6. El algoritmo QR

6.1. ¿Cómo reducir las ineficiencias del método QR simple?

6.1.1. Reducir el trabajo hecho en un paso del ciclo.
→ El algoritmo QR con matrices Hessenberg.

¿De dónde salio la idea?
Sean B, C matrices de la forma triangular superior. Entonces

- La descomposición QR de B es simple (tiene $Q = I$, ignorando signos).
- El producto $B * C$ es triangular superior.
- A ver que pasa cuando una de las matrices tiene una subdiagonal ...

Definición 9. Una matriz tiene forma Hessenberg (superior) si

$$\begin{bmatrix} * & * & \cdots & \cdots & * \\ * & * & \cdots & \cdots & * \\ & * & \ddots & & \vdots \\ & & \ddots & \ddots & \vdots \\ & & & * & * \end{bmatrix} \quad \textit{i.e.} \quad a_{ij} = 0 \ \text{ para } \ i > j+1\,.$$

Nota. Sea A Hessenberg, B triangular superior. Entonces

1. El producto $A * B$ es de forma Hessenberg (ejercicio) y requiere $\approx \frac{1}{2}O(n^3)$ flops.
2. La descomposición QR de una matriz A (Hessenberg) requiere $\approx O(n^2)$ flops.
   → "the QR decomposition of a Hessenberg matrix"
3. El algoritmo QR preserva la forma Hessenberg de A_m.

   Demostración. Solo para $A_0 := A$ regular.
   La descomposición QR ($A_0 = QR$) da que tanto R, como R^{-1}, es regular y triangular superior. Entonces $Q = A_0 R^{-1}$ es de forma Hessenberg, puesto que es un producto de A_0 (Hessenberg) y R^{-1} (triangular superior), ver a).
   Entonces $A_1 = RQ$ es Hessenberg. □

4. Poner una matriz en forma Hessenberg es un método directo (número finito de pasos). Más bonito, para $A \in \mathbb{R}^{n\times n}$ existe $Q \in \mathbb{R}^{n\times n}$ ortogonal tal que $Q^\top AQ = H$ es de forma Hessenberg. En MatLab: `[Q, H] = hess(A);`

¿Cómo aceleramos? → Vemos que A y su forma Hessenberg H son similares unitarias. Entonces, A y H tienen los mismos eigenvalores. Por lo tanto, iniciamos el ciclo con $A_0 := H$ donde H es la forma Hessenberg superior de A. Debido a las notas 1–3 ahorramos trabajo en cada paso del ciclo, si realmente evitamos a calcular las entradas que valen 0.

↑ día 6

6.1.2. Acelerar la convergencia (reducir el numero de pasos).

En el ejercicio (de implementar) QR simple podemos observar que

$$a_{2,1}^{(m)} \to 0 \quad \text{linealmente con radio} \quad r_2 = \frac{1}{3} = \left|\frac{\lambda_2}{\lambda_1}\right| .$$

En general, si consideramos A en forma Hessenberg con

$$|\lambda_1| \geq |\lambda_2| \geq \ldots \geq |\lambda_n|$$

y una secuencia de matrices $\{A_m\}_m \subset \text{hess}(\mathbb{C}^{n\times n})$ generada por el algoritmo QR simple, entonces se puede demostrar que

$$|\lambda_j| > |\lambda_{j+1}| \implies a_{j+1,j}^{(m)} \to 0 \quad \text{linealmente con radio} \quad r_j = \left|\frac{\lambda_{j+1}}{\lambda_j}\right| .$$

Intuición, cada paso del QR simple construye una matriz Q tal que

$$A_+ = Q^H A Q \, .$$

Ideal seria encontrar $Q = [Q_\star \; \boldsymbol{q}]$ tal que $\boldsymbol{q}^H A = \lambda \boldsymbol{q}^H$, pues en ese caso

$$A_+ = \begin{pmatrix} Q_\star^H A Q_\star & Q_\star^H A \boldsymbol{q} \\ 0 & \lambda \end{pmatrix} .$$

Pero, Q es definido por

$$A = QR \iff Q^H A = R$$

y el último renglón de este sistema es

$$\boldsymbol{e}_n^\top Q^H A = \boldsymbol{e}_n^\top R \iff \boldsymbol{q}^H A = r_{nn} \boldsymbol{e}_n^\top \iff A^H \boldsymbol{q} = r_{nn} \boldsymbol{e}_n \, ,$$

un paso del método de la potencia inversa (con $\boldsymbol{q}_0 = \boldsymbol{e}_n$). Por lo tanto, la columna $\boldsymbol{q}$ se acerca con la velocidad de ese método hacia el eigenvector (de la izq.) ideal.

¿Cómo acelerar? $\to$ emplear un *shift* como en el método de la potencia, es decir, cambiar A por $\tilde{A} = (A - \kappa I)$. El *shift* puede ser dinámico (un cociente de Rayleigh).

Nota. Supongamos que escogemos un buen *shift* ($0 \approx |\lambda_n - \kappa| \ll |\lambda_j - \kappa|$, $j \neq n$), entonces después de unos pasos del QR simple tenemos

$$\tilde{A}_m = \begin{pmatrix} \tilde{B}_m & \boldsymbol{h} \\ \approx \boldsymbol{0}^\top & \tilde{a}_{nn}^{(m)} \end{pmatrix} \quad \text{y} \quad A_m = \tilde{A}_m + \kappa I = \begin{pmatrix} B_m & \boldsymbol{h} \\ \approx \boldsymbol{0}^\top & a_{nn}^{(m)} \end{pmatrix} ,$$

una matriz diagonal en bloques. Por similitud $a_{nn}^{(m)}$ aproxima el eigenvalor de A más cerca de κ. El resto del espectro de A se encuentra en B_m. Para economizar, podemos trabajar sólo con B_m. Continuando así, operando cada vez con matrices mas pequeñas (*Deflación*), obtenemos los otros eigenvalores de A.

6.1.3. ¿Cómo elegir los shifts*?*

- transformar A a forma Hessenberg
- iterar unas pocas veces con QR simple lleva la matriz cerca a una triangular superior con eigenvalores ordenados talque

$$A_m = \begin{pmatrix} B_m & \boldsymbol{h} \\ \boldsymbol{g}^H & a_{nn}^{(m)} \end{pmatrix} \quad \text{y} \quad \|\boldsymbol{g}\| \ll 1\,.$$

  Entonces $\kappa_m \coloneqq a_{nn}^{(m)}$ es un buen shift.

¿Cómo aproximar un eigenvalor?

Un paso: QR con shift dinámico (cociente de Rayleigh)

Sea $A_1 \in \mathbb{R}^{k\times k}$ (Hessenberg)

Para $m = 1, 2, \ldots$ hasta $a_{k,k-1}^{(m)} < tol$

- $\kappa_m \coloneqq a_{kk}^{(m)}$
- $Q_m R_m \coloneqq A_m - \kappa_m I$ (Factorización QR)
- $A_{m+1} \coloneqq R_m Q_m + \kappa_m I$

End

Ejercicio 7.

a. Compruebe que las matrices A_m y A_{m+1} son similares unitarias.
b. Ver que $\kappa = a_{kk}$ es el cociente de Rayleigh asociado al vector $\boldsymbol{e}_k^\top = (0, \ldots, 0, 1)$.
c. Ver que el vector $\boldsymbol{e}_k$ aproxima un eigenvector de $A^\top$ si $|a_{k,k-1}^{(m)}| \ll |a_{kk}^{(m)}|$.

Nota. La convergencia es cuadrática, como en la iteración de Rayleigh (cúbica para $A = A^H$).

Ejercicio 8. Iniciar directamente y tomar como *shift* el cociente de Rayleigh $\kappa_m \coloneqq a_{nn}^{(m)}$ no siempre funciona. Por ejemplo, considere la matriz

$$A = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}$$

y aplique un paso del algoritmo QR con *shift* de Rayleigh. ¿Qué pasa?

Algoritmo: QR con shift dinámico

Entradas: $A \in \mathbb{R}^{n\times n}$ con eigenvalores $|\lambda_1| > |\lambda_2| > \ldots > |\lambda_n|$ y una tolerancia tol.

Salidas: TAREA!

Pseudo code:

$A = hess(A)$;
% haz un paso con QR simple
$k = n$;
mientras $k > 1$
  mientras $|a_{k,k-1}| > tol$ % un paso
    · $\kappa = a_{kk}$;
    · $[Q, R] = qr(A - \kappa I)$; % Factorización QR
    · $A \coloneqq R * Q + \kappa I$;
  end
  $\lambda_k \coloneqq a_{kk}$;
  $k \coloneqq k - 1$;
  $A \coloneqq A(1 : k, 1 : k)$; % Deflación
end
$\lambda_1 \coloneqq a_{1,1}$

Nota.

- Matrices singulares no son un problema. Si de casualidad κ es un eigenvalor, entonces un paso del algo. QR con ese *shift* es suficiente para extraer ese eigenvalor.
- Si $A \in \mathbb{R}^{n\times n}$, entonces el método se queda en los reales, no puede aproximar eigenvalores complejos. La solución para obtener eigenvalores complejos es usar el *shift de Wilkinson*, el cual es un eigenvalor de la última submatriz 2×2. Cuando esa matriz es

$$A_m = \begin{bmatrix} & \\ & \begin{bmatrix} \alpha & \beta \\ -\beta & \alpha \end{bmatrix} \end{bmatrix},$$

entonces $\lambda = \alpha + i\beta$ y $\bar{\lambda} = \alpha - i\beta$ son eigenvalores y es común tomar los dos para evitar calcular en números complejos.

↑ día 7