Lecture 3: Eigendecomposition Algorithms and Application

Wenfu Zhong

Supervisors: Ziping Zhao, Jie Lu
ShanghaiTech University

June 2024

Outline

Table of Contents

Power iteration

- Assumptions:
  - $\mathbf{A}$ admits an eigendecomposition $\mathbf{A} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^{-1}$
  - λ_i's are ordered such that $|\lambda_1| > |\lambda_2| \geq \cdots \geq |\lambda_n|$
  - We have an initial guess $\mathbf{q}^{(0)}$ that satisfies $\left[\mathbf{V}^{-1}\mathbf{q}^{(0)}\right]_1 \neq 0$ (random guess should do)
- Consider $\mathbf{A}^k\mathbf{x}$. Let $\boldsymbol{\alpha} = \mathbf{V}^{-1}\mathbf{q}^{(0)}$, and observe

$$\mathbf{A}^k\mathbf{q}^{(0)} = \mathbf{V}\mathbf{\Lambda}^k\mathbf{V}^{-1}\mathbf{q}^{(0)} = \sum_{i=1}^{n} \alpha_i \lambda_i^k \mathbf{v}_i = \alpha_1 \lambda_1^k \left(\mathbf{v}_1 + \underbrace{\sum_{i=2}^{n} \frac{\alpha_i}{\alpha_1} \left(\frac{\lambda_i}{\lambda_1}\right)^k \mathbf{v}_i}_{\mathbf{r}_k} \right)$$

  where $\mathbf{r}_k$ is a residual and has

$$\|\mathbf{r}_k\|_2 \leq \sum_{i=2}^{n} \left|\frac{\alpha_i}{\alpha_1}\right| \left|\frac{\lambda_i}{\lambda_1}\right|^k \|\mathbf{v}_i\|_2 \leq \left|\frac{\lambda_2}{\lambda_1}\right|^k \sum_{i=2}^{n} \left|\frac{\alpha_i}{\alpha_1}\right|$$

- Convergence: Let $c_k = \frac{|\alpha_1||\lambda_1|^k}{\alpha_1 \lambda_1^k}$, i.e., the sign of $\alpha_1\lambda_1^k$(note $|c_k| = 1$). We have

$$\lim_{k\to\infty} c_k \frac{\mathbf{A}^k\mathbf{q}^{(0)}}{\|\mathbf{A}^k\mathbf{q}^{(0)}\|_2} = \mathbf{v}_1$$

Power iteration

Algorithm 1: Power Iteration

Input: $\mathbf{A} \in \mathbb{C}^{n \times n}$ and a starting vector $\mathbf{q}^{(0)} \in \mathbb{C}^n$

$k = 0$;

repeat

$\tilde{\mathbf{q}}^{(k+1)} = \mathbf{A}\mathbf{q}^{(k)}$;

$\mathbf{q}^{(k+1)} = \tilde{\mathbf{q}}^{(k+1)} / \|\tilde{\mathbf{q}}^{(k+1)}\|_2$ % normalization;

$\lambda^{(k+1)} = R(\mathbf{q}^{(k+1)}) = (\mathbf{q}^{(k+1)})^H \mathbf{A}\mathbf{q}^{(k+1)}$;

$k := k + 1$;

until *a stopping rule is satisfied*;

Output: $\mathbf{q}^{(k)}, \lambda^{(k)}$

- It finds the dominant eigen-pair, i.e., dominant eigenvalue λ_1 (largest eigenvalue in modulus) and dominant eigenvector $\mathbf{v}_1$ only, unless $\alpha_1 = 0$.
- Complexity per iteration: $\mathcal{O}(n^2)$, or $\mathcal{O}(\text{nnz}(\mathbf{A}))$ for sparse $\mathbf{A}$.

Convergence (Demo)

Convergence rate depends on $\left|\frac{\lambda_2}{\lambda_1}\right|$,

- $\left\|\mathbf{q}^{(k)} - \mathbf{v}_1\right\|_2 = \mathcal{O}\left(\left|\frac{\lambda_2}{\lambda_1}\right|^k\right)$ and $\left|\lambda^{(k)} - \lambda_1\right| = \mathcal{O}\left(\left|\frac{\lambda_2}{\lambda_1}\right|^k\right)$ and $\left|\lambda^{(k)} - \lambda_1\right| = \mathcal{O}\left(\left|\frac{\lambda_2}{\lambda_1}\right|^k\right)$($\mathcal{O}\left(\left|\frac{\lambda_2}{\lambda_1}\right|^{2k}\right)$ for Hermitian $\mathbf{A}$)
- Slower if $|\lambda_2|$ is closer to λ_1, i.e., $\left|\frac{\lambda_2}{\lambda_1}\right|$is closer to 1
- Reduction per iteration is a constant, i.e., linear convergence

Exercise

Exercise 1. Given a matrix $\mathbf{A}$ generated with eigenvalues $[7, 5, 2, 4, 1]$ and random eigenvectors, restore its largest eigenvalue (i.e. 7) and the corresponding eigenvector by the power iteration.

Algorithm 2: Inverse Iteration

Input: $\mathbf{A} \in \mathbb{C}^{n \times n}$ and a starting vector $\mathbf{q}^{(0)} \in \mathbb{C}^n$

$k = 0$;

repeat

$\tilde{\mathbf{q}}^{(k+1)} = (\mathbf{A} - \mu \mathbf{I})^{-1} \mathbf{q}^{(k)}$ % solve $(\mathbf{A} - \mu \mathbf{I}) \tilde{\mathbf{q}}^{(k+1)} = \mathbf{q}^{(k)}$;

$\mathbf{q}^{(k+1)} = \tilde{\mathbf{q}}^{(k+1)} / \|\tilde{\mathbf{q}}^{(k+1)}\|_2$ % normalization ;

$\lambda^{(k+1)} = R(\mathbf{q}^{(k+1)}) = (\mathbf{q}^{(k+1)})^H \mathbf{A} \mathbf{q}^{(k+1)}$;

$k := k + 1$;

until *a stopping rule is satisfied*;

Output: $\mathbf{q}^{(k)}, \lambda^{(k)}$

- $(\mathbf{A} - \mu \mathbf{I})^{-1}$ is with eigenvalue $(\lambda_i - \mu)^{-1}$'s and the same eigenvector $\mathbf{v}_i$'s as $\mathbf{A}$
- Inverse (power) iteration with shift: apply power iteration on $(\mathbf{A} - \mu \mathbf{I})^{-1}$
- Complexity per iteration: $\mathcal{O}(n^2)$ (matrix $(\mathbf{A} - \mu \mathbf{I})$ is processed in advance)

Convergence (Demo)

- $\left\|\mathbf{q}^{(k)} - \mathbf{v}_J\right\|_2 = \mathcal{O}\left(\left(\max_{i=1,\ldots,n}\left|\frac{\lambda_J-\mu}{\lambda_i-\mu}\right|\right)^k\right)$
- $\left|\lambda^{(k)} - \lambda_J\right| = \mathcal{O}\left(\left(\max_{i=1,\ldots,n}\left|\frac{\lambda_J-\mu}{\lambda_i-\mu}\right|\right)^k\right)$, where λ_J is the closest eigenvalue to μ
- Reduction per iteration is a constant, i.e., linear convergence

Exercise

Exercise 2. Given a matrix $\mathbf{A}$ generated with eigenvalues $[7, 5, 2, 4, 1]$ and random eigenvectors, restore its second eigenvalue (i.e. 5) and the corresponding eigenvector by the inverse iteration. Choose an appropriate μ as you desire.

Rayleigh quotient iteration

Algorithm 3: Rayleigh Quotient Iteration

Input: $\mathbf{A} \in \mathbb{C}^{n\times n}$ and a starting vector $\mathbf{q}^{(0)} \in \mathbb{C}^n$

$k = 0$;

$\mu^{(k)} = R\left(\mathbf{q}^{(k)}\right)$;

repeat

$\tilde{\mathbf{q}}^{(k+1)} = (\mathbf{A} - \mu^{(k)}\mathbf{I})^{-1}\mathbf{q}^{(k)}$ % solve $(\mathbf{A} - \mu^{(k)}\mathbf{I})\tilde{\mathbf{q}}^{(k+1)} = \mathbf{q}^{(k)}$;

$\mathbf{q}^{(k+1)} = \tilde{\mathbf{q}}^{(k+1)} / \|\tilde{\mathbf{q}}^{(k+1)}\|_2$ % normalization ;

$\mu^{(k)} = \lambda^{(k+1)} = R(\mathbf{q}^{(k+1)}) = (\mathbf{q}^{(k+1)})^H \mathbf{A}\mathbf{q}^{(k+1)}$;

$k := k + 1$;

until *a stopping rule is satisfied*;

Output: $\mathbf{q}^{(k)}, \lambda^{(k)}$

- At least quadratic convergence, but uncertain to which eigenvalue it will converge
- Complexity per iteration: $\mathcal{O}\left(n^3\right)$

Convergence (Exercise)

Exercise 3. Given a matrix $\mathbf{A}$ generated with eigenvalues $[7, 5, 2, 4, 1]$ and random eigenvectors, plot the convergence curve which depicts the minimal estimation error from the eigenvalues by executing the Rayleigh quotient iteration on $\mathbf{A}$.

Deflation

Problem

How can we compute all the eigenvalues and eigenvectors?

- consider a Hermitian $\mathbf{A}$ with $|\lambda_1| > |\lambda_2| > \cdots > |\lambda_n|$, and note the outer-product representation

$$\mathbf{A} = \mathbf{V}\boldsymbol{\Lambda}\mathbf{V}^H = \sum_{i=1}^{n} \lambda_i \mathbf{v}_i \mathbf{v}_i^H.$$

- Hotelling's deflation: use the power iteration to obtain $\mathbf{v}_1$, λ_1, do the subtraction

$$\mathbf{A} := \mathbf{A} - \lambda_1 \mathbf{v}_1 \mathbf{v}_1^H = \sum_{i=2}^{n} \lambda_i \mathbf{v}_i \mathbf{v}_i^H,$$

and repeat until all the eigenvectors and eigenvalues are found.

Exercise

Exercise 4. Given a matrix $\mathbf{A}$ generated with eigenvalues $[7, 5, 2, 4, 1]$ and random eigenvectors, restore all the eigenvalues and eigenvectors of $\mathbf{A}$ by the three methods mentioned above utilizing the matrix deflation technology.

Discussion. What is the difference between the results obtained by the three methods? What is the problem with the Rayleigh quotient? Why?

Table of Contents

Orthogonal Iteration

- **subspace iteration**: starting with a set of linearly independent vectors or a subspace

$$\mathbf{Q}^{(0)} = \text{span}\{\mathbf{q}_1^{(0)}, \mathbf{q}_2^{(0)}, \ldots, \mathbf{q}_r^{(0)}\}, \quad \mathbf{Q}^{(k)} = \mathbf{A}^k\mathbf{Q}^{(0)}$$

  will converge (under suitable assumptions) to a subspace spanned by eigenvectors associated with the r largest eigenvalues in magnitude, i.e., the **dominant invariant subspace**. { in contrast, the power iteration is sometimes called **vector iteration** } { use thin QR to get the bases $\mathbf{Q}^{(k)}$ as $\mathbf{Q}^{(k)}\mathbf{R}^{(k)} = \mathbf{A}^k \begin{bmatrix} \mathbf{q}_1^{(0)} & \mathbf{q}_2^{(0)} & \cdots & \mathbf{q}_r^{(0)} \end{bmatrix}$ }
- the above subspace iteration is an **unnormalized simultaneous (power) iteration**; since all of $\{\mathbf{A}^k\mathbf{q}_1^{(0)}, \mathbf{A}^k\mathbf{q}_2^{(0)}, \ldots, \mathbf{A}^k\mathbf{q}_r^{(0)}\}$ will converge to a multiple of $\mathbf{v}_1$, columns of $\mathbf{Q}^{(k)}$ will form an extremely ill-conditioned basis for $\mathbf{Q}^{(k)}$.

Orthogonal Iteration

Suppose there is a gap between the $r\,(1 \le r \le n)$ largest eigenvalues and λ_{r+1} in magnitude, i.e., $|\lambda_1| \ge |\lambda_2| \ge \cdots \ge |\lambda_r| > |\lambda_{r+1}|$

Algorithm 4: Orthogonal Iteration

Input: $\mathbf{A} \in \mathbb{C}^{n\times n}$ and a starting semi-unitary matrix $\mathbf{Q}^{(0)} \in \mathbb{C}^{n\times r}$

$k = 0$;

repeat

$\tilde{\mathbf{Q}}^{(k+1)} = \mathbf{A}\mathbf{Q}^{(k)}$;

$\mathbf{Q}^{(k+1)}\mathbf{R}^{(k+1)} = \tilde{\mathbf{Q}}^{(k+1)}$ % orthogonalization; perform thin QR;

$\left\{\lambda_1^{(k+1)}, \lambda_2^{(k+1)}, \ldots, \lambda_r^{(k+1)}\right\} = \text{diag}\left(\left(\mathbf{Q}^{(k+1)}\right)^H \mathbf{A}\mathbf{Q}^{(k+1)}\right)$;

$k := k + 1$;

until *a stopping rule is satisfied*;

Output: $\mathbf{Q}^{(k)}, \left\{\lambda_1^{(k+1)}, \lambda_2^{(k+1)}, \ldots, \lambda_r^{(k+1)}\right\}$

Convergence (Demo)

- $\mathbf{Q}^{(k)}$ converges linearly to an orthonormal basis for the dominant invariant subspace associated with the r largest eigenvalues in magnitude $\mathcal{R}\left(\mathbf{U}\left(:,1:r\right)\right)$
- $\left[\lambda_1^{(k)}, \lambda_2^{(k)}, \ldots, \lambda_r^{(k)}\right] = \text{diag}\left(\left(\mathbf{Q}^{(k)}\right)^H \mathbf{A}\mathbf{Q}^{(k)}\right) \rightarrow [\lambda_1, \lambda_2, \ldots, \lambda_r]$
- $\left|\lambda_i^{(k)} - \lambda_i\right| = \mathcal{O}\left(\left(\max_{i=1,\ldots,r}\left|\frac{\lambda_{i+1}}{\lambda_i}\right|\right)^k\right), \quad i = 1, 2, \ldots, r$

Exercise

Exercise 5. Given a matrix $\mathbf{A}$ generated with eigenvalues $[27, 15, 13, 9, 9, 8]$ and random eigenvectors, restore its eigenvalues and calculate a set of orthogonal bases for the dominant invariant subspace by the orthogonal iteration.

Table of Contents

LR Iteration

Algorithm 5: LR Iteration

Input: $\mathbf{A} \in \mathbb{C}^{n\times n}$

$\mathbf{A}^{(0)} = \mathbf{A}$;

$k = 0$;

repeat

  $\mathbf{L}^{(k+1)}\mathbf{R}^{(k+1)} = \mathbf{A}^{(k)}$ % perform LU decomp. for $\mathbf{A}^{(k)}$;

  $\mathbf{A}^{(k+1)} = \mathbf{R}^{(k+1)}\mathbf{L}^{(k+1)}$;

  $k := k + 1$;

until *a stopping rule is satisfied*;

Output: $\mathbf{A}^{(k)}$

- Under some mild assumptions, $\mathbf{A}^{(k)}$ converges linearly to an upper-triangular matrix with eigenvalues on its diagonal.

QR Iteration

Algorithm 6: QR Iteration

Input: $\mathbf{A} \in \mathbb{C}^{n\times n}$

$\mathbf{A}^{(0)} = \mathbf{A}$;

$k = 0$;

repeat

$\quad \mathbf{Q}^{(k+1)}\mathbf{R}^{(k+1)} = \mathbf{A}^{(k)}$ % perform QR decomp. for $\mathbf{A}^{(k)}$;

$\quad \mathbf{A}^{(k+1)} = \mathbf{R}^{(k+1)}\mathbf{Q}^{(k+1)}$;

$\quad k := k + 1$;

until *a stopping rule is satisfied*;

Output: $\mathbf{A}^{(k)}$

Denote the Schur decomposition of $\mathbf{A}$ by $\mathbf{A} = \mathbf{U}\mathbf{T}\mathbf{U}^H$; Under some mild assumptions, $\mathbf{A}^{(k)}$ converges linearly to $\mathbf{T}$

- To compute all the eigenvalues of $\mathbf{A}$, picking the diagonal elements of $\mathbf{A}^{(k)}$ for a sufficiently large k would do.
- For $\mathbf{A} \in \mathbb{C}^{n\times n}$, each iteration requires $\mathcal{O}\left(n^3\right)$ to compute the QR decomposition and the matrix multiplication.

QR Iteration

- How to find the eigenvectors?
  - for λ_i, solve the eigen-equation $(\mathbf{T} - \lambda_i \mathbf{I})\,\mathbf{v} = \mathbf{0}$, which is an upper-triangular linear system.
  - for λ_i, use the inverse iteration.

Example (Demo)

- $\mathbf{A}^{(k)}$ converges linearly to $\mathbf{T}$ (upper-triangular).

QR Iteration with Shift

Algorithm 7: QR Iteration With Shift

Input: $\mathbf{A} \in \mathbb{C}^{n\times n}$

$\mathbf{A}^{(0)} = \mathbf{A}$;

$k = 0$;

repeat

  choose a shift $\mu^{(k)}$;

  $\mathbf{Q}^{(k+1)}\mathbf{R}^{(k+1)} = \mathbf{A}^{(k)} - \mu^{(k)}\mathbf{I}$ % perform QR for $\mathbf{A}^{(k)} - \mu^{(k)}\mathbf{I}$;

  $\mathbf{A}^{(k+1)} = \mathbf{R}^{(k+1)}\mathbf{Q}^{(k+1)} + \mu^{(k)}\mathbf{I}$;

  $k := k + 1$;

until *a stopping rule is satisfied*;

Output: $\mathbf{A}^{(k)}$

- shift $\mu^{(k)}$ may differ from iteration to iteration.

QR Iteration with Shift

- Rayleigh quotient shift
  - $\mu^{(k)} = \mathbf{A}^{(k)}(n,n)$ which will converge to the smallest eigenvalue in modulus
  - No guarantee on convergence
  - If converged, at least quadratic convergence
- Wilkinson shift
  - Denote the lower-rightmost 2×2 matrix of $\mathbf{A}^{(k)}$ by

$$\bar{\mathbf{A}}^{(k)} = \begin{bmatrix} \mathbf{A}^{(k)}(n-1,n-1) & \mathbf{A}^{(k)}(n-1,n) \\ \mathbf{A}^{(k)}(n,n-1) & \mathbf{A}^{(k)}(n,n) \end{bmatrix}$$

  - Chose the eigenvalue of $\bar{\mathbf{A}}^{(k)}$ closer to $\mathbf{A}^{(k)}(n,n)$
  - Always converge with at least linear convergence

Example (Demo)

- $\mathbf{A}^{(k)}$ converges linearly to $\mathbf{T}$ (upper-triangular).

Hessenberg Matrix

Any $\mathbf{A} \in \mathbb{C}^{n \times n}$ is unitarily similar to an upper Hessenberg matrix H (i.e., introducing zeros below the first subdiagonal), i.e., $\mathbf{Q}^H \mathbf{A} \mathbf{Q} = \mathbf{H}$.

Hessenberg Reduction

Algorithm 8: Householder Reduction to Upper-Hessenberg Form

Input: $\mathbf{A} \in \mathbb{C}^{n \times n}$

for $k = 1 : n-2$ **do**

$\quad \mathbf{x} = \mathbf{A}(k+1:n, k)$;

$\quad \mathbf{v}_k = \mathbf{x} + \mathsf{sign}(x_1)\|\mathbf{x}\|_2 \mathbf{e}_1$;

$\quad \mathbf{v}_k = \mathbf{v}_k / \|\mathbf{v}_k\|_2$;

$\quad \mathbf{A}(k+1:n, k:n) = \mathbf{A}(k+1:n, k:n) - 2\mathbf{v}_k(\mathbf{v}_k^H \mathbf{A}(k+1:n, k:n))$;

$\quad \mathbf{A}(1:n, k+1:n) = \mathbf{A}(1:n, k+1:n) - 2(\mathbf{A}(1:n, k+1:n)\mathbf{v}_k)\mathbf{v}_k^H$;

Output: $\mathbf{A}$

- Complexity: $\frac{10}{3}n^3 + \mathcal{O}\left(n^2\right)$.

Hessenberg QR Iteration

- Using Givens rotations $\mathbf{G}_1, \mathbf{G}_2, \mathbf{G}_3$ to compute $\mathbf{A}^{(k+1)} = \left(\mathbf{Q}^{(k+1)}\right)^H \mathbf{A}^{(k)} \mathbf{Q}^{(k+1)}$.
- $\mathbf{A}^{(k)}$ preserves the upper-Hessenberg property over iterations.

$$
\mathbf{A}^{(k)} = \begin{bmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ & \times & \times & \times \\ & & \times & \times \end{bmatrix} \to \underbrace{\begin{bmatrix} \times & \times & \times & \times \\ & \times & \times & \times \\ & \times & \times & \times \\ & & \times & \times \end{bmatrix}}_{\mathbf{G}_1^H \mathbf{A}} \to \underbrace{\begin{bmatrix} \times & \times & \times & \times \\ & \times & \times & \times \\ & & \times & \times \\ & & \times & \times \end{bmatrix}}_{\mathbf{G}_2^H \mathbf{G}_1^H \mathbf{A}} \to \underbrace{\begin{bmatrix} \times & \times & \times & \times \\ & \times & \times & \times \\ & & \times & \times \\ & & & \times \end{bmatrix}}_{\mathbf{G}_3^H \mathbf{G}_2^H \mathbf{G}_1^H \mathbf{A} = \mathbf{R}^{(k+1)}}
$$

$$
\mathbf{R}^{(k+1)} = \begin{bmatrix} \times & \times & \times & \times \\ & \times & \times & \times \\ & & \times & \times \\ & & & \times \end{bmatrix} \to \underbrace{\begin{bmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ & & \times & \times \\ & & & \times \end{bmatrix}}_{\mathbf{R}^{(k+1)} \mathbf{G}_1} \to \underbrace{\begin{bmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ & \times & \times & \times \\ & & & \times \end{bmatrix}}_{\mathbf{R}^{(k+1)} \mathbf{G}_1 \mathbf{G}_2} \to \underbrace{\begin{bmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ & \times & \times & \times \\ & & \times & \times \end{bmatrix}}_{\mathbf{R}^{(k+1)} \mathbf{G}_1 \mathbf{G}_2 \mathbf{G}_3 = \mathbf{A}^{(k+1)}}
$$

Hessenberg QR Iteration

Algorithm 9: Hessenberg QR Iteration With Shift

Input: $\mathbf{A} \in \mathbb{C}^{n \times n}$

$\mathbf{H} = \mathbf{Q}^H \mathbf{A}^{(0)} \mathbf{Q}$, $\mathbf{A}^{(0)} = \mathbf{H}$;

$k = 0$;

repeat

  choose a shift $\mu^{(k)}$;

  $\mathbf{Q}^{(k+1)} \mathbf{R}^{(k+1)} = \mathbf{A}^{(k)} - \mu^{(k)} \mathbf{I}$;

  $\mathbf{A}^{(k+1)} = \mathbf{R}^{(k+1)} \mathbf{Q}^{(k+1)} + \mu^{(k)} \mathbf{I}$;

  $k := k + 1$;

until *a stopping rule is satisfied*;

Output: $\mathbf{A}^{(k)}$

- QR decomposition step for $\mathbf{A}^{(k)}$ and the matrix multiplication step requires $\mathcal{O}\left(n^2\right)$.

Exercise (Optional)

Exercise 6. Compute the Hessenberg reduction for the following matrix,

$$\mathbf{A} = \begin{bmatrix} 4 & 1 & 6 & 3 \\ 2 & 5 & 8 & 7 \\ 1 & 1 & 6 & 2 \\ 9 & 3 & 2 & 1 \end{bmatrix}.$$

Table of Contents

Algebraic Ricatti Equation (ARE)

- The quadratic matrix problem,

$$\mathbf{G} + \mathbf{X}\mathbf{A} + \mathbf{A}^\top\mathbf{X} - \mathbf{X}\mathbf{F}\mathbf{X} = \mathbf{0},$$

  arises in optimal control and a symmetric solution is sought so that the eigenvalues of $\mathbf{A} - \mathbf{F}\mathbf{X}$ are in the open left half plane.
- The Hamiltonian Matrix,

$$\mathbf{M} = \begin{bmatrix} \mathbf{A} & \mathbf{G} \\ \mathbf{F} & -\mathbf{A}^\top \end{bmatrix},$$

  together with the assumption that $\mathbf{M}$ has no purely imaginary eigenvalues, it is possible to show that an orthogonal symplectic matrix $\mathbf{Q}$ exists so that

$$\mathbf{Q}^\top\mathbf{M}\mathbf{Q} = \begin{bmatrix} \mathbf{Q}_1 & \mathbf{Q}_2 \\ -\mathbf{Q}_2 & \mathbf{Q}_1 \end{bmatrix}^\top \begin{bmatrix} \mathbf{A} & \mathbf{F} \\ \mathbf{G} & -\mathbf{A}^\top \end{bmatrix} \begin{bmatrix} \mathbf{Q}_1 & \mathbf{Q}_2 \\ -\mathbf{Q}_2 & \mathbf{Q}_1 \end{bmatrix} = \begin{bmatrix} \mathbf{T} & \mathbf{R} \\ 0 & -\mathbf{T}^\top \end{bmatrix}.$$

- It is easy to show that $\mathbf{A} - \mathbf{F}\mathbf{X} = \mathbf{Q}_1\mathbf{T}\mathbf{Q}_1^{-1}$ and so the eigenvalues of $\mathbf{A} - \mathbf{F}\mathbf{X}$ are the eigenvalues of $\mathbf{T}$.
- $\mathbf{X} = \mathbf{Q}_2\mathbf{Q}_1^{-1}$ is a symmetric solution.

see Ch7.8, Ch 1.3.10 [Golub-van-Loan'13] and the reference pdf file in the folder for details.

Exercise (Optional)

Exercise 7. Consider the following ARE problem,

$$\mathbf{A}^\top \mathbf{X} + \mathbf{X}\mathbf{A} - \mathbf{X}\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^\top \mathbf{X} + \mathbf{Q} = \mathbf{0},$$

in which

$$\mathbf{A} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}, \quad B = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad \mathbf{Q} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \mathbf{R} = 1.$$

- Find the symmetric solution $\mathbf{X}$ by computing the real Hamiltonian-Schur decomposition and ordering the eigenvalues so that $\lambda(\mathbf{T})$ is in the left half plane.
- Calculate the residual to determine whether the solution is accurate enough.