como definiciones de algún concepto que se mencionará en algún resultado teórico o en la explicación de alguno de los métodos que expondremos más adelante.

Definición 1 (producto matriz-vector):

Sea $x \in \mathbb{R}^n$ vector columna, $A \in \mathbb{R}^{m \times n}$ matriz. Así el producto matriz-vector $b = A \cdot x$ se definen de la siguiente manera:

$$b_i = \sum_{j=1}^n a_{ij} x_j \quad i = 1, \dots, m$$

Así sea a_j la j-ésima columna de A. Podemos reescribirlo como:

$$b = A \cdot x = \sum_{j=1}^n x_j a_j$$

Definición 2 (producto matriz-matriz):

El producto matriz-matriz, $B = A \cdot C$, en él cada columna de B es una combinación lineal de las columnas de la matriz A. Sea $A \in \mathbb{C}^{l \times m}$ y $C \in \mathbb{C}^{m \times n}$, entonces $B \in \mathbb{C}^{l \times n}$ y queda definido por:

$$b_{ij} = \sum_{k=1}^m a_{ik} c_{kj}$$

Siendo b_{ij} , a_{ik} y c_{kj} elementos de B, A y C respectivamente.

Definición 3 (Rango de una matriz)

El rango de una matriz A, es la dimensión del conjunto de vectores que se pueden expresar como $A \cdot x$ para algunos x. Es decir, el rango es el número máximo de columnas que son linealmente independientes, y es un resultado fundamental que este número iguala al de filas linealmente independientes de A

Definición 4 (Núcleo de una matriz)

El núcleo de una matriz $A \in \mathbb{C}^{m \times m}$, se denota por $\text{Ker}(A)$, y es el conjunto de todas las soluciones de la ecuación homogénea $A \cdot x = 0$.

Definición 5 (Matriz inversa)

Dada una matriz cuadrada A de orden m , $A \in \mathbb{C}^{m \times m}$ de rango m , existe su inversa X , tal que:

$$A \cdot X = I_{m \times m}$$

Esto es, el producto de A por su inversa produce la matriz identidad. Equivalentemente,

$$e_j = \sum_{i=1}^m x_{ij} a_i$$

Siendo e_j el j -ésimo vector de la base canónica.

Definición 6 (Conjugado)

El conjugado de un escalar z , se denota por \bar{z} o z^* , y se obtiene al negar la parte imaginaria. Así si $z \in \mathbb{R}$, $\bar{z} = z$.

Definición 7 (Hermítico adjunto o hermítico conjugado)

El hermítico adjunto o hermítico conjugado de una matriz $A \in \mathbb{C}^{n \times n}$, se denota por A^* y es el resultado de tomar el complejo conjugado de cada elemento y después permutando las filas por columnas o viceversa. Si $A = A^*$ se dice que es hermítico. Así, por la definición, una matriz hermítica debe ser cuadrada.

Si $A \in \mathbb{R}^{n \times m}$ denotamos $A^* = A^T$ y la llamamos matriz traspuesta.

Definición 8 (Producto interno)

El producto interno de dos vectores columnas $x, y \in \mathbb{C}^n$ es el producto del adjunto de x por y :

$$y^* \cdot x = \sum_{i=1}^m \bar{y}_i x_i$$

La norma de un vector es $\|x\| = \sqrt{x^* \cdot x}$

Definición 9 (vectores ortogonales y ortonormales)

Un par de vectores x, y son ortogonales si $x^* \cdot y = 0$. Un conjunto de vectores es ortonormal si es ortogonal y además todos ellos tienen norma igual a 1.

Definición 10 (Matriz unitaria)

Sea $Q \in \mathbb{C}^{m \times m}$, una matriz. Decimos que Q es unitaria (ortonormal en el caso real) si $Q^* = Q^{-1}$, es decir, si $Q^* \cdot Q = I$.

Definición 11 (Norma)

Una norma es una función, $\|\cdot\|: \mathbb{C}^m \rightarrow \mathbb{R}$ que asigna a cada valor real la longitud de cada vector. Toda norma debe cumplir las siguientes condiciones, para todos los vectores $x, y \forall \alpha \in \mathbb{C}$,

$$(1) \|x\| \geq 0 \text{ y } \|x\| = 0 \Leftrightarrow x = 0$$

$$(2) \|x + y\| \leq \|x\| + \|y\| \text{ (Desigualdad Triangular)}$$

$$(3) \|\alpha \cdot x\| = |\alpha| \cdot \|x\|$$

Definición 12 (Normas matriciales inducidas por normas vectoriales)

Dada una norma vectorial $\|\cdot\|_{(n)}$ y $\|\cdot\|_{(m)}$, $A \in \mathbb{C}^{m \times n}$, la norma matricial inducida, $\|A\|_{(m,n)}$ es el menor número C que cumple la siguiente inecuación $\forall x \in \mathbb{C}^n$:

$$\|A \cdot x\|_{(m)} \leq C \cdot \|x\|_{(n)}$$

En otras palabras:

$$\|A\|_{(m,n)} = \sup_{x \in \mathbb{C}^n, x \neq 0} \frac{\|A \cdot x\|_{(m)}}{\|x\|_{(n)}} = \sup_{x \in \mathbb{C}^n, \|x\|=1} \|A \cdot x\|_{(m)}$$

Definición 13 (Desigualdad de Hölder y de Cauchy-Schwarz)

Sea p y q tal que $\frac{1}{p} + \frac{1}{q} = 1$, $1 < p, q \leq \infty$. Así, la desigualdad de Hölder nos dice que $\forall x, y$,

$$|x^* \cdot y| \leq \|x\|_p \cdot \|y\|_q$$

Y la desigualdad de Cauchy-Schwarz es un caso especial, $p=q=2$,

$$|x^* \cdot y| \leq \|x\|_2 \cdot \|y\|_2$$

En ambos casos en ciertas situaciones se puede dar la igualdad.

Definición 14 (Descomposición completa en valores singulares de A)

Sea m, n enteros positivos, y $A \in \mathbb{C}^{m \times n}$, una descomposición en valores singulares (completa) de A es una factorización $A = U \Sigma V^*$, donde $U \in \mathbb{C}^{m \times m}$ y $V \in \mathbb{C}^{n \times n}$ son unitarias y Σ es diagonal con entradas reales no-negativas, llamadas valores singulares de A .

Definición 15 (Proyector)

Un proyector es una matriz cuadrada P que satisface $P^2 = P$. Una matriz con esta propiedad se llama idempotente. Esta definición incluye tanto a los proyectores ortogonales como a los no ortogonales (ver Definición 17). Usamos el término proyector oblicuo en el caso de no ser ortogonal.

Definición 16 (Proyector complementario)

Sea P un proyector, entonces $I - P$ es también un proyector, porque también es idempotente,

$$(I - P)^2 = I - 2 \cdot P + P^2 = I - P$$

La matriz $I - P$ se llama proyector complementario a P .

Definición 17 (Proyector ortogonal)

Geoméricamente, un proyector ortogonal es uno que proyecta sobre un subespacio S_1 según un subespacio S_2 , donde S_1 y S_2 son ortogonales.

Análíticamente, un proyector P es ortogonal si todos sus valores singulares son 0 o 1.

Definición 18 (valor propio)

Si $A \in \mathbb{C}^{n \times n}$, $\lambda_0 \in \mathbb{C}$ es un valor propio de A si existe un vector no nulo $x \in \mathbb{C}^n$ tal que $A \cdot x = \lambda_0 \cdot x$.

Definición 19 (espectro)

Al conjunto de valores propios distintos de A se le denomina espectro y se denota por $\sigma(A)$.

Definición 20 (vector propio)

Si $\lambda_0 \in \sigma(A)$, los vectores $x \neq 0$ para los que $A \cdot x = \lambda_0 \cdot x$ se llaman vectores propios asociados al valor propio λ_0 .

Definición 21 (polinomio característico)

Se llama polinomio característico de $A \in \mathbb{C}^{m \times m}$ y se denota como p_A con grado m al polinomio definido como: $p_A(z) = \det(z \cdot I - A)$

Definición 22 (Matriz Hessenberg superior e inferior)

Una matriz cuadrada A de dimensión $n > 1$ diremos que es Hessenberg superior si todos los elementos por debajo de la diagonal -1 son nulos. Recordamos que la diagonal $-k$ es la diagonal número k por debajo de la diagonal (principal).

Una matriz cuadrada A de dimensión $n > 1$ diremos que es Hessenberg inferior si todos los elementos por arriba de la diagonal -1 son nulos. Recordemos que la diagonal k es la diagonal número k por arriba de la diagonal (principal).

Definición 23 (Matriz no-defectuosa)

Sea $A \in \mathbb{C}^{n \times n}$ se dice no-defectuosa si y sólo si es diagonalizable.

Definición 24 (Polinomio mínimo)

Denotaremos por $p(A)$ al polinomio mínimo y es tal que $q(A) = h(x) \cdot p(A)$.

Definición 25 (Multiplicidad geométrica)

Para $\lambda_0 \in \sigma(A)$, denominamos la multiplicidad geométrica de λ_0 como la dimensión del subespacio propio de A asociado a λ_0 , y lo denotamos por $mg_A(\lambda_0)$, esto es

$$mg_A(\lambda_0) = \dim(\text{Ker}(\lambda_0 \cdot I_n - A))$$

Definición 26 (Multiplicidad algebraica)

Si λ_0 es una raíz de multiplicidad α del polinomio característico de A , decimos que λ_0 es un valor propio de orden α de A . A α se le llama multiplicidad algebraica de λ y suele lo denotamos por $ma_A(\lambda_0)$.

3. RESULTADOS TEÓRICOS RELEVANTES

En esta sección enunciaremos y demostraremos una serie de resultados teóricos básicos para poder analizar con mayor profundidad los métodos que estudiaremos en las siguientes secciones. No obstante, dentro de los siguientes apartados analizaremos otros resultados teóricos relevantes a esas secciones.

Teorema 1:

Todo valor propio de A es un cero del polinomio mínimo y todo cero de p_A es un valor propio de A .

Demostración:

\Leftarrow Sea λ_0 valor propio y v_0 el vector propio asociado a él, tenemos:

$$p_A(\lambda_0)v_0 = p_A(A)v_0 = 0 \cdot v_0 = 0$$

Deducimos, entonces que v_0 es no nulo, que $p_A(\lambda_0) = 0$.

\Rightarrow Por otro lado, como ya sabemos que todos los valores propios $\lambda_1, \dots, \lambda_p$ de A son ceros de p_A ,

$$p_A(\lambda) = (\lambda - \lambda_1)^{r_1} \dots (\lambda - \lambda_p)^{r_p} \cdot q(\lambda)$$

Supongamos que m_A tiene algún cero λ_0 que no es valor propio: $p_A(\lambda) = (\lambda - \lambda_1)^{r_1} \dots (\lambda - \lambda_p)^{r_p} (\lambda - \lambda_0) \cdot r(\lambda)$. Como λ_0 no es valor propio, $(A - \lambda_0 \cdot I)$ invertible, de donde:

$$0 = (A - \lambda_0 \cdot I)^{-1} \cdot p_A(A) = (A - \lambda_1 \cdot I)^{r_1} \dots (A - \lambda_p \cdot I)^{r_p} \cdot r(A)$$

Esto es, $(\lambda - \lambda_1)^{r_1} \dots (\lambda - \lambda_p)^{r_p} \cdot r(\lambda)$ es un anulador de A de grado menor que p_A , por lo cual llegamos a contradicción.

▪

Teorema 2:

Para $A \in \mathbb{C}^{n \times n}$ y $\lambda_0 \in \sigma(A)$, $ma_A(\lambda_0) \geq mg_A(\lambda_0)$.

Demostración:

Sea $m_g(\lambda_0) = r$ y $\{t_1, \dots, t_r\}$ una base del subespacio propio $S_{\lambda_0} = \text{Ker}(\lambda_0 \cdot I_n - A)$. Ampliamos esta base, $\{t_1, \dots, t_r, t_{r+1}, \dots, t_n\}$ para obtener una base de \mathbb{C}^n . Pongamos $T = [t_1, \dots, t_n]$ la matriz cuyas columnas son los vectores de la base escogida.

Sea $B = T^{-1} \cdot A \cdot T$. Esto es,

$$A \cdot T = [At_1 \dots At_r At_{r+1} \dots At_n] = [\lambda_0 t_1 \dots \lambda_0 t_r y_{r+1} \dots y_n]$$

$$= [t_1 \dots t_r t_{r+1} \dots t_n] \cdot \begin{bmatrix} \lambda_0 I_r & C \\ 0 & D \end{bmatrix} = T \cdot B$$

Luego, $\lambda \cdot I_n - B = \begin{bmatrix} (\lambda - \lambda_0) \cdot I_r & -C \\ 0 & \lambda I_{n-r} - D \end{bmatrix}$

Así, $\det(\lambda I_n - B) = (\lambda - \lambda_0)^r \cdot \det(\lambda I_{n-r} - D)$

Y como A y B tienen para cada valor propio las mismas multiplicidades algebraicas y geométricas y λ_0 podría ser valor propio de D , concluimos que $r = m_g(\lambda_0) \leq m_A(\lambda_0)$.

▪

Teorema 3:

Se dice que $A \in F^{n \times n}$ es no defectuosa si sólo si es diagonalizable

Demostración:

Es obvio que si A es diagonalizable, es no defectuosa porque la m_A de sus valores propios es 1.

Recíprocamente, si A es no defectuosa, entonces para cada $\lambda_0 \in \sigma(A)$ se tiene: $m_g(\lambda_0) = m_A(\lambda_0)$. supongamos que $\sigma(A) = \{\lambda_1, \dots, \lambda_s\}$ con $m_A(\lambda_i) = m_g(\lambda_i) = m_i, i = 1, \dots, s$

Esto es $A \cdot T = T \cdot D$ en las condiciones de la demostración del Teorema 2.

▪

Teorema 4:

- El producto de los valores propio de A es igual al determinante de A :

$$\lambda_1 \cdot \dots \cdot \lambda_n = \det(A)$$

- La suma de los valores propios de A es igual a la traza de A :

$$\lambda_1 + \dots + \lambda_n = \text{tr}(A)$$

Demostración:

Por un lado, el término independiente de un polinomio cualquiera de grado n con coeficiente principal 1 es igual a $(-1)^n$ multiplicado por el producto de sus n raíces.

$$p(x) = x^n + (\text{términos de grado entre } 1 \text{ y } n - 1) + a_0 = (x - \lambda_1) \cdot \dots \cdot (x - \lambda_n)$$

Así, $a_0 = (-\lambda_1) \cdot \dots \cdot (-\lambda_n) = (-1)^n \cdot \lambda_1 \cdot \dots \cdot \lambda_n$

Por otro lado, el polinomio característico de una matriz $n \times n$ tiene coeficiente principal igual a 1 y el término independiente igual al determinante multiplicado por $(-1)^n$ de la matriz, luego:

$$\det(A) = \lambda_1 \cdot \dots \cdot \lambda_n$$

Para comprobar la otra igualdad, el coeficiente del término $n-1$ grado, de un polinomio mónico cualquiera de grado n , es igual a menos la suma de sus raíces.

$$p(x) = (x - \lambda_1) \cdot \dots \cdot (x - \lambda_n) = x^n + x^{n-1} \cdot (-\lambda_1) + \dots + x^{n-1} \cdot (-\lambda_n)$$

Por otro lado, el polinomio característico de una matriz $n \times n$ tiene coeficiente principal igual a 1 y coeficiente del término del grado $n-1$ igual a -1 multiplicado por la traza de la matriz.

▪

Teorema 5:

Los valores propios de una matriz hermítica son todos reales

Demostración:

Sea A una matriz hermítica, esto es, una matriz tal que $A^* = A$ y sea λ un valor propio de A asociado al vector propio x . Se verifica entonces: $A \cdot x = \lambda \cdot x$.

Multiplicando esta última expresión por la izquierda por x^* tenemos que:

$$x^* \cdot A \cdot x = x^* \cdot \lambda \cdot x \quad (1)$$

Trasponemos y conjugamos:

$$(x^* \cdot A \cdot x)^* = (x^* \cdot \lambda \cdot x)^* \rightarrow x^* \cdot A^* \cdot x = \bar{\lambda} \cdot x^* \cdot x \rightarrow x^* \cdot A \cdot x = \bar{\lambda} \cdot x^* \cdot x \quad (2)$$

Al comparar (1) y (2): $x^* \cdot \lambda \cdot x = \bar{\lambda} \cdot x^* \cdot x$

Y como x es un vector propio no nulo, sabemos que $x^* \cdot x = \|x\|^2 \neq 0$, por tanto podemos dividir por $x^* \cdot x$ para obtener $\lambda = \bar{\lambda}$, es decir $\lambda \in \mathbb{R}$.

▪

El siguiente resultado que vamos a mencionar de un modo menos formal, es muy importante puesto que vamos a tenerlo en cuenta y es un paso fundamental para la ejecución de los métodos que mencionaremos posteriormente y el estudio que realizaremos, ya que entrará en juego en el tiempo de ejecución que presenten los siguientes algoritmos.

Este resultado a nos dice que antes de proceder a calcular los valores propios hay que reducir la matriz a forma Hessenberg. En caso de que la matriz sea hermítica, la forma Hessenberg es tridiagonal.

La matriz de Hessenberg de una matriz A tiene la siguiente estructura:

$$H = \begin{bmatrix} X & X & X & X & X & X \\ X & X & X & X & X & X \\ & X & X & X & X & X \\ & & X & X & X & X \\ & & & X & X & X \\ & & & & X & X \end{bmatrix}$$

Esta matriz se define como $H = Q^* \cdot A \cdot Q$, tal que Q es una matriz unitaria y H es la matriz en forma de Hessenberg.

La idea para construir la matriz H se basa en la introducción de ceros por debajo de la subdiagonal.

$$A = \begin{bmatrix} X & X & X & X & X & X \\ X & X & X & X & X & X \\ X & X & X & X & X & X \\ X & X & X & X & X & X \\ X & X & X & X & X & X \end{bmatrix} \rightarrow Q_1^* \cdot A = \begin{bmatrix} X & X & X & X & X & X \\ * & * & * & * & * & * \\ 0 & * & * & * & * & * \\ 0 & * & * & * & * & * \\ 0 & * & * & * & * & * \end{bmatrix} \rightarrow Q_1^* \cdot A \cdot Q_1 =$$

$$= \begin{bmatrix} X & \cdot & \cdot & \cdot & \cdot & \cdot \\ * & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & \cdot & \cdot & \cdot & \cdot & \cdot \end{bmatrix}$$

Se debe encontrar una matriz Q_1^* tal que al multiplicar la matriz A cree ceros bajo la subdiagonal y al premultiplicarse por Q_1 mantenga los ceros ya creados. Repitiendo este proceso $m-2$ veces se logra obtener la matriz Hessenberg, como resultado de:

$$Q_{m-2}^* \cdot \dots \cdot Q_2^* \cdot Q_1^* \cdot A \cdot Q_1 \cdot Q_2 \cdot \dots \cdot Q_{m-2} = Q^* \cdot A \cdot Q$$

ALGORITMO HESSENBERG

for $k = 1 : m - 2$

$x = A_{k+1:m,k}$

$v_k = \text{sign}(x_1) \cdot \|x\|_2 \cdot e_1 + x$

$$v_k = \frac{v_k}{\|v_k\|_2}$$

$$A_{k+1:m,k:m} = A_{k+1:m,k:m} - 2 \cdot v_k \cdot (v_k^* \cdot A_{k+1:m,k:m})$$

$$A_{1:m,k+1:m} = A_{1:m,k+1:m} - 2 \cdot (A_{1:m,k+1:m} \cdot v_k) \cdot v_k^*$$

SUBESPACIOS DE KRYLOV

En los métodos iterativos, las sucesivas aproximaciones del sistema $A \cdot x = b$, vienen dadas por una relación de recurrencia de la forma,

$$x_{k+1} = x_k + B^{-1} \cdot (b - A \cdot x_k)$$

O equivalentemente,

$$B \cdot x_{k+1} = C \cdot x_k + b, \text{ siendo } C = B - A$$

Los subespacios de Krylov son los subespacios generados por los vectores $b, A \cdot b, A^2 \cdot b, \dots$

Por ejemplo, $K(A, b) = \langle b, A \cdot b, A^2 \cdot b, A^3 \cdot b, \dots, A^i \cdot b \rangle$ es un subespacio de Krylov asociado a (A, b) de orden i .

Los subespacios de Krylov se utilizan en muchos métodos iterativos, como veremos en otras secciones más adelante.

4. FACTORIZACIÓN QR

A) FACTORIZACIÓN QR REDUCIDA

En muchas aplicaciones, estamos interesados en los espacios de las columnas de una matriz A . Esto es, son los espacios sucesivos generados por las columnas a_1, a_2, \dots de A :

$$\langle a_1 \rangle \subseteq \langle a_1, a_2 \rangle \subseteq \langle a_1, a_2, a_3 \rangle \subseteq \dots$$

Siendo $\langle a_1 \rangle$ es el espacio unidimensional generado por a_1 , $\langle a_1, a_2 \rangle$ el espacio genéricamente bidimensional generado por a_1 y a_2 , y así sucesivamente.

La idea de la factorización QR es la construcción de una secuencia de vectores ortonormales q_1, q_2, \dots que generan estos espacios sucesivos. Supongamos de momento que $A \in \mathbb{C}^{m \times n}$ ($m \geq n$) tiene rango completo n .

Queremos que la secuencia q_1, q_2, \dots tenga la propiedad: $\langle q_1, q_2, \dots, q_j \rangle = \langle a_1, a_2, \dots, a_j \rangle$ con $j = 1, \dots, n$. Esto es, (1)

$$a_1 = r_{11}q_1$$

$$a_2 = r_{12}q_1 + r_{22}q_2$$

...

$$a_n = r_{1n}q_1 + r_{2n}q_2 + \dots + r_{nn}q_n$$

Observándolo en forma de matriz, tenemos:

$$A = \hat{Q} \cdot \hat{R}$$

Donde \hat{Q} es $m \times n$ con columnas ortonormales y \hat{R} es $n \times n$ y triangular superior.

Esta factorización se denomina, factorización QR reducida de A .

B) FACTORIZACIÓN QR COMPLETA

Una factorización QR completa de $A \in \mathbb{C}^{m \times n}$ ($m \geq n$), añade $m - n$ columnas ortonormales adicionales a \hat{Q} , convirtiéndola en una matriz unitaria $m \times m$ Q .

Se añaden filas de ceros a \hat{R} de forma que se convierte en una matriz $m \times n$ R , todavía triangular superior.

Todas las matrices tienen factorizaciones QR, y bajo restricciones apropiadas (asumiendo que la diagonal de R tiene entradas >0 , por ejemplo), son únicas.

- COMPARACIÓN ENTRE FACTORIZACIONES

En la factorización QR completa, Q es $m \times n$, R es $m \times n$ y las últimas $m - n$ columnas de Q están multiplicadas por ceros en R , con lo que podemos pensar en ellas como y filas “silenciadas”.

En la factorización QR reducida, las columnas y fila silenciadas son eliminadas. Ahora \hat{Q} es $m \times n$, \hat{R} es $n \times n$ y ninguna de las filas de \hat{R} son necesariamente cero.

En la factorización QR completa, las columnas q_j para $j > n$ son ortogonales a $Im(A)$. Suponiendo que A es de rango completo n , constituyen una base ortonormal de $Im(A)^\perp$, o equivalentemente, de $Ker(A^*)$.

ORTOGONALIZACIÓN DE GRAM-SCHMIDT

Las ecuaciones (1) sugieren un método para calcular la factorización QR reducida.

Dados a_1, a_2, \dots podemos construir los vectores q_1, q_2, \dots y las componentes r_{ij} mediante un proceso de ortogonalización sucesiva. Esta es una idea antigua, conocida como ortogonalización de Gram-Schmidt.

El proceso funciona de la siguiente forma: en el paso j -ésimo, deseamos encontrar un vector unitario $q_j \in \langle a_1, \dots, a_j \rangle$ que sea ortogonal a q_1, \dots, q_{j-1} . El vector,

$$v_j = a_j - (q_1^* a_j)q_1 - \dots - (q_{j-1}^* a_j)q_{j-1} \quad (2)$$

Es del tipo buscado, salvo que todavía no esté normalizado. Si dividimos por $\|v_j\|_2$, el resultado es un vector apropiado q_j . Así, reescribimos (1) de la siguiente forma (3):

$$\begin{aligned} q_1 &= \frac{a_1}{r_{11}} \\ q_2 &= \frac{a_2 - r_{12} \cdot q_1}{r_{22}} \\ &\dots \\ q_n &= \frac{a_n - \sum_{i=1}^{n-1} r_{in} \cdot q_i}{r_{nn}} \end{aligned}$$

De (2) se desprende que una definición apropiada para los coeficientes r_{ij} en los numeradores de (3) es:

$$r_{ij} = q_i^* a_j \quad (i \neq j)$$

Los coeficientes r_{jj} en los denominadores se escogen para normalizar:

$$|r_{jj}| = \left\| a_j - \sum_{i=1}^{j-1} r_{ij} \cdot q_i \right\|_2$$

Si escogemos $r_{jj} \geq 0$, \hat{R} tendrá componentes no negativas en la diagonal.

ALGORITMO CLÁSICO DE GRAM-SCHMIDT

for $j = 1$ *to* n

$$v_j = a_j$$

for $i = 1$ *to* $j - 1$

$$r_{ij} = q_i^* \cdot a_j$$

$$v_j = v_j - r_{ij}q_i$$

end for

$$r_{jj} = \|v_j\|_2$$

$$q_j = \frac{v_j}{r_{jj}}$$

end for

Este algoritmo es correcto matemáticamente, pero es inestable numéricamente. Esta inestabilidad, la corregiremos mediante el algoritmo de Gram-Schmidt modificado.

GRAM-SCHMIDT MODIFICADO

Si $Q \in \mathbb{C}^{m \times n}$ con $m \geq n$, tiene columnas ortonormales la proyección ortogonal QQ^* es suma de n proyecciones ortogonales de rango 1. Recordemos que en el método de Gram-Schmidt cada q_j se obtiene proyectando a_j sobre $\langle q_1, \dots, q_{j-1} \rangle^\perp$, y después normalizando. Es decir,

$$q_j = \frac{(I_m - Q_{j-1}Q_{j-1}^*) \cdot a_j}{\|(I_m - Q_{j-1}Q_{j-1}^*) \cdot a_j\|_2} \text{ donde } Q_i = (q_1, \dots, q_i), \text{ de dimensión } m \times i$$

Así, es fácil ver que:

$$I_m - Q_{j-1}Q_{j-1}^* = I_m - \sum_{i=1}^{j-1} q_i q_i^*$$

Pero si hacemos el producto,

$$(I_m - q_{j-1}q_{j-1}^*) \cdot (I_m - q_{j-2}q_{j-2}^*) \cdot \dots \cdot (I_m - q_1q_1^*)$$

Y tenemos en cuenta que $q_i^* \cdot q_j = 0$ si $i \neq j$, resulta que este producto es

$$I_m - q_{j-1}q_{j-1}^* - \dots - q_1q_1^* = I - Q_{j-1}Q_{j-1}^*$$

Luego, si tomamos $P_i = I_m - q_iq_i^*$, entonces

$$I_m - Q_{j-1}Q_{j-1}^* = P_{j-1} \cdot P_{j-2}^* \cdot \dots \cdot P_1 \quad (4)$$

La identidad (4) nos dice que es lo mismo hacer este proceso de proyecciones sucesivas que hacer directamente la proyección sobre $\langle q_1, \dots, q_{j-1} \rangle^\perp$. Esto último es lo que hace el método clásico de Gram-Schmidt. Pero el resultado teórico es el mismo si se hacen las proyecciones sucesivas de rango n-1. La implementación de este procedimiento es lo que se denomina algoritmo modificado de Gram-Schmidt.

ALGORITMO MODIFICADO DE GRAM-SCHMIDT

for $j = 1$ *to* n

$$v_i = a_i$$

for $i = 1$ *to* n

$$r_{ii} = \|v_i\|$$

$$q_i = \frac{v_i}{r_{ii}}$$

for $j = i + 1$ *to* n

$$r_{ij} = q_i^* \cdot v_j$$

$$v_j = v_j - r_{ij} \cdot q_i$$

end for

end for

end for

Teorema

Los algoritmos clásico y modificado de Gram-Schmidt requieren $\sim 2 \cdot m \cdot n^2$ flops para calcular la factorización QR de una matriz $m \times n$.

Demostración:

Cuando m y n son grandes, el mayor número de operaciones se encuentran en las dos siguientes sentencias del algoritmo:

$$r_{ij} = q_i^* \cdot q_j \text{ (Método clásico } r_{ij} = q_i^* \cdot a_j \text{)}$$

$$q_j = q_j - r_{ij}q_i$$

La primera línea calcula el producto escalar $q_i^* \cdot q_j$ y requiere m multiplicaciones y $m-1$ sumas. Mientras que la segunda línea calcula $q_j - r_{ij}q_i$, precisando m multiplicaciones y n restas. Así pues, el trabajo que se hace en cada iteración es del orden de $\sim 4 \cdot m$. Pero como hay que hacer $i = 1:j-1$ iteraciones para $j = 1:n$, el número total de flops necesarios es asintóticamente:

$$\begin{aligned} \sum_{j=1}^n \sum_{i=1}^{j-1} 4 \cdot m &= 4 \\ &\cdot m \sum_{j=1}^n (j-1) \\ &= 4 \cdot m \sum_{j=1}^{n-1} j = 4 \cdot m \cdot \frac{(n-1) \cdot n}{2} = 2 \cdot m \cdot n^2 + O(m \cdot n) \end{aligned}$$

▪

5. MÉTODO DE LA POTENCIA

Sea $A \in \mathbb{R}^{n \times n}$ una matriz diagonalizable y supongamos que sus valores propios cumplen la siguiente propiedad:

$$|\lambda_1| > |\lambda_2| \geq \dots \geq |\lambda_n|$$

En cuyo caso diremos que λ_1 es el autovalor dominante de la matriz.

Sea $B = \{x_1, \dots, x_n\}$ una base de \mathbb{R}^n formada por vectores propios asociados a $\lambda_1, \dots, \lambda_n$ respectivamente. Se cumple entonces,

$$A^2 x_i = A(Ax_i) = A(\lambda_i x_i) = \lambda_i Ax_i = \lambda_i (\lambda_i x_i) = \lambda_i^2 x_i$$

Por inducción es fácil probar que para todo $k \geq 0$ se tiene:

$$A^k x_i = \lambda_i^k x_i, i = 1, \dots, n$$

Dado un vector $z_0 \in \mathbb{R}^n$ se define, a partir de él, la sucesión (z_n) con

$$z_n = Az_{n-1} = A^2 z_{n-2} = \dots = A^n z_0$$

Si las coordenadas del vector z_0 respecto de la base B son $(\alpha_1, \dots, \alpha_n)$ se tiene que $z_0 = \alpha_1 x_1 + \dots + \alpha_n x_n$, por lo que

$$\begin{aligned} z_k = A^k z_0 &= A^k (\alpha_1 x_1 + \dots + \alpha_n x_n) = \alpha_1 A^k x_1 + \dots + \alpha_n A^k x_n = \\ &= \lambda_1^k \alpha_1 x_1 + \dots + \lambda_n^k \alpha_n x_n = \lambda_1^k \left[\alpha_1 x_1 + \sum_{i=2}^n \left(\frac{\lambda_i}{\lambda_1}\right)^k \alpha_i x_i \right] \end{aligned}$$

Dado que $|\lambda_1| > |\lambda_i|$ se tiene que $\lim_{k \rightarrow \infty} \left(\frac{\lambda_i}{\lambda_1}\right)^k = 0, i = 2, 3, \dots, n$

Se cumple entonces que $\lim_{k \rightarrow \infty} \frac{z_k}{\lambda_1^k} = \alpha_1 x_1$ es un autovalor de A asociado al autovalor λ_1 .

Si k es suficientemente grande, tenemos:

$$Az_k \approx z_{k+1} \approx \lambda_1^{k+1} \alpha_1 x_1 = \lambda_1 (\lambda_1^k \alpha_1 x_1) = \lambda_1 z_k$$

Así, la sucesión (z_n) nos proporciona un método para aproximar el autovalor λ_1 .

El método no convergerá a un vector concreto ya que

$$z_{k+1} \approx \lambda_1 z_k \neq z_k$$

Dado que z_{k+1} y z_k tienen la misma dirección, ambos representan el vector propio buscado.

ESCALADO:

Si $|\lambda_1| > 1$ la sucesión (z_n) converge a una dirección determinada, dada por el vector propio, pero las normas de los vectores van creciendo llegando a divergir a un vector de coordenadas infinitas. Si por el contrario $|\lambda_1| < 1$, la sucesión (z_n) converge al vector nulo. Es decir, en ninguno de los dos casos podremos calcular un el valor propio ni el vector propio directamente.

Una forma de acelerar el proceso es: una vez calculado el valor w (ver el algoritmo siguiente), el vector z_n es escalado, y el vector propio se aproxima mediante el llamado Cociente de Rayleigh.

ALGORITMO MÉTODO DE LA POTENCIA SIMPLE

$v^{(0)}$ vector inicial tal que $\|v^{(0)}\| = 1$

Para $k = 1, 2, \dots$

$$w = A \cdot v^{(k-1)}$$

$$v^{(k)} = \frac{w}{\|w\|}$$

$$\lambda^{(k)} = (v^{(k)})^T \cdot A \cdot v^{(k)}$$

6. MÉTODO DE LA POTENCIA INVERSA

El método de la potencia inversa es el método de la potencia pero aplicado a la matriz $(\mu \cdot I_n - A)^{-1}$ para algún número μ , que es denominado desplazamiento y es usado para aumentar la velocidad de convergencia del método de la potencia. No hay que olvidar, que los valores propios de esta matriz son los mismos que los de la matriz A desplazados por μ . Además este método nos permite obtener un valor propio que no sea necesariamente el valor propio dominante siempre y cuando tengamos algún conocimiento de ellos.

ALGORITMO DE LA POTENCIA INVERSA

$v_0 = \text{vector aleatorio tal que } \|v_0\| = 1$

Para $k = 1, 2, 3, \dots$

Resolver $(A - \mu \cdot I) \cdot w = v^{(k-1)}$

$$v^{(k)} = \frac{w}{\|w\|}$$

$$\lambda^{(k)} = (v^{(k)})^T \cdot A \cdot v^{(k)} \quad (\text{Cociente de Rayleigh})$$

Todas las propiedades y todo lo mencionado para el método de la Potencia es válido para este método de la iteración inversa.

7. MÉTODO DEL COCIENTE DE RAYLEIGH

El método del cociente de Raleigh es exactamente igual que el de la iteración inversa pero con un desplazamiento que varía en cada iteración. La cuestión es, ¿Cómo elegir el desplazamiento para hacer que el residuo sea lo más pequeño posible en cada iteración?

Residuo: $r = \|\lambda \cdot x - A \cdot x\|$, donde λ y x son el valor y vector unitario propios aproximados que se calculan en cada iteración. Como λ es un valor propio aproximado, que varía en cada iteración, bien podemos hacer uso de él en cada iteración.

Queremos hallar el valor de λ para el que se hace mínimo el residuo. Esto es, queremos hallar,

$$\min_{\alpha \in \mathbb{C}} \|\alpha \cdot x - A \cdot x\|_2$$

En nuestro caso, el sistema de ecuaciones normales es

$$x^* \cdot x \cdot \alpha_0 = x^* \cdot A \cdot x$$

Por lo tanto, $\alpha_0 = \frac{x^* \cdot A \cdot x}{x^* \cdot x}$ es el número que minimiza el residuo.

Definición:

Para $A \in F^{n \times n}$ se llama cociente de Rayleigh de A en $x \in \mathbb{C}^n$ al número:

$$R(x) = \frac{x^* \cdot A \cdot x}{x^* \cdot x}$$

Así, lo dicho anteriormente nos demuestra el siguiente teorema

Teorema:

Si $A \in \mathbb{C}^{n \times n}$ y $x \in \mathbb{C}^n$ entonces el cociente de Rayleigh de A en x es el número donde alcanza el mínimo residuo $\|\alpha \cdot x - A \cdot x\|_2$. Es decir

$$\min_{\alpha \in \mathbb{C}} \|\alpha \cdot x - A \cdot x\|_2 = \|\alpha_0 \cdot x - A \cdot x\|_2$$

En la práctica: este algoritmo consiste en sustituir μ por λ en el algoritmo de la potencia inversa iniciando el primer desplazamiento con el cociente de Rayleigh de la matriz dada en el vector inicial elegido.

De esta forma, resulta que $\lambda = R(x)$ en cada iteración

Teorema:

Si para una matriz simétrica A y un vector q_0 el algoritmo del cociente de Rayleigh (ver abajo) converge, lo hace a partir de cierto momento, cuadráticamente.

Demostración:

Si el algoritmo del cociente de Rayleigh converge, lo hace a un valor-vector propio unitario (λ_0, x_0) de A. Esto es, el algoritmo genera una secuencia (λ_k, x_k, q_k) de valores-vectores propios aproximados de A que converge a (λ_0, x_0) . Pero $\lambda_k = R(\alpha_k q_k)$ y desarrollando en serie de Taylor $R(x)$ alrededor de x_0 tenemos que:

$$R(x) - R(x_0) = \nabla R(x_0)(x - x_0) + O(\|x - x_0\|^2) = O(\|x - x_0\|^2)$$

Porque x_0 es un vector propio de A. Esto conlleva a que para k suficientemente grande, $R(\alpha_k q_k) - R(x_0) = O(\|q_k - x_0\|^2)$.

Como $q_k \rightarrow x_0$ converge linealmente (al menos), el residuo converge a cero, a partir de ese k al menos cuadráticamente.

▪

ALGORITMO MÉTODO DEL COCIENTE DE RAYLEIGH

$v^{(0)}$ = un vector cualquiera tal que $\|v^{(0)}\| = 1$

$\lambda^{(0)} = (v^{(0)})^T \cdot A \cdot v^{(0)}$ = cociente de Rayleigh

Para $k = 1, 2, \dots$

Resolver $(A - \lambda^{(k-1)} \cdot I) \cdot w = v^{(k-1)}$

$$v^{(k)} = \frac{w}{\|w\|}$$

$\lambda^{(k)} = (v^{(k)})^T \cdot A \cdot v^{(k)}$

8. MÉTODO DE RESIDUOS MÍNIMOS GENERALIZADO (GMRES)

El método Residuos Mínimos Generalizado (GMRES) se utiliza para sistemas no simétricos y su finalidad es resolver $A \cdot x = b$, siendo A no singular. El objetivo de GMRES es minimizar la norma 2 del residuo sobre K_k . Es decir, la iteración de GMRES está caracterizada por

$$x_k \in K_k \ ; \ \|b - A \cdot x_k\|_2 = \min_{z \in K_k} \|b - A \cdot z\|_2$$

Así, GMRES resuelve un problema de mínimos cuadrados sobre el espacio K_k . El análisis que realizamos sobre este método se lleva a cabo suponiendo que no ha sido dada solución inicial, o que ésta es el vector nulo.

GMRES utiliza una base ortonormal que se forma aplicando directamente el proceso de ortonormalización de Gram-Schmidt modificado a la sucesión $\{b, A \cdot b, A^2 \cdot b, \dots\}$. Ese algoritmo es conocido con el nombre de Arnoldi y los vectores (q_j) que obtienen se denominan vectores de Arnoldi y forman una base ortonormal para K_k . Además este algoritmo es usado para calcular los valores propios de una matriz.

ALGORITMO DE ARNOLDI

b arbitrario

$$q_1 = \frac{b}{\|b\|}$$

Para n = 1,2,3, ...

$$v = A \cdot q_n$$

Para j = 1 hasta n

$$h_{jn} = q_j^* \cdot v$$

$$v = v - h_{jn} \cdot q_j$$

fin para

$$h_{n+1,n} = \|v\|$$

$$q_{n+1} = \frac{v}{h_{n+1,n}}$$

fin para

Es importante que para calcular v_{k+1} es necesario que

$$A \cdot v_k - \sum_{j=1}^k \langle v_j, A \cdot v_k \rangle \cdot v_j \neq 0$$

Si esto no ocurre entonces $A \cdot v_k \in K_k = \text{gen}(v_1, \dots, v_k) = \text{gen}(b, A \cdot b, A^2 \cdot b, \dots, A^{k-1} \cdot b)$, $A^k \cdot b \in K_k$, y por tanto $K_{k+1} \subset K_k$. Por tanto, el algoritmo de Arnoldi nos muestra la información necesaria para obtener dicha base. Además, en tal paso k se puede encontrar la solución exacta del sistema.

Teorema:

Sean A una matriz real $n \times n$ singular y $b \in \mathbb{R}^n$. Supongamos que hemos calculado los vectores de Arnoldi v_1, \dots, v_k y que $A \cdot v_k = \sum_{j=1}^k \langle v_j, A \cdot v_k \rangle \cdot v_j$, entonces $x = A^{-1} \cdot b \in K_k$.

Demostración:

Puesto que $V_k = [v_1, \dots, v_k]$ es una matriz ortogonal $n \times k$, entonces $H = V_k^T \cdot A \cdot V_k$ es una matriz $k \times k$ no singular. Tomemos $\beta = \|b\|_2$ y sea $y \in \mathbb{R}^k$ cualquiera. Así, $V_k \cdot y \in K_k$ y $\|b - A \cdot V_k \cdot y\|_2 = \|\beta \cdot v_1 - V_k \cdot H \cdot y\|_2 = \|V_k \cdot (\beta \cdot e_1 - H \cdot y)\|_2$

Donde $e_1 \in \mathbb{R}^k$. Así, tomando $y = \beta \cdot H^{-1} \cdot e_1$ se obtiene $b - A \cdot V_k \cdot y = 0$ y así $x = A^{-1} \cdot b = V_k \cdot y$.

▪

Es importante que veamos que este teorema muestra que el algoritmo de Arnoldi calcula v_{k+1} mientras que $x = A^{-1} \cdot b$ no pertenece a K_k .

Teorema (Existencia y unicidad de cada elemento en la definición de GMRES):

Supongamos que $r_j \neq 0$ para $0 \leq j \leq k-1$. Existe un único $x_k \in K_k$ que satisface

$$x_k \in K_k ; \quad \|b - A \cdot x_k\|_2 = \min_{z \in K_k} \|b - A \cdot z\|_2$$

Demostración:

Cómo $\min_{y \in K_{k-1}} \|b - A \cdot y\|_2 = r_{k-1} \neq 0$, entonces x no pertenece a K_{k-1} . Sean $V_k = [v_1, \dots, v_k]$, donde $\{v_1, \dots, v_k\}$ es la base de Arnoldi para K_k . Tomando $S = \text{Im}(A \cdot V_k)$, $y_0 = b$ y $\|\cdot\| = \|\cdot\|_2$. Además sabemos que existe un único $y_k \in \mathbb{R}^k$ tal que

$$\|b - A \cdot V_k \cdot y_k\|_2 = \min_{z \in \text{Im}(A \cdot V_k)} \|b - z\|_2 = \min_{y \in \mathbb{R}^k} \|b - A \cdot V_k \cdot y_k\|_2$$

Así, $x_k = V_k \cdot y_k$ es el único elemento de K_k que satisface; $\|b - A \cdot x_k\|_2 = \min_{z \in K_k} \|r_0 - A \cdot z\|_2$.

▪

*IMPLEMENTACIÓN

Notemos que en el algoritmo de Arnoldi se construye una matriz Hessenberg Superior H_k de orden $(k + 1) \times k$ con la propiedad $A \cdot v_k = h_{1k} \cdot v_1 + \dots + h_{kk} \cdot v_k + h_{k+1,k} \cdot v_{k+1}$.

Es decir, H_k esta caracterizada por la identidad $A \cdot V_k = V_{k+1} \cdot H_k$. Además, como V_{k+1} es ortogonal, es isometría y tal que $V_{k+1} \cdot e_1 = \frac{b}{\beta}$, siendo $\beta = \|b\|_2$, así para $y \in \mathbb{R}^k$ tenemos $\|b - A \cdot V_k \cdot y_k\|_2 = \|V_{k+1} \cdot (\beta \cdot e_1 - H_k \cdot y)\|_2 = \|\beta \cdot e_1 - H_k \cdot y\|_2$, con $e_1 \in \mathbb{R}^{k+1}$. Así,

$$\begin{aligned} \|b - A \cdot x_k\|_2 &= \min_{x \in K_k} \|b - A \cdot x\|_2 \\ &= \min_{y \in \mathbb{R}^k} \|b - A \cdot V_k \cdot y\|_2 = \min_{y \in \mathbb{R}^k} \|\beta \cdot e_1 - H_k \cdot y\|_2 \\ &= \|\beta \cdot e_1 - H_k \cdot y_k\|_2 \end{aligned}$$

Así reducimos el problema de minimización original a uno cuya estructura es Hessenberg, que facilita el proceso de cálculo de y_k . De esta forma podemos plantear el primer algoritmo de GMRES con la simplificación mencionada.

ALGORITMO GMRES

$$q_1 = \frac{b}{\|b\|}$$

for $n = 1, 2, 3, \dots$

< Paso n de la iteración de Arnoldi >

{Resolver para y_k , por mínimos cuadrados, $H_n \cdot y_n = \text{norm}(b) \cdot e_1$ }

$$x_n = Q_n \cdot y_n$$

end

10. ANÁLISIS DE RESULTADOS

En esta sección vamos a mostrar los resultados junto con las conclusiones obtenidas tras realizar el cálculo de valores propios, así como la resolución de sistemas lineales mencionados con anterioridad, mediante distintos algoritmos.

Para realizar el análisis, vamos a distribuir los resultados en cuatro tablas, dos para matrices reales y dos para complejas y a su vez, dos para caso simétrico (hermítico en caso complejo) o no simétrico.

En cuanto al análisis nos centraremos en el tiempo necesario por cada método para obtener el resultado y la precisión con la que realiza el cálculo. Tras la obtención de los datos, podremos dar una conclusión coherente.

Este estudio consiste en ejecutar a través de Matlab los métodos que hemos mencionado con anterioridad. Para ello, vamos a generar matrices de distinto tamaño aleatorias ($n=100$, $n=500$, $n=1000$) para ver la evolución que siguen los métodos tanto en tiempo como en precisión y poder sacar conclusiones.

Para realizar el siguiente análisis, las matrices son generadas mediante el comando `randn(n)` para el caso real (`randn(n)+i*randn(n)` para el caso complejo), con lo que cada entrada sigue una distribución de centro 0 y varianza 1 (en el caso complejo, las partes real e imaginaria siguen dicha distribución). No solamente realizamos el estudio para matrices reales o complejas, sino que también para matrices reales simétricas, usando $(A+A')/2$, de manera análoga para matrices hermíticas. Respecto al estudio de la precisión, dependiendo del objetivo de cada método hemos usado una fórmula u otra. Para los algoritmos encargados del cálculo de valores propios, hemos usado $\min(\text{svd}(\lambda \cdot \text{eye}(n)-A))$, para GMRES, encargado de resolver sistemas lineales $\text{norm}(A \cdot x - b, \text{inf})$, y para Arnoldi a través de la fórmula $\text{norm}(A \cdot Q - Q \cdot H)$ y Lanczos viendo si la matriz obtenida es tridiagonal. Todas estas fórmulas han sido escogidas entre las distintas opciones de medida de precisión que mostraba el libro de la bibliografía.

Comenzamos el análisis con los métodos implementados para el cálculo de valores propios.

10.1 ANÁLISIS DE RESULTADOS DEL MÉTODO DE LA POTENCIA

En las siguientes tablas se muestran los resultados obtenidos al ejecutar el método de la potencia:

CASO REAL		N=100	N=500	N=1000
Método de la Potencia	Tiempo medio	3.042609	23.084848	228.126076
	Precisión	0.0219	0.7304	0.0880

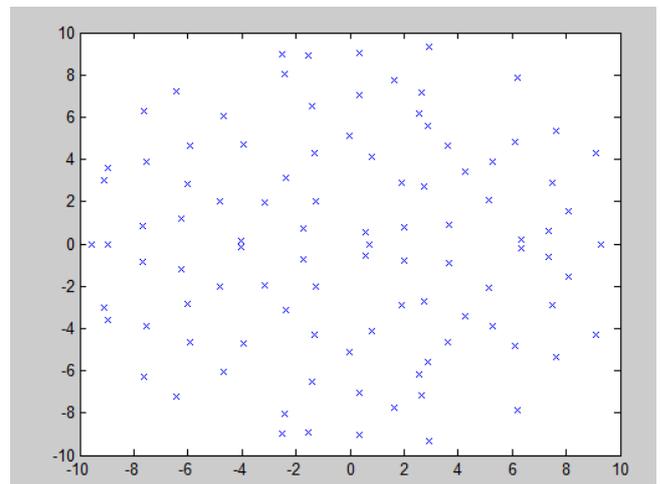
CASO REAL SIMÉTRICO		N=100	N=500	N=1000
Método de la Potencia	Tiempo medio	0.344925	23.308711	55.7710541
	Precisión	4.0055e-13	4.7785e-13	2.3898e-13

CASO COMPLEJO		N=100	N=500	N=1000
Método de la Potencia	Tiempo medio	0.368135	17.971464	136.995087
	Precisión	3.0195e-13	3.1612e-15	2.3906e-15

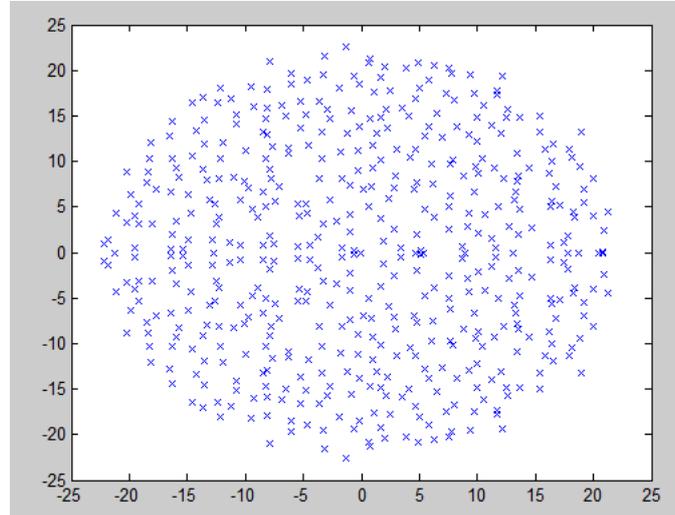
CASO COMPLEJO HERMÍTICO		N=100	N=500	N=1000
Método de la Potencia	Tiempo medio	1.243732	13.691878	108.866787
	Precisión	1.0813e-14	2.4632e-15	3.6814e-15

A continuación realizamos un análisis gráfico de la situación en el caso de la potencia real, puesto que el método no nos da una buena precisión al contrario de los demás casos. Este resultado, es consecuencia de cómo aparece en las siguientes gráficas que hay varios valores propios parecidos al valor propio de módulo máximo, como veremos en las siguientes gráficas:

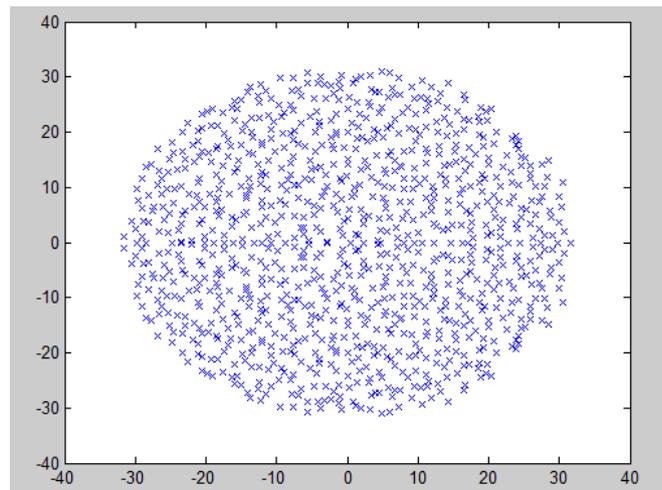
En esta primera imagen se muestran los valores propios para el caso real con $n=100$.



En esta segunda imagen se muestran los valores propios para el caso real con $n=500$.



En esta tercera imagen se muestran los valores propios para el caso $n=1000$.



10.2 ANALISIS DE RESULTADOS DEL MÉTODO DE LA POTENCIA INVERSA

En las siguientes tablas se muestran los resultados obtenidos al ejecutar el método de la potencia inversa:

CASO REAL		N=100	N=500	N=1000
Método de la Potencia Inversa	Tiempo medio	0.276677	10.428855	70.592862
	Precisión	2.7992e-16	11265e-17	0.0040

--	--	--	--	--

CASO REAL SIMETRICO		N=100	N=500	N=1000
Método de la Potencia Inversa	Tiempo medio	0.288409	10.424358	69.979606
	Precisión	2.2255e-15	1.3161e-15	6.1288e-15

CASO COMPLEJO		N=100	N=500	N=1000
Método de la Iteración Inversa	Tiempo medio	0.5592	17.265467	116.453995
	Precisión	1.6997e-10	6.4305e-15	2.4122e-16

CASO COMPLEJO HERMÍTICO		N=100	N=500	N=1000
Método de la Potencia Inversa	Tiempo medio	0.346496	17.004654	116.298042
	Precisión	2.2934e-13	5.7548e-16	3.2518e-15

Tras la ejecución de este método, podemos decir que en todos los casos se obtiene una buena precisión y un buen tiempo de resolución, sin observarse ninguna anomalía en ningún caso concreto.

10.3 ANALISIS DE RESULTADOS DEL COCIENTE DE LA RAYLEIGH

En las siguientes tablas se muestran los resultados obtenidos al ejecutar el método del Cociente de Rayleigh:

CASO REAL		N=100	N=500	N=1000
Método del	Tiempo medio	0.328852	8.9815	50.578418

Cociente de Rayleigh	Precisión	3.7382e-17	1.3824e-15	2.3796e-16
----------------------	-----------	------------	------------	------------

CASO REAL SIMETRICO		N=100	N=500	N=1000
Método del Cociente de Rayleigh	Tiempo medio	0.393121	8.610496	50.468184
	Precisión	4.5250e-17	2.1523e-16	2.5491e-15

CASO COMPLEJO		N=100	N=500	N=1000
Método del Cociente de Rayleigh	Tiempo medio	0.482999	14.43989	El programa no puede manejar un problema de este tamaño (problemas en el condicionamiento de la matriz)
	Precisión	1.0015e-14	4.4021e-15	

CASO COMPLEJO HERMÍTICO		N=100	N=500	N=1000
Método del Cociente de Rayleigh	Tiempo medio	0.398345	14.020565	92.223521
	Precisión	1.3453e-15	7.7376e-16	1.7525e-15

El estudio realizado de este algoritmo es en general bueno, tanto a nivel de precisión como de rapidez. Sin embargo, dentro del caso complejo general, para matrices de gran tamaño (por ejemplo n=1000) nos encontramos con que nuestro algoritmo no es capaz de resolver por problemas en el condicionamiento de la matriz.

10.4 CONCLUSIONES DE LOS RESULTADOS EXPERIMENTADOS DEL CÁLCULO DE VALORES PROPIOS

Tras la ejecución de estos tres métodos, en los cuales el bloque destinado en las tablas a tiempo medio corresponde con el tiempo total utilizado a la hora de

resolver el método correspondiente y la transformación previa de la matriz a forma Hessenberg, podemos concluir que en general, en los tres métodos implementados es el los casos reales simétricos donde los métodos son más rápidos como cabía esperar. En cuanto a la precisión se observa que hay una gran similitud en todos los casos a excepción del caso de la potencia en el que podemos decir que la precisión no es muy buena. En conclusión, para la obtención de valores propios, teniendo en cuenta tanto la precisión como el tiempo que tarda en obtener el resultado, podemos afirmar que el método del Cociente de Rayleigh es el más rápido y más preciso para el caso simétrico.

10.5 ANÁLISIS DE RESULTADOS DEL MÉTODO DE GRAM-SCHMIDT Y GRAM-SCHMIDT MODIFICADO

A continuación pasamos a hacer un estudio en el que mostraremos las diferencias, si es que existen, entre los métodos Gram-Schmidt y Gram-Schmidt modificado (gsinestable.m y gstable.m) que mencionamos anteriormente. Estos algoritmos nos muestran la factorización reducida y completa QR respectivamente de una matriz A que generamos aleatoriamente, usando las mismas dimensiones que en el cuadro comparativo anterior.

En las siguientes tablas se muestran los resultados obtenidos al ejecutar el método de la Gram-Schmidt modificado:

CASO REAL		N=100	N=500	N=1000
Método de Gram-Schmidt modificado	Tiempo medio	0.015782	0.430447	2.751243
	Precisión	2.7979e-14	2.6213e-13	7.0163e-13

CASO REAL SIMETRICO		N=100	N=500	N=1000
Método de Gram-Schmidt modificado	Tiempo medio	0.020574	0.429577	2.846488
	Precisión	1.9842e-14	1.8591e-13	4.8826e-13

CASO COMPLEJO		N=100	N=500	N=1000
Método de Gram-Schmidt modificado	Tiempo medio	0.037809	1.063653	7.235391
	Precisión	4.3755e-14	4.3274e-13	1.01071e-12

CASO COMPLEJO HERMÍTICO		N=100	N=500	N=1000
Método de Gram-Schmidt modificado	Tiempo medio	0.041442	0.996913	7.094827
	Precisión	3.1270e-14	3.0649e-13	7.9518e-13

9.5 CONCLUSIONES DE LOS RESULTADOS DEL COCIENTE DEL MÉTODO DE GRAM-SCHMIDT Y GRAM-SCHMIDT MODIFICADO

Las tablas anteriores solo contienen los resultados asociados al método G-S modificado puesto que el método G-S presenta problemas a la otra de resolver debido a problemas de tipo NaN. Por este motivo, existe y se observa claramente la necesidad de mejora del método G-S por G-S modificado, donde hay una buena precisión y es relativamente rápido el método.

9.6 ANALISIS DE RESULTADOS DEL MÉTODO DE ARNOLDI Y LANCZOS

En el siguiente estudio vamos a comparar al método de Arnoldi y el método de Lanczos, siendo la diferencia en implementar uno método u otro.

En las siguientes tablas se muestran los resultados obtenidos al ejecutar el método de Arnoldi:

CASO REAL		N=100	N=500	N=1000
Método de Arnoldi	Tiempo medio	0.007141	0.017816	0.055933
	Precisión	1.0450e-15	2.0904e-15	2.9459e-15

CASO REAL SIMÉTRICA		N=100	N=500	N=1000
Método de Arnoldi	Tiempo medio	0.025486	0.6297268	3.962506
	Precisión	1.4304e-15	4.2232e-15	6.3615e-15

CASO COMPLEJA		N=100	N=500	N=1000
Método de Arnoldi	Tiempo medio	0.041827	1.545405	11.168266
	Precisión	3.6267e-15	8.2236e-15	1.1809e-14

CASO COMPLEJA HERMÍTICA		N=100	N=500	N=1000
Método de Arnoldi	Tiempo medio	0.042178	0.146738	0.419261
	Precisión	2.6211e-15	4.2168e-15	5.3907e-15

En las siguientes tablas se muestran los resultados obtenidos al ejecutar el método de Lanczos:

CASO REAL SIMÉTRICA		N=100	N=500	N=1000
Método de Lanczos	Tiempo medio	0.000339	0.001081	0.008260
	Precisión	2.6007e-13	4.1858e-13	5.2104e-13

CASO COMPLEJA HERMÍTICA		N=100	N=500	N=1000
Método de Lanczos	Tiempo medio	0.003672	0.002523	0.008903
	Precisión	2.5763e-12	3.4929e-12	9.9077e-12

10.7 CONCLUSIONES DE LA COMPARACION DE ARNOLDI Y LANCZOS

En este estudio, cabe comentar que para realizar el cálculo de la precisión, hemos implementado dos veces el algoritmo de Arnoldi, pero en la contabilización del tiempo que tarda en obtener la solución, solamente hemos incluido lo que tarda en realizar una vez la llamada de Arnoldi, que es lo que nos interesa.

Respecto al algoritmo de Lanczos la forma de computar la precisión se ha realizado mirando si la matriz obtenida es tridiagonal.

Como ya sabíamos, el método de Lanczos es una adaptación del método de Arnoldi, por ello observamos una mayor precisión en éste frente al tiempo computado por el método de Arnoldi.

10.8 ANALISIS DE RESULTADOS DEL METODO DE RESIDUOS MINIMOS GENERALIZADO

A continuación, procedemos a mostrar los resultados obtenidos al implementar el método de Residuos Mínimos Generalizado (GMRES):

CASO REAL		N=100	N=500	N=1000
Método GMRES	Tiempo medio	0.015878	0.031177	0.074044
	Precisión	2.8866e-15	1.0658e-14	1.1102e-14

CASO COMPLEJO		N=100	N=500	N=1000
Método GMRES	Tiempo medio	0.029485	0.085898	0.215092
	Precisión	2.8581e-15	1.0500e-14	1.2229e-14

CASO COMPLEJO HERMITICO		N=100	N=500	N=1000
Método GMRES	Tiempo medio	0.032763	0.081890	0.217440
	Precisión	4.8672e-12	3.5857e-05	8.0071e-04

10.9 CONCLUSIONES DEL MÉTODO GMRES

Recordemos que dijimos que GMRES se usa para matrices no simétricas. En efecto, para simétricas no observamos mejoras, pero para ese caso hay otro algoritmo (Gradiente Conjugado) que es mucho más rápido aunque no lo hemos incluido en este trabajo.

11. NOTA FINAL

Dado que hemos ido analizando las conclusiones en distintas secciones, hacemos simplemente algunos comentarios finales a modo de resumen general del trabajo:

En el Álgebra Lineal Numérica hay muchos métodos, usados sobre todo para matrices de gran tamaño, y la elección entre uno y otro método es un factor clave para asegurar la velocidad de convergencia y precisión.

Utilizando estos métodos en las implementaciones realizadas íntegramente en este trabajo, concluimos que incluso mediante una programación no profesional podemos calcular valores propios y resolver sistemas de ecuaciones para tamaños de matrices de 1000 o más filas y columnas.

Finalmente, hemos visto como en ocasiones el uso de un sencillo resultado mejora considerablemente la solución de problemas numéricos, el caso de Gram-Schmidt modificado, por ejemplo.

12. BIBLIOGRAFÍA

Para la elaboración de este trabajo se ha utilizado únicamente un libro de texto, considerado una referencia clásica en el área del Álgebra Lineal Numérica:

-Numerical Linear Algebra (Lloyd N. Trefethen, David Bau, III)

ISBN: 978-0-898713-61-9, Editorial: Society for Industrial and Applied Mathematics

En dicho libro hemos encontrado la teoría correspondiente, las descripciones (en ocasiones muy esquemáticas) de los algoritmos, y algunos ejemplos importantes con los que comprobar su eficacia.

Aparte de esta referencia, se ha utilizado con frecuencia la ayuda del programa Matlab.