

El algoritmo QR con desplazamiento

Objetivos

- Explicar claramente el algoritmo QR con desplazamiento, haciendo énfasis en los aspectos teóricos y computacionales relevantes,
- implementar el algoritmo QR con desplazamiento en Julia,
- realizar experimentos numéricos bien diseñados, que permitan ilustrar el comportamiento del método bajo diferentes condiciones,
- analizar y discutir los resultados, con observaciones fundamentadas tanto en la teoría como en los datos obtenidos.
- resumir los hallazgos más relevantes y, plantear preguntas o ideas para trabajo futuro.

Teoría

Método QR básico y limitaciones

Esquema del algoritmo

El método QR es una iteración matricial basada en la descomposición QR:

Dado $A_0 = A$, se realiza en cada paso:

- Descomposición QR de A_k : $A_k = Q_k R_k$
- Cálculo de A_{k+1} : $A_{k+1} = R_k Q_k$

Notemos que cada iteración produce una matriz similar a la anterior: $A_{k+1} = Q_k^T A_k Q_k$

Por tanto, todas las A_k son similares a la original A , y tienen los mismos autovalores.

Bajo ciertas condiciones (e.g., A simétrica), las A_k convergen a una matriz diagonal cuya diagonal contiene los autovalores de A .

Convergencia

Este tema es tratado en la sección 7.3.3 del libro.

Sea $A \in \mathbb{C}^{n \times n}$ una matriz diagonalizable con valores propios ordenados por magnitud:

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

Supongamos que A tiene una descomposición de Schur:

$$A = Q_0 T_0 Q_0^H,$$

donde Q_0 es unitario y T_0 es triangular superior con los valores propios de A en la diagonal.

El algoritmo QR sin shift genera una secuencia de matrices $\{A^{(k)}\}$ mediante:

$$A^{(k)} = Q_k R_k, \quad A^{(k+1)} = R_k Q_k,$$

con lo cual:

$$A^{(k+1)} = Q_k^{-1} A^{(k)} Q_k.$$

Teorema (convergencia de subespacios):

Sea $e_1 \in \mathbb{C}^n$ el primer vector canónico. Si A tiene vectores propios $\{x_i\}$, con $Ax_i = \lambda_i x_i$, entonces:

$$\lim_{k \rightarrow \infty} A^{(k)} = T = \text{matriz triangular superior similar a } A,$$

y además, la subespacio generado por $A^{(k)}e_1$ converge al **autovector dominante** x_1 , con razón:

$$\left| \frac{\lambda_2}{\lambda_1} \right|^k.$$

Más específicamente, los elementos subdiagonales de $A^{(k)}$ se atenúan con razón lineal:

$$|(A^{(k)})_{i+1,i}| = O\left(\left|\frac{\lambda_{i+1}}{\lambda_i}\right|^k\right).$$

Es decir que el algoritmo QR sin shift **converge linealmente**, y su velocidad depende del cociente entre valores propios consecutivos. La convergencia puede ser muy lenta si los valores propios están cercanos en magnitud.

Estabilidad del método QR básico

El método QR se basa en transformaciones ortogonales (o unitarias, en el caso complejo), las cuales son numéricamente estables porque **preservan la norma 2** y no amplifican los errores de redondeo. En cada paso, se realiza una transformación de similaridad de la forma:

$$A_{k+1} = Q_k^\top A_k Q_k$$

Estas transformaciones no degradan la condición del problema ni introducen inestabilidad inherente. Por esta razón, el método QR es considerado numéricamente estable: los autovalores obtenidos son aproximaciones fiables de los autovalores exactos de la matriz original, dentro de los límites del error de redondeo.

Precisión del método QR básico

La precisión en los autovalores calculados depende principalmente de:

- La separación entre los autovalores (problemas con autovalores cercanos presentan mayor sensibilidad).
- La acumulación de errores en la descomposición QR (influenciada por la técnica empleada: Householder, Givens, MGS, etc.).

Según el análisis de Golub y Van Loan, el método QR aplicado a una matriz simétrica produce autovalores $\hat{\lambda}_i$ tales que:

$$|\hat{\lambda}_i - \lambda_i| \approx u \|A\|_2$$

donde u es la unidad de redondeo (por ejemplo, $u \approx 10^{-16}$ en doble precisión), y $\|A\|_2$ es la norma espectral de la matriz.

Método QR con reducción previa a forma Hessenberg

Matrices de Hessenberg

Una matriz $H \in \mathbb{R}^{n \times n}$ se dice que está en **forma de Hessenberg superior** si:

$$h_{ij} = 0 \quad \text{para } i > j + 1$$

Es decir, todos los elementos **por debajo de la subdiagonal** son cero.

Por ejemplo, una matriz de Hessenberg de orden 4 tiene la forma:

$$H = \begin{bmatrix} * & * & * & * \\ * & * & * & * \\ 0 & * & * & * \\ 0 & 0 & * & * \end{bmatrix}$$

Propiedades:

- Si la matriz original A es simétrica, la forma de Hessenberg será **tridiagonal**.

Algoritmo

Dado $A \in \mathbb{R}^{n \times n}$, el algoritmo comienza con una **reducción ortogonal** a forma de Hessenberg:

$$A = Q^T H Q$$

donde Q es ortogonal y H es Hessenberg. A partir de allí, se aplican iteraciones QR:

1.
$$H_k = Q_k R_k$$

(factorización QR de la matriz desplazada),
2.
$$H_{k+1} = R_k Q_k$$

Gracias a que H_k es Hessenberg, estas operaciones se realizan en tiempo reducido, y la estructura se conserva en cada paso.

Ventajas

En el método QR (con o sin desplazamiento), cada iteración requiere calcular la factorización QR de la matriz actual A_k . Si A_k es densa, esta operación cuesta $O(n^3)$ flops. En cambio:

Si A_k tiene forma de Hessenberg, su factorización QR puede realizarse en $O(n^2)$ flops usando **rotaciones de Givens**, que operan sólo sobre los elementos no nulos.

Además:

- La forma de Hessenberg **se preserva** bajo cada iteración QR con desplazamiento:

$$A_k - \mu_k I = Q_k R_k \quad \Rightarrow \quad A_{k+1} = R_k Q_k + \mu_k I$$

implica que si A_k es Hessenberg, entonces A_{k+1} también lo es.

Esto significa que **basta con reducir la matriz original una sola vez** a forma de Hessenberg antes de iniciar el proceso iterativo.

Método QR con desplazamiento

Motivación

Dado que la velocidad de convergencia del método QR básico es el cociente entre los dos primeros autovalores, este método puede presentar **convergencia lenta** cuando los autovalores de la matriz están cercanos entre sí. Esta lentitud se traduce en una mayor cantidad de iteraciones y, por tanto, un mayor costo computacional.

Definición

Se realiza sobre matrices de Hessenberg.

Sea $A \in \mathbb{R}^{n \times n}$ una matriz real. El **método QR con desplazamiento** es una variante del algoritmo QR clásico que acelera la convergencia mediante la incorporación de un escalar $\mu_k \in \mathbb{R}$ en cada iteración.

Esquema general de la iteración

Dado $A_0 := A$, para cada $k \geq 0$ realizamos:

1. **Aplicar el shift:**

$$A_k - \mu_k I = Q_k R_k$$

donde Q_k es ortogonal y R_k es triangular superior.

2. **Actualizar la matriz:**

$$A_{k+1} = R_k Q_k + \mu_k I$$

Justificación algebraica del shift estático

Esta operación puede reescribirse como: $A_{k+1} = Q_k^T A_k Q_k$, debido a que:

$$\begin{aligned} & Q_k^T A_k Q_k \\ &= Q_k^T (Q_k R_k + \mu_k I) Q_k \\ &= Q_k^T Q_k R_k Q_k + \mu_k Q_k^T Q_k \\ &= R_k Q_k + \mu_k I \\ &= A_k \end{aligned}$$

Esto implica que $A_{k+1} \sim A_k \sim A$: la matriz resultante es similar a la anterior, y por tanto **preserva los autovalores**. En otras palabras, el método QR con shift genera una sucesión de matrices similares entre sí.

Tipos de desplazamiento

Los desplazamientos μ_k pueden clasificarse según cómo se seleccionan en cada iteración:

Desplazamiento estático

Se fija un valor constante $\mu_k = \mu$ para todas las iteraciones. Es fácil de implementar, pero su eficacia depende fuertemente de que (μ) esté cerca de algún autovalor dominante. La convergencia sigue siendo lineal en general.

Desplazamiento dinámico

El valor de μ_k se **actualiza en cada iteración** en función del contenido espectral de la matriz actual A_k . Este tipo de desplazamiento busca adaptarse dinámicamente a la estructura de la matriz para acelerar la convergencia.

Entre los desplazamientos dinámicos más comunes se encuentran:

- **Shift de Rayleigh:** Usa una estimación del autovalor dominante basada en el cociente de Rayleigh asociado a un vector aproximado.
- **Shift de Wilkinson:** Basado en los autovalores de la submatriz 2×2 en la esquina inferior derecha de A_k .
- **Multishift (shifts múltiples):** Usa varios desplazamientos simultáneamente para mejorar la convergencia global, especialmente en implementaciones paralelas o matrices grandes.

Convergencia

El uso de desplazamientos en el método QR tiene un impacto directo en la velocidad de convergencia hacia la forma triangular (o diagonal, en el caso simétrico). A diferencia del QR sin desplazamiento, cuya convergencia es lineal, los desplazamientos permiten obtener tasas **aceleradas** de convergencia, especialmente cuando se emplean estrategias dinámicas.

Shift estático

Como se analizó en la sección 7.5.2 de *Matrix Computations*, si se utiliza un desplazamiento constante μ , la convergencia del método sigue siendo **lineal**, y su velocidad depende del cociente entre los valores propios modificados por el shift. En particular, si $\lambda_1, \lambda_2, \dots$ son los autovalores de A ordenados por cercanía a μ , entonces la velocidad de convergencia de la componente asociada a λ_1 es proporcional a:

$$\left| \frac{\lambda_2 - \mu}{\lambda_1 - \mu} \right|^k$$

Por tanto, para que el método converja rápidamente hacia λ_1 , es deseable que μ esté lo más cerca posible de este autovalor.

Shift dinámico

Cuando el shift μ_k se actualiza en cada iteración de acuerdo a la evolución espectral de la matriz A_k , se puede lograr **convergencia cuadrática** en casos favorables. El ejemplo más importante es el **shift de Wilkinson**, aplicado a matrices simétricas tridiagonales:

En este caso, si

- la matriz A_k es simétrica tridiagonal,
- y se utiliza el shift de Wilkinson,

entonces la entrada subdiagonal $a_{n,n-1}^{(k)}$ decrece cuadráticamente:

$$|a_{n,n-1}^{(k)}| = O\left(|a_{n,n-1}^{(k-1)}|^2\right)$$

Este comportamiento acelerado permite realizar **deflación efectiva**, es decir, aislar autovalores individuales tras pocas iteraciones, y aplicar el método recursivamente sobre bloques más pequeños.

Comparación

Tipo de shift	Velocidad de convergencia	Complejidad por iteración	Adaptabilidad
Ninguno	Lineal	$O(n^2)$ con Hessenberg	Ninguna
Estático	Lineal (mejorada)	$O(n^2)$	Fija
Dinámico (Wilkinson)	Cuadrática	$O(n^2)$	Alta

Análisis computacional

Implementación y experimentos

La implementación de los algoritmos y los experimentos realizados pueden ser consultados en el anexo correspondiente. En él se detallan las variantes del método QR implementadas y los criterios numéricos utilizados para evaluar su desempeño.

Análisis de resultados computacionales

En la sección de experimentos se graficaron tres métricas relevantes para evaluar la eficiencia y precisión de los algoritmos:

- El **número de iteraciones promedio** hasta convergencia,
- La **norma de la subdiagonal promedio**, y
- El **error espectral promedio**,

para matrices tanto **simétricas** como **no simétricas**, todas con espectro real. Cada punto experimental corresponde al promedio de 5 muestras aleatorias del tamaño de matriz indicado, con el objetivo de observar tendencias generales.

Iteraciones

El **número de iteraciones** nos permite visualizar el comportamiento de convergencia de cada método:

- En matrices **simétricas**, el método `no_shift` fue el que más rápido convergió.
- En matrices **no simétricas**, los métodos `shift_hnn` y `shift_mean` mostraron un mejor desempeño conforme aumentó el tamaño de la matriz.
- Los métodos `hessenberg` y `shift_static` se ubicaron en un punto intermedio.

A primera vista, podría parecer sorprendente que los desplazamientos espectrales mejoren más en matrices no simétricas que en simétricas. Sin embargo, esto se explica por las características de las matrices generadas:

- Las matrices simétricas fueron generadas con $A + A'$, lo que produce matrices densas, simétricas pero **no tridiagonales**. En este caso, el algoritmo QR con desplazamiento **no alcanza la convergencia cuadrática** que se observa en matrices tridiagonales, donde el shift de Wilkinson es más eficaz.
- Las matrices no simétricas fueron construidas como $A = Q\Lambda Q^{-1}$, con Λ diagonal real. Esto garantiza que todas tengan **espectro real y bien condicionado**, y por tanto, el desplazamiento (incluso aproximado) puede acelerar eficazmente la convergencia.

En resumen, el comportamiento observado está en línea con la teoría: **el desplazamiento dinámico funciona mejor cuando puede explotar información espectral útil**, lo cual no sucede completamente en matrices simétricas densas pero sí en matrices no simétricas bien estructuradas.

Norma de la subdiagonal

La **norma de la subdiagonal** fue utilizada como criterio de parada. Aunque teóricamente permite evaluar qué tan cerca está la matriz de ser triangular, en estos experimentos la mayoría de las matrices alcanzaron convergencia completa, por lo que esta métrica no aportó información diferenciadora significativa en esta prueba.

Precisión espectral

La **precisión espectral**, medida como el error relativo promedio en los autovalores obtenidos, mostró:

- Diferencias mínimas entre métodos (excepto `no_shift`, ligeramente peor en casos no simétricos).
- Mejores resultados en matrices **no simétricas** que en las simétricas, lo cual puede deberse a una **mayor separación espectral promedio** y mejor condicionamiento de las matrices generadas como $Q\Lambda Q^{-1}$.

Resultados

A lo largo de este trabajo se ha estudiado el método QR para el cálculo de autovalores reales, con énfasis en su versión con desplazamiento espectral. Se ha desarrollado una presentación teórica cuidadosa del algoritmo QR básico, sus limitaciones, y cómo la incorporación de desplazamientos (shifts) mejora su comportamiento. Los aspectos computacionales fueron explorados en un cuaderno anexo mediante distintas implementaciones y experimentos controlados.

Desde el punto de vista **teórico**, el método QR con desplazamiento:

- Conserva el espectro de la matriz original mediante transformaciones ortogonales.
- Acelera la convergencia en comparación con el método QR sin desplazamiento, especialmente cuando los autovalores están cercanos.
- Al emplear desplazamientos dinámicos, como el shift de Wilkinson, se puede lograr convergencia cuadrática en matrices simétricas tridiagonales.
- Su eficiencia mejora significativamente cuando se aplica a matrices previamente reducidas a forma de Hessenberg.

Desde el punto de vista **computacional**, los experimentos mostraron que:

- El QR sin desplazamiento (`no_shift`) funciona bien en matrices simétricas pequeñas, pero escala peor que las versiones con desplazamiento.
- El desplazamiento dinámico (`shift_hnn`) mostró mejor rendimiento en matrices **no simétricas**, lo cual puede explicarse porque dichas matrices fueron generadas con espectro real y bien condicionado, permitiendo al algoritmo separar eficientemente los autovalores.
- En matrices simétricas densas (no tridiagonales), el beneficio del desplazamiento dinámico fue menor, debido a que no se alcanza la convergencia cuadrática teóricamente esperada.
- La reducción previa a forma de Hessenberg permitió realizar iteraciones más eficientes sin pérdida de precisión.
- Todos los métodos (excepto `no_shift`) mostraron una precisión espectral comparable, con errores pequeños y estables a lo largo de los experimentos.

En conjunto, estos resultados confirman las predicciones teóricas: el uso de desplazamientos espectrales en el método QR mejora la eficiencia sin sacrificar la estabilidad ni la precisión, especialmente cuando se combina con una estructura numérica favorable como la forma de Hessenberg. La elección del tipo de desplazamiento (estático o dinámico) y su implementación práctica deben adaptarse al tipo de matriz y a las características del espectro esperadas.

Reflexión

Este proyecto representó un ejercicio completo en el análisis teórico, la implementación computacional y la evaluación experimental de un algoritmo numérico clásico. A lo largo del desarrollo, se enfrentaron desafíos tanto conceptuales como técnicos, que permitieron consolidar aprendizajes valiosos en álgebra lineal numérica y cómputo científico.

Una de las principales dificultades fue comprender en profundidad el impacto de los desplazamientos espectrales en la convergencia del método QR. Si bien la teoría (especialmente en el caso simétrico tridiagonal) es clara, su comportamiento práctico depende fuertemente de la estructura espectral y de la forma de la matriz. Esto se reflejó en los experimentos, donde algunos resultados desafiaron las expectativas iniciales, especialmente en el caso de matrices no simétricas.

Desde el punto de vista computacional, fue importante diseñar y controlar los experimentos de manera reproducible, equilibrando la variabilidad inherente a los datos aleatorios con la necesidad de obtener resultados estadísticamente representativos. Además, implementar versiones eficientes del algoritmo (por ejemplo, con reducción a forma de Hessenberg) exigió atención cuidadosa a los detalles numéricos y estructurales.

A nivel personal, el proyecto reforzó la importancia de conectar teoría y práctica en problemas numéricos. Aprendí a interpretar con mayor madurez los resultados de un algoritmo, considerando tanto sus fundamentos matemáticos como sus condiciones de uso realistas. También adquirí mayor familiaridad con herramientas computacionales para simular, visualizar y comparar algoritmos iterativos, siguiendo buenas prácticas de código.

Trabajo futuro

El proyecto deja abiertas varias posibilidades de extensión y mejora, tanto en la profundidad del análisis como en la variedad de experimentos:

- **Ampliar el tamaño de las matrices y el número máximo de iteraciones**, para explorar cómo escalan los algoritmos en casos más demandantes y confirmar tendencias asintóticas de convergencia.
- **Sustituir los promedios por visualizaciones individuales (como diagramas de dispersión)**, con el fin de detectar y excluir outliers que puedan distorsionar el análisis global. Esto permitiría observar la variabilidad interna de los resultados con mayor precisión.
- **Comparar distintas estrategias de desplazamiento dinámico**, incluyendo el shift de Wilkinson verdadero en matrices tridiagonales y experimentos con matrices mal condicionadas o con espectro complejo.
- **Estudiar la estabilidad numérica de los métodos en presencia de perturbaciones**, analizando la sensibilidad de los autovalores aproximados frente a errores en los datos o en los cálculos intermedios.

Estas direcciones futuras permitirían consolidar una comprensión aún más profunda del método QR y su comportamiento computacional, y abrir la puerta a comparaciones con otros métodos espectrales avanzados.

Declaración de fuentes externas

Uso de IA

Durante el desarrollo de este trabajo se utilizó el modelo de lenguaje ChatGPT (OpenAI) como herramienta de apoyo académico. En particular, la IA fue empleada para:

- **Redacción técnica y explicación teórica:** Solicité apoyo para redactar secciones del informe relacionadas con la teoría del método QR, incluyendo motivación, definición formal, análisis de convergencia, estructura de Hessenberg y desplazamientos espectrales (estáticos y dinámicos).
- **Revisión del estilo académico:** La IA fue utilizada como asistente editorial para mejorar la claridad, organización y precisión de las secciones escritas.
- **Apoyo conceptual en interpretación de resultados:** Se consultó a la IA para analizar comportamientos inesperados observados en los experimentos, especialmente la diferencia entre el desempeño del QR con desplazamiento en matrices simétricas y no simétricas.
- **Sugerencias de estructura del documento:** Recibí ayuda para organizar el informe en secciones coherentes: teoría, análisis computacional, resultados, reflexión y declaración de fuentes.
- **Código (mínimamente):** Si bien el código fue desarrollado directamente por el autor, se consultó a la IA para aclarar aspectos conceptuales de algunas funciones (por ejemplo, generación de matrices con espectro real o estructura simétrica).

Prompts principales utilizados

A lo largo del proceso, se formularon consultas como:

- *"Explica el método QR con desplazamiento con base en el libro de Golub y Van Loan."*
- *"Justifica algebraicamente por qué el QR con shift conserva el espectro."*
- *"Redacta una sección de reflexión sobre las dificultades y aprendizajes del proyecto."*
- *"¿Por qué el QR con shift puede comportarse mejor en matrices no simétricas?"*

Todas las respuestas obtenidas fueron evaluadas críticamente, reescritas cuando fue necesario y validadas.

Recursos utilizados

- **Libro guía:** Golub, G. H., & Van Loan, C. F. (2013). *Matrix Computations* (4th ed.). Johns Hopkins University Press. Secciones 7.3, 7.4 y 7.5.2 fueron las principales referencias teóricas.
- **Lenguaje Julia y librerías estándar:** Para implementar algoritmos QR y realizar experimentos numéricos, se utilizaron los paquetes estándar `LinearAlgebra` y `Plots` en el entorno `Pluto.jl`.

Declaración ética

Este trabajo se ha desarrollado respetando principios de honestidad académica y responsabilidad intelectual:

- El uso de IA se ha declarado de forma transparente.
- Toda fuente externa consultada ha sido debidamente citada.
- Las ideas, interpretaciones y decisiones sobre el contenido final son responsabilidad del autor.
- El trabajo ha sido realizado con el propósito de aprender, comprender y aplicar técnicas de

álgebra lineal numérica, no de delegar su ejecución.

