

Eigenvalues and Eigenvectors



Numerical Analysis

Proffs. Gianluigi Rozza - Luca Heltai

2019-SISSA mathLab Trieste

Eigenvalues and Eigenvectors

Let $A \in \mathbb{C}^{n \times n}$, the eigenvalue problem consists in finding a scalar λ (real or complex) and a nonnull vector \mathbf{x} such that

$$A\mathbf{x} = \lambda\mathbf{x} \quad (1)$$

λ is called an **eigenvalue** of A , while \mathbf{x} is the associated **eigenvector**. The latter is not unique; indeed $\alpha\mathbf{x}$ with $\alpha \neq 0$, are also eigenvectors associated with λ . Should \mathbf{x} be known, λ can be recovered by using the **Rayleigh quotient**

$$\mathbf{x}^H A\mathbf{x} / \|\mathbf{x}\|^2.$$

The eigenvalues of A are the roots of the **characteristic polynomial** of A :

$$p_A(\lambda) = \det(A - \lambda I).$$

A $n \times n$ matrix has exactly n eigenvalues (real or complex), not necessarily distinct. A matrix $A \in \mathbb{C}^{n \times n}$ is said to be diagonalizable if there exists a nonsingular matrix $U \in \mathbb{C}^{n \times n}$ such that $U^{-1}AU = \Lambda = \text{diag}(\lambda_1, \dots, \lambda_n)$.

The power method

Let A with real entries and assume that its eigenvalues are ordered

$$|\lambda_1| > |\lambda_2| \geq |\lambda_3| \geq \dots \geq |\lambda_n|. \quad (2)$$

The **power method** is: Let $\mathbf{x}^{(0)} \in \mathbb{C}^n$ be given and set $\mathbf{y}^{(0)} = \mathbf{x}^{(0)} / \|\mathbf{x}^{(0)}\|$, for $k = 1, 2, \dots$ compute

$$\mathbf{x}^{(k)} = A\mathbf{y}^{(k-1)}, \quad \mathbf{y}^{(k)} = \frac{\mathbf{x}^{(k)}}{\|\mathbf{x}^{(k)}\|}, \quad \lambda^{(k)} = (\mathbf{y}^{(k)})^H A\mathbf{y}^{(k)}$$

Until $|\lambda^{(k)} - \lambda^{(k-1)}| < \varepsilon |\lambda^{(k)}|$, where ε is the desired tolerance.

Convergence

Since we have assumed that the eigenvectors $\mathbf{x}_1, \dots, \mathbf{x}_n$ of A are linearly independent, these eigenvectors form a basis for \mathbb{C}^n . Thus the vectors $\mathbf{x}^{(0)}$ and $\mathbf{y}^{(0)}$ can be written as

$$\mathbf{x}^{(0)} = \sum_{i=1}^n \alpha_i \mathbf{x}_i, \quad \mathbf{y}^{(0)} = \beta^{(0)} \sum_{i=1}^n \alpha_i \mathbf{x}_i, \quad \text{with } \beta^{(0)} = 1/\|\mathbf{x}^{(0)}\| \text{ and } \alpha_i \in \mathbb{C}.$$

At the first step the power method gives

$$\begin{aligned} \mathbf{x}^{(1)} = A\mathbf{y}^{(0)} &= \beta^{(0)} A \sum_{i=1}^n \alpha_i \mathbf{x}_i = \beta^{(0)} \sum_{i=1}^n \alpha_i \lambda_i \mathbf{x}_i && \text{and, similarly,} \\ \mathbf{y}^{(1)} &= \beta^{(1)} \sum_{i=1}^n \alpha_i \lambda_i \mathbf{x}_i, && \beta^{(1)} = \frac{1}{\|\mathbf{x}^{(0)}\| \|\mathbf{x}^{(1)}\|}. \end{aligned}$$

At a given step k we will have

$$\mathbf{y}^{(k)} = \beta^{(k)} \sum_{i=1}^n \alpha_i \lambda_i^k \mathbf{x}_i, \quad \beta^{(k)} = \frac{1}{\|\mathbf{x}^{(0)}\| \dots \|\mathbf{x}^{(k)}\|}$$

Convergence

And therefore

$$\mathbf{y}^{(k)} = \lambda_1^k \beta^{(k)} \left(\alpha_1 \mathbf{x}_1 + \sum_{i=2}^n \alpha_i \frac{\lambda_i^k}{\lambda_1^k} \mathbf{x}_i \right).$$

Since $|\lambda_i/\lambda_1| < 1$ for $i = 2, \dots, n$, the vector $\mathbf{y}^{(k)}$ tends to align along the same direction as the eigenvector \mathbf{x}_1 when k tends to $+\infty$, provided $\alpha_1 \neq 0$.

Convergence

Example 1. Consider the family of matrices

$$A(\alpha) = \begin{bmatrix} \alpha & 2 & 3 & 13 \\ 5 & 11 & 10 & 8 \\ 9 & 7 & 6 & 12 \\ 4 & 14 & 15 & 1 \end{bmatrix}, \quad \alpha \in \mathbb{R}.$$

We want to approximate the eigenvalue with largest modulus by the power method. When $\alpha = 30$, the eigenvalues of the matrix are given by $\lambda_1 = 39.396$, $\lambda_2 = 17.8208$, $\lambda_3 = -9.5022$ and $\lambda_4 = 0.2854$.

The method approximates λ_1 in 22 iterations with a tolerance $\varepsilon = 10^{-10}$ and $\mathbf{x}^{(0)} = \mathbf{1}^T$.

If $\alpha = -30$ the iterations are 708. The reason is that $|\lambda_2|/|\lambda_1| = 0.9704$ is close to unity: $\lambda_1 = -30.643$, $\lambda_2 = 29.7359$, $\lambda_3 = -11.6806$ and $\lambda_4 = 0.5878$.

The inverse power method

The **inverse power method** can find the smaller eigenvalue of a non-singular matrix A :

Let $\mathbf{x}^{(0)} \in \mathbb{C}^n$ be given and set $\mathbf{y}^{(0)} = \mathbf{x}^{(0)} / \|\mathbf{x}^{(0)}\|$, for $k = 1, 2, \dots$ compute

$$\mathbf{x}^{(k)} = A^{-1} \mathbf{y}^{(k-1)}, \quad \mathbf{y}^{(k)} = \frac{\mathbf{x}^{(k)}}{\|\mathbf{x}^{(k)}\|}, \quad \mu^{(k)} = (\mathbf{y}^{(k)})^H A^{-1} \mathbf{y}^{(k)}$$

Until $|\mu^{(k)} - \mu^{(k-1)}| < \varepsilon |\mu^{(k)}|$, where ε is the desired tolerance.

If A admits n linearly independent eigenvectors, and if also the eigenvalue λ_n of minimum modulus is distinct from the others, then

$$\lim_{k \rightarrow \infty} \mu^{(k)} = 1/\lambda_n,$$

At each step k we have to solve a linear system of the form $A \mathbf{x}^{(k)} = \mathbf{y}^{(k-1)}$.

The power method with shift

The **power method with shift** can find the eigenvalue of A near to a given number μ :

Define $A_\mu = A - \mu I$, whose eigenvalues are $\lambda(A_\mu) = \lambda(A) - \mu$.

In order to approximate λ_μ , we can at first approximate the eigenvalue of minimum length of A_μ

Let $\mathbf{x}^{(0)} \in \mathbb{C}^n$ be given and set $\mathbf{y}^{(0)} = \mathbf{x}^{(0)} / \|\mathbf{x}^{(0)}\|$, for $k = 1, 2, \dots$ compute

$$\mathbf{x}^{(k)} = A_\mu^{-1} \mathbf{y}^{(k-1)}, \quad \mathbf{y}^{(k)} = \frac{\mathbf{x}^{(k)}}{\|\mathbf{x}^{(k)}\|}, \quad \lambda_\mu^{(k)} = 1 / (\mathbf{y}^{(k)})^H A_\mu^{-1} \mathbf{y}^{(k)}$$

Until $|\lambda_\mu^{(k)} - \lambda_\mu^{(k-1)}| < \varepsilon |\lambda_\mu^{(k)}|$, where ε is the desired tolerance. 1 The searched eigenvalue of A is approximated by $\lambda = \lambda_\mu + \mu$.

Example 2. For the matrix $A(30)$ of Example 1 we seek the eigenvalue closest to the value 17. We set $\mu = 17$ and apply the power method with shift with a tolerance $tol = 10^{-10}$ and initial guess $x_0 = (1, 1, 1, 1)^T$. After 8 iterations the algorithm returns the value $\lambda = 17.82079703055703$. A less accurate knowledge of the *shift* would involve more iterations. For instance, if we set $\mu = 13$ the program returns the value $\lambda = 17.82079703064106$ after 19 iterations.

The value of the shift can be modified during the iterations, by setting $\mu = \lambda^{(k)}$. This yields a faster convergence; however the computational cost grows substantially since now at each iteration the matrix A_μ does change and the LU factorization has to be performed at each iteration.

How to compute the shift

We need to locate (more or less accurately) the eigenvalues of A in the complex plane.

Let A be a square matrix of dimension n . The Gershgorin circles $C_i^{(r)}$ and $C_i^{(c)}$ associated with its i -th row and i -th column are respectively defined as

$$C_i^{(r)} = \{z \in \mathbb{C} : |z - a_{ii}| \leq \sum_{j=1, j \neq i}^n |a_{ij}|\},$$

$$C_i^{(c)} = \{z \in \mathbb{C} : |z - a_{ii}| \leq \sum_{j=1, j \neq i}^n |a_{ji}|\}.$$

$C_i^{(r)}$ is called the i -th **row circle** and $C_i^{(c)}$ the i -th **column circle**.

All the eigenvalues of a given matrix $A \in \mathbb{C}^{n \times n}$ belong to the region of the complex plane which is the intersection of the two regions formed respectively by the union of the row circles and column circles.

Moreover, should m row circles (or column circles), with $1 \leq m \leq n$, be disconnected from the union of the remaining $n - m$ circles, then their union contains exactly m eigenvalues.

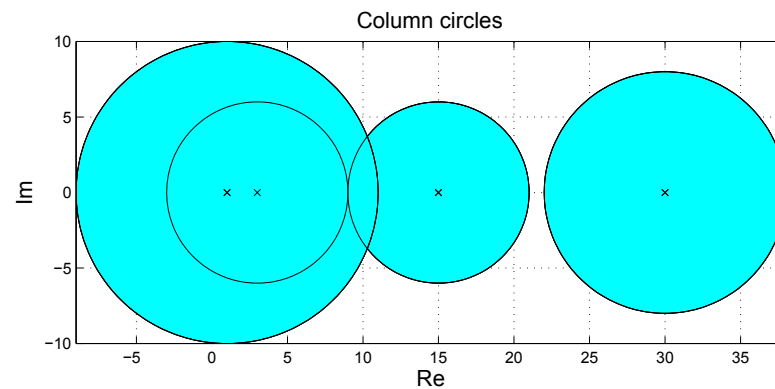
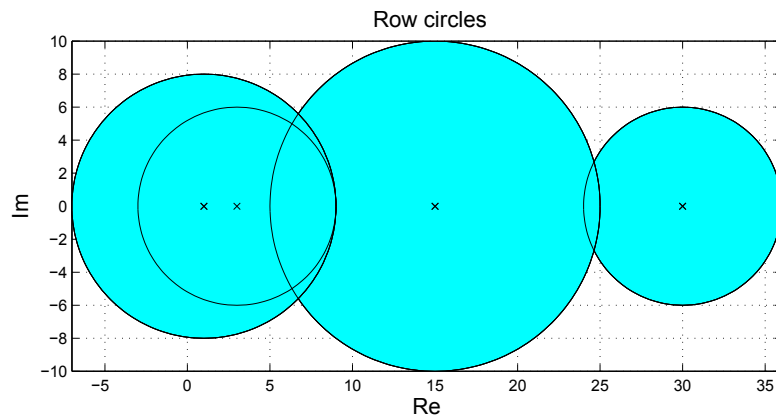
There is no guarantee that a circle should contain eigenvalues, unless it is isolated from the others. The information provided by Gershgorin circles are in general quite coarse, thus the previous result can provide only a preliminary guess of the shift.

Note that from Proposition, all the eigenvalues of a strictly diagonally dominant matrix are non-null.

Example 3. Below we have plotted the Gershgorin circles associated with the matrix

$$A = \begin{bmatrix} 30 & 1 & 2 & 3 \\ 4 & 15 & -4 & -2 \\ -1 & 0 & 3 & 5 \\ -3 & 5 & 0 & -1 \end{bmatrix}.$$

The centers of the circles have been identified by a cross.



Row circles (*left*) and column circles (*right*) for the matrix of Example