Plan général

Partie 1: Généralités

Partie 2: Méthodes directes

Partie 3: Méthodes itératives

Partie 4: Problèmes aux valeurs et vecteurs propres

Partie 5: Systèmes linéaires mal conditionnés

Motivations for eigenvalue / eigenvector computation

Mechanics and physics: Eigenvalues/eigenvectors useful for a very diverse array of reasons.

- Stability in time of mechanical and physical systems determined by eigenvalue problems. Likewise, eigenvalues reveal potential resonances.
- Eigenvectors often used (e.g. in structural dynamics) for low-dimensional approximation of dynamical responses.

Example: perturbation of dynamical system about equilibrium solution x_0 ($x(t)$, $x_0 \in \mathbb{R}^n$)

$$\dot{x} = \mathcal{F}(x), \qquad x(t) = x_0 + y(t),\ \mathcal{F}(x_0) = 0 \implies \dot{y} = \mathcal{F}'(x_0)y + o(\|y\|)$$

Try $y(t) = Ye^{\lambda t}$ on linearized model, then $\lambda Y = \mathcal{F}'(x_0)Y$, stability if $\mathrm{Re}(\lambda) < 0$.

Statistics: eigenvalues/eigenvectors of covariance matrices

Computation and algorithms:

- Computing matrix SVDs ($A = USV^{\mathrm{H}}$) requires eigenvalues/eigenvectors. Then, singular values/vectors quantify information content of linear system.
- Knowing eigenvalue properties essential in *preconditioning* of iterative solution methods (lecture 4).

This lecture:

- A few major ideas and methods for eigenvalue/eigenvector computation;
- Focus on symmetric (Hermitian) eigenvalue problem;
- First, methods for isolated eigenvalues
- Then, methods for complete matrix spectra

Summary of facts about eigenvalues

For any $A \in \mathbb{K}^{n\times n}$, there exist n eigenvalues $\lambda_1, \ldots, \lambda_n \in \mathbb{C}$ and eigenvectors $x_1, \ldots, x_n$ such that

$$\boxed{Ax_i = \lambda_i x_i \qquad 1 \le i \le n.}$$

Eigenvalues: the n roots of the n-th degree *characteristic polynomial* $p_A(\lambda) := \det(A - \lambda I)$; can have multiplicities.

- If $\mathbb{K}^n = \text{span}(x_1, \ldots, x_n)$, A is *diagonalizable*: $A = X\Lambda X^{-1}$ ($\Lambda := \text{diag}(\lambda_1, \ldots, \lambda_n)$, $X := [x_1, \ldots, x_n]$).
- If $A = Q\Lambda Q^{-1} = Q\Lambda Q^{\mathsf{H}}$ with Q unitary, A is *unitarily diagonalizable*.
- Unitary diagonalization of A possible if and only if A *normal* ($AA^{\mathsf{H}} = A^{\mathsf{H}}A$).
- A Hermitian ($A = A^{\mathsf{H}}$) is normal, hence unitarily diagonalizable, and $\lambda_i \in \mathbb{R}$.
  A real symmetric $\implies$ Q orthogonal, $A = Q\Lambda Q^{\mathsf{T}}$.
- A not diagonalizable is called *defective*. Example:

$$A = \begin{bmatrix} a & b \\ 0 & a \end{bmatrix} : \qquad \lambda_1 = \lambda_2 = a, \quad E_\lambda = \text{span}(x_1),\ x_1 = \begin{Bmatrix} 1 \\ 0 \end{Bmatrix}$$

  Algebraic multiplicity $\ge$ geometric multiplicity (here $2 \ge 1$).
- Gershgorin theorem: Each λ_k in one of the disks $\mathcal{D}_i(A) := \left\{ z \in \mathbb{C} : |z - a_{ii}| \le \sum_{j\neq i} |a_{ij}| \right\}$.

Convention: eigenvalues ordered such that $|\lambda_1| \ge |\lambda_2| \ge \ldots |\lambda_n|$.

Impossibility of direct algorithm for eigenvalue computation

- (a) Eigenvalues are roots of (degree-n) characteristic polynomial.
  (b) Any (degree-n) polynomial is characteristic polynomial of some matrix:

$$P(X) = a_0 + a_1 X + \ldots + a_{n-1} X^{n-1} + X^n$$

$$A_P = \begin{bmatrix} 0 & 0 & \ldots & 0 & -a_0 \\ 1 & 0 & \ldots & 0 & -a_1 \\ 0 & 1 & \ldots & 0 & -a_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \ldots & 1 & -a_{n-1} \end{bmatrix}. \qquad \textit{companion matrix of } P$$

  Therefore eigenvalue problems and polynomial root-finding problems are **equivalent**
- No direct general root-finding method if $n \geq 5$ (impossibility result, Galois)
- Therefore **no direct method for general eigenvalue problems**

Any general method for computing matrix eigenvalues must be iterative.

This lecture:

- Computation of isolated eigenvalues
- Computation of matrix spectra

Computation of isolated eigenvalues: Rayleigh quotient

Computation of isolated eigenvalues: main ingredients are

- Power iterations
- Rayleigh quotients
- Matrix shifts

Let $A \in \mathbb{K}^{n\times n}$ Hermitian, $x \in \mathbb{K}^n$. Rayleigh quotient $r(x)$: $r(x) := \dfrac{x^{\mathsf{H}} A x}{x^{\mathsf{H}} x}$.

We have $\boxed{\nabla_x r(x) = \dfrac{2}{x^{\mathsf{H}} x}\left(Ax - r(x)x\right)}$.

- Rayleigh quotient is stationary ($\nabla_x r(x) = 0$) if $x, r(x)$ eigenvector/eigenvalue pair.
- The smallest (largest) eigenvalue of A minimizes (maximizes) $r(x)$ over $x \in \mathbb{K}^n \setminus \{0\}$.
- Infinite-dimensional counterpart: Courant-Fischer min-max principle (ENSTA ANA 202)
- Dynamics of mechanical systems: $r(x)$ ratio of strain and kinetic energies.

Computation of isolated eigenvalues: power iterations

Repeated evaluations $x^{(0)} \mapsto Ax^{(0)} \mapsto A^2x^{(0)} \mapsto \ldots$, compute Rayleigh quotients along the way:

Algorithm 9 Power iteration

$A \in \mathbb{K}^{n\times n}$ Hermitian (input), $x^{(0)} \in \mathbb{K}^n$ with $\|x^{(0)}\| = 1$ (initialization)
for $k = 0, 1, 2, \ldots$ **do**
  $v = Ax^{(k)}$ (apply A to current normalized iterate)
  $\lambda^{(k)} = v^{\mathrm{H}}x^{(k)}$ (Rayleigh quotient)
  $x^{(k+1)} = v/\|v\|$ (next normalized iterate)
  Stop if convergence, set $\lambda_1 = \lambda^{(k)}$, $q_1 = x^{(k)}$
end for

Power iterations promote λ_1, q_1 (assuming $|\lambda_2| > |\lambda_1|$): use $A = Q\Lambda Q^{\mathrm{H}}$, expand on eigenvectors:

$$x^{(0)} = y_1 q_1 + y_2 q_2 + \cdots + y_n q_n$$
$$x^{(k)} = c_k A^k x^{(0)}$$
$$= c_k \lambda_1^k \left[y_1 q_1 + y_2 (\lambda_2/\lambda_1)^k q_2 + \cdots + y_n (\lambda_n/\lambda_1)^k q_n \right]$$

Convergence of power iterations

Assume $q_1^{\mathrm{H}} x^{(0)} \neq 0$ and $|\lambda_1| > |\lambda_2| \geq \ldots \geq |\lambda_n|$. Then:
$$|\lambda^{(k)} - \lambda_1| = O(|\lambda_2/\lambda_1|^{2k}), \qquad \| \pm x^{(k)} - q_1\| = O(|\lambda_2/\lambda_1|^k) \quad (k \to \infty)$$

Computation of isolated eigenvalues: power iterations (properties, limitations)

$$|\lambda^{(k)} - \lambda_1| = O(|\lambda_2/\lambda_1|^{2k}), \qquad \| \pm x^{(k)} - q_1\| = O(|\lambda_2/\lambda_1|^{k}) \quad (k \to \infty)$$

- Convergence of λ^k to λ_1 linear (error reduced by constant factor at each iteration);
- Convergence of $x^{(k)}$ to q_1 also linear.
- Error reduction depends on closeness of $|\lambda_1|$ and $|\lambda_2|$.
- "raw" power iteration, evaluate λ_1, q_1 only.
- Computational work: one matrix-vector product per power iteration.

Computation of isolated eigenvalues: inverse iterations

Idea: if A invertible, can apply power iterations to A^{-1}.

- Iterates: $x^{(k+1)} = A^{-1}x^{(k)}$ (i.e. solve $Ax^{(k+1)} = x^{(k)}$)
- Expected to produce λ_n, q_n (smallest eigenvalue)

Extension: apply power iterations to shifted matrix $(A-\mu I)^{-1}$, μ close to some λ_j:

- $\sigma_j := (\lambda_j - \mu)^{-1}$ an eigenvalue of $(A-\mu I)^{-1}$ with same eigenvector q_j; moreover

$$\boxed{|\mu - \lambda_j| < |\mu - \lambda_i| \ (i \neq j) \qquad \Longrightarrow \ |\sigma_j| > |\sigma_i| \ (i \neq j)}$$

- Power iterations on $(A-\mu I)^{-1} \implies (\sigma_j, q_j) \implies \lambda_j = \sigma_j^{-1} + \mu$
  Finds eigenvalue closest to μ

Algorithm 10 Inverse iteration

$A \in \mathbb{K}^{n\times n}$ Hermitian (input), $x^{(0)} \in \mathbb{K}^n$ with $\|x^{(0)}\| = 1$ (initialization), $\mu \in \mathbb{R}$ close to λ_j
for $k = 1, 2, \ldots$ **do**
  solve $(A-\mu I)x^{(k)} = x^{(k-1)}$ (apply $(A-\mu I)^{-1}$ to $x^{(k-1)}$)
  $x^{(k)} = x^{(k)}/\|x^{(k)}\|$ (next normalized iterate)
  $\lambda_j^{(k)} = (x^{(k)})^{\mathsf{H}} A x^{(k)}$ (Rayleigh quotient)
  Stop if convergence, set $\lambda_j = \lambda_j^{(k)}$, $q = x^{(k)}$
end for

$$|\lambda_j^{(k)} - \lambda_j| = O\Big(\frac{|\lambda_j - \mu|}{|\lambda_\ell - \mu|}\Big)^{2k}, \qquad \|\pm x_k - q_j\| = O\Big(\frac{|\lambda_j - \mu|}{|\lambda_\ell - \mu|}\Big)^{k}$$

(similarly to power iterations)

Computation of isolated eigenvalues: Rayleigh quotient iterations

- Power iterations yield eigenvectors
- Inverse iterations yield eigenvalues
- Combine both: Rayleigh quotient iterations (set shift μ to current eigenvalue estimate)

Algorithm 11 Rayleigh quotient iteration

$A \in \mathbb{K}^{n\times n}$ Hermitian (input), $x^{(0)} \in \mathbb{K}^n$ with $\|x_0\| = 1$ (initialization)
$\lambda^{(0)} = (x^{(0)})^H A x^{(0)}$ (Initialize Rayleigh quotient)
for $k = 1, 2, \ldots$ **do**
  solve $(A - \lambda^{(k-1)} I)x^{(k)} = x^{(k-1)}$ (apply $(A - \lambda^{(k-1)} I)^{-1}$ to $x^{(k-1)}$)
  $x^{(k)} = x^{(k)}/\|x^{(k)}\|$ (next normalized iterate)
  $\lambda^{(k)} = (x^{(k)})^H A x^{(k)}$ (Rayleigh quotient)
  Stop if convergence, set $\lambda = \lambda^{(k)}$, $q = x^{(k)}$
end for

Convergence of Rayleigh quotient iterations

Rayleigh iterations converge for almost all starting vectors $x^{(0)}$. Assume (normalized) $x^{(0)}$ close to eigenvector q_j. Then:

$$|\lambda^{(k+1)} - \lambda_j| = O(|\lambda^{(k)} - \lambda_j|^3), \qquad \| \pm x^{(k+1)} - q_j\| = O(\| \pm x^{(k)} - q_j\|^3) \quad (k \to \infty).$$

Computation of isolated eigenvalues: example

$A \in \mathbb{R}^{10\times 10}$ a random real symmetric matrix.

1. Power iterations for $\lambda_1 \approx 10.681$. Tolerance: $|\lambda^{(k+1)} - \lambda^{(k)}| \leq 10^{-10}$

k	$\lvert\lambda^{(k)} - \lambda_1\rvert$	$(\lambda_2/\lambda_1)^{2k}$	$\lVert q^{(k)} - q_1\rVert$	$(\lambda_2/\lambda_1)^{k}$
1	2.5383e+00	3.9667e-02	5.1770e-01	1.9916e-01
2	4.4221e-02	1.5734e-03	6.6508e-02	3.9667e-02
3	9.2950e-04	6.2413e-05	9.6536e-03	7.9002e-03
4	2.5329e-05	2.4757e-06	1.5871e-03	1.5734e-03
5	7.8583e-07	9.8202e-08	2.7731e-04	3.1337e-04
6	2.5948e-08	3.8953e-09	4.9888e-05	6.2413e-05
7	8.8738e-10	1.5451e-10	9.1258e-06	1.2430e-05
8	3.1084e-11	6.1290e-12	1.6892e-06	2.4757e-06
9	1.1084e-12	2.4312e-13	3.1567e-07	4.9307e-07

Computation of isolated eigenvalues: example

1. Inverse iterations ($(A-\mu I)^{-1}$ with $\mu = 0.4$ close to $\lambda_5 \approx 0.46740$).

k	$\lvert\lambda^{(k)}-\lambda_5\rvert$	$\left\lvert\frac{\lambda_5-\mu}{\lambda_4-\mu}\right\rvert^{2k}$	$\lVert q^{(k)}-q_5\rVert$	$\left\lvert\frac{\lambda_5-\mu}{\lambda_4-\mu}\right\rvert^{k}$
1	3.2395e-01	9.9940e-02	1.0111e+00	3.1613e-01
2	8.1309e-03	9.9880e-03	1.3554e-01	9.9940e-02
3	3.8992e-05	9.9819e-04	1.3551e-02	3.1594e-02
4	6.6151e-07	9.9759e-05	2.7233e-03	9.9880e-03
5	9.4566e-08	9.9699e-06	8.1688e-04	3.1575e-03
6	9.6321e-09	9.9639e-07	2.5726e-04	9.9819e-04
7	9.6379e-10	9.9579e-08	8.1308e-05	3.1556e-04
8	9.6329e-11	9.9519e-09	2.5704e-05	9.9759e-05
9	9.6276e-12	9.9459e-10	8.1257e-06	3.1537e-05

2. Rayleigh iterations for $\lambda_5 \approx 0.46740$.

k	$\lvert\lambda^{(k)}-\lambda_5\rvert$
1	6.6642e-03
2	7.0746e-07
3	4.9960e-16

Extension: generalized symmetric eigenvalue problems

- Many engineering applications (e.g. vibrations, forced dynamical motions) involve

  Find λ, x such that $\quad Kx = \lambda Mx.$

  $K, M \in \mathbb{R}^{n\times n}$: (symmetric, positive) stiffness and mass matrices.
- Usually, lower part of spectrum (for which FE model most accurate) sought
- Assume M SPD (true for dynamics/vibrations), set $M = GG^{\mathsf{T}}$ (Cholesky); then:

  $$Kx - \lambda Mx = G\big(A - \lambda I\big)G^{\mathsf{T}}x, \qquad A = G^{-1}KG^{-\mathsf{T}}$$
- Equivalent symmetric eigenvalue problem:

  Find λ, x such that $\quad Ay = \lambda y, \quad G^{\mathsf{T}}x = y.$

  Do not actually evaluate $A = G^{-1}KG^{-\mathsf{T}}$; reinterpret algorithms for standard eigenvalue pbs.

Algorithm 12 Inverse iteration for structural vibrations

$K, M \in \mathbb{R}^{n\times n}$ (stiffness / mass matrices), $x^{(0)} \in \mathbb{R}^n$, $\|x^{(0)}\| = 1$ (initialization), $\mu \in \mathbb{R}$ close to λ_j
for $k = 1, 2, \ldots$ **do**
  solve $(K - \mu M)x^{(k)} = Mx^{(k-1)}$ (solve forced vibration problem with $Mx^{(k-1)}$ as load)
  $x^{(k)} = x^{(k)}/\sqrt{x^{(k)\mathsf{T}}Mx^{(k)}}$ (next mass-normalized iterate)
  $\lambda_j^{(k)} = (x^{(k)})^{\mathsf{H}}Kx^{(k)}$ (Rayleigh quotient)
  Stop if convergence, set $\lambda_j = \lambda_j^{(k)}$, $q = x^{(k)}$
end for

Plan général, organisation

Partie 1: Généralités
Partie 2: Méthodes directes
Partie 3: Méthodes itératives
Partie 4: Problèmes aux valeurs et vecteurs propres
- Séance 5a: Généralités, Puissances itérées, puissances inverses
- Séance 6a: **Itérations orthogonales et algorithme QR**

Partie 5: Systèmes linéaires mal conditionnés

Computation of matrix spectra

General approach: use invariance of eigenvalues under similarity transformations:

- Let $A \in \mathbb{K}^{n\times n}$, let $X \in \mathbb{K}^{n\times n}$ invertible, set $T := X^{-1}AX$.
  Then $P_A(\lambda) = P_T(\lambda)$ (same eigenvalues and multiplicities).

Matrix spectra computations: find similarity decomposition $A = XTX^{-1}$ where eigenvalues of T "easy" to compute.

- Factorizations for direct methods (e.g. LU, Cholesky) not appropriate
  For instance: $A = LU$ reveals eigenvalues of L, U, but no connection to eigenvalues of A.
- In fact, we know LU etc cannot work (since direct eigenvalue algorithms inpossible)

Better starting point: the Schur decomposition of A:

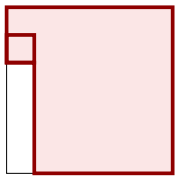
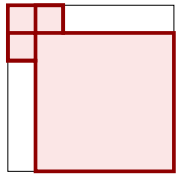
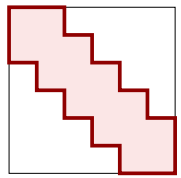
- Any $A \in \mathbb{K}^{n\times n}$ has a *Schur decomposition* $A = QTQ^{\mathrm{H}}$ ($Q \in \mathbb{K}^{n\times n}$ unitary, $T \in \mathbb{K}^{n\times n}$ upper triangular)
- Schur decomposition is a similarity transformation of A (so $P_A(\lambda) = P_T(\lambda)$)
- Since T triangular, $\mathrm{diag}(T)$ holds the eigenvalues of A.
- Even if $A \in \mathbb{R}^{n\times n}$, $Q, T \in \mathbb{C}^{n\times n}$ in general (as real matrices may have complex eigenvalues).
- If A is Hermitian, T is real and diagonal.

Computation of matrix spectra: a possible outline

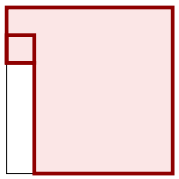
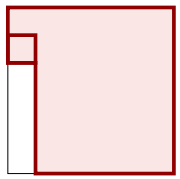
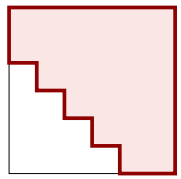
Ideal general approach: compute eigenvalues by finding Schur decomposition of A.

Towards finding $A = QTQ^{\mathrm{H}}$: introduce zeros in lower triangle of A (again!), but note carefully:

- Let F unitary; assume FA puts zeros in whole 1st column under diagonal.
  Then, similarity needs forming FAF^{H}, but right multiplication undoes zeroing-out (try with Householder reflector)
- Remedy: use instead (e.g. Householder) transformations such that (for Hermitian A)

$F_1 A =$  $F_1 A F_1^{\mathrm{H}} =$  $\ldots FAF^{\mathrm{H}} =$ 

  A Hermitian $\Longrightarrow$ FAF^{H} tridiagonal
- For non-Hermitian A, can reach FAF^{H} upper Hessenberg:

$F_1 A =$  $F_1 A F_1^{\mathrm{H}} =$  $\ldots FAF^{\mathrm{H}} =$ 

- This is not yet the Schur decomposition (but we get closer)
- Reduction to tridiagonal/Hessenberg takes fixed computational work (direct step)
  Then, the rest (e.g. tridiagonal/Hessenberg to Schur) is iterative
- Finding $A = QTQ^{\mathrm{H}}$ may need complex arithmetic even if A real (but then we prefer real arithmetic).

Computation of matrix spectra: orthogonal iterations

- Assume $|\lambda_1| > |\lambda_2| > \dots |\lambda_n|$, with corresponding eigenvectors $Q_1, \dots, q_n$.
- Starting idea: apply power iterations to a set of p vectors $X_p = [x_1, \dots, x_p] \in \mathbb{K}^{n\times p}$:

  $$X_p^{(0)} = X,\ X_p^{(1)} = AX_p^{(0)}, \dots\ X_p^{(k)} = AX_p^{(k-1)} \dots$$

  Then $E_p^{(k)} := \text{span}\big(x_1^{(k)}, \dots, x_p^{(k)}\big) \ \to\ E_p := \text{span}\big(q_1, \dots, q_p\big)$
- Conceivably: (a) run k iterations (until convergence of $\text{span}\big(x_1^{(k)}, \dots, x_p^{(k)}\big)$),
  (b) diagonalize smaller matrix $A_p^{(k)} := (X^{(k)})^{\mathsf{H}} A X^{(k)} \in \mathbb{K}^{p\times p}$.
- However, vectors of $X_p^{(k)}$ increasingly collinear
- Remedy: orthogonalization (again!), i.e. find next iterate $X_p^{(k)}$ via

  $$X_p^{(k)} R^{(k)} = AX_p^{(k-1)}$$ use QR decomposition on $AX_p^{(k-1)}$

Algorithm 13 Orthogonal iterations

1: $A \in \mathbb{K}^{n\times n}$ Hermitian, $X_p^{(0)} = [x_1^{(0)}, \dots, x_p^{(0)}] \in \mathbb{K}^{n\times p}$ with orthonormal columns (initialization)
2: **for** $k = 1, 2, \dots$ **do**
3: $\quad Z^{(k)} = AX_p^{(k-1)}$ (apply A)
4: $\quad X_p^{(k)} R_p^{(k)} = Z^{(k)}$ (compute reduced QR decomposition of $Z^{(k)} \in \mathbb{K}^{n\times p}$)
5: $\quad$ **Stop** if convergence, set $\lambda_i = (x_i^{(k)})^{\mathsf{H}} A q_i^{(k)}$, $q_i = x_i^{(k)}$
6: **end for**

How and why orthogonal iterations work

Focus on case $p = 2$ (recall $p = 1$ is standard power iteration):

$$AX_p^{(k-1)} = X_p^{(k)} R^{(k)} \qquad \text{with} \quad R^{(k)} = \begin{bmatrix} r_{11}^{(k)} & r_{12}^{(k)} \\ 0 & r_{22}^{(k)} \end{bmatrix} \qquad \text{i.e.} \quad \begin{cases} \text{(a)} & r_{11}^{(k)} x_1^{(k)} = A x_1^{(k-1)}, \\ \text{(b)} & r_{12}^{(k)} x_1^{(k)} + r_{22}^{(k)} x_2^{(k)} = A x_2^{(k-1)} \end{cases}$$

(a) $x_1^{(0)}, x_1^{(1)}, x_1^{(2)} \dots x_1^{(k)} \dots$ generated by power iterations, hence $x_1^{(k)} \to q_1$.

(b) Write $x_1^{(k)} = q_1 + \varepsilon_1^{(k)}$ with $\|\varepsilon_1^{(k)}\| \to 0$, then set

$$\begin{aligned} \widehat{A} &= (I - q_1 q_1^{\mathsf{H}})^{\mathsf{H}} A (I - q_1 q_1^{\mathsf{H}}) \\ &= A - \lambda_1 q_1 q_1^{\mathsf{H}} \qquad \Longrightarrow \ \widehat{A} q_1 = 0, \ \widehat{A} q_i = A q_i \ (i \geq 2) \end{aligned}$$

Consequently:

$$\begin{aligned} \widehat{A} x_2^{(k-1)} &= A x_2^{(k-1)} - \lambda_1 (q_1^{\mathsf{H}} x_2^{(k-1)}) q_1 \\ &= r_{12}^{(k)} x_1^{(k)} + r_{22}^{(k)} x_2^{(k)} - \lambda_1 (q_1^{\mathsf{H}} x_2^{(k-1)}) q_1 \end{aligned} \qquad \begin{aligned} r_{12}^{(k)} &= (x_1^{(k)})^{\mathsf{H}} A x_2^{(k-1)} \\ &= (q_1 + \varepsilon_1^{(k)})^{\mathsf{H}} A x_2^{(k-1)} \\ &= \lambda_1 (q_1^{\mathsf{H}} x_2^{(k-1)}) + (\varepsilon_1^{(k)})^{\mathsf{H}} A x_2^{(k-1)} \end{aligned}$$

$$\begin{aligned} \widehat{A} x_2^{(k-1)} &= r_{22}^{(k)} x_2^{(k)} + (q_1^{\mathsf{H}} x_2^{(k-1)}) \varepsilon_1^{(k)} + ((\varepsilon_1^{(k)})^{\mathsf{H}} A x_2^{(k-1)}) x_1^{(k)} \\ &= r_{22}^{(k)} x_2^{(k)} + (\varepsilon_1^{(k)}) \end{aligned}$$

$x_2^{(0)}, x_2^{(1)}, x_2^{(2)} \dots x_2^{(k)} \dots$ generated by power iterations for $\widehat{A}$.

Convergence of orthogonal iterations

Let $A \in \mathbb{K}^{n\times n}$ Hermitian with $|\lambda_1| > |\lambda_2| > \dots |\lambda_p|$.

Assume all leading submatrices $(Q_p^{\mathsf{H}} X_p^{(0)})_{1:q,1:q}$ $(1 \le q \le p)$ of $Q_p^{\mathsf{H}} X_p^{(0)} \in \mathbb{K}^{p\times p}$ are nonsingular.

Let $X_p^{(k)} = [x_1^{(0)}, \dots, x_p^{(0)}]$: set of orthonormal vectors produced by k orthogonal iterations. Then:

$$\|x_i^{(k)} \pm q_i\| = O(C^k), \qquad \text{with} \quad C := \max_{1\le j\le p} |\lambda_{j+1}|/|\lambda_j| < 1.$$

Computation of matrix spectra: QR iterations

- Adapt orthogonal iterations to complete spectrum of $A \in \mathbb{K}^{n\times n}$ (Hermitian);
- Remove restriction $|\lambda_1| > |\lambda_2| > \ldots |\lambda_p| > \ldots$

Focus on $T^{(k)} := X^{(k)\mathsf{H}} A X^{(k)}$ (note $T^{(k)} \to \mathrm{diag}(\lambda_1, \ldots, \lambda_n)$):

$$
\begin{aligned}
T^{(k-1)} &= X^{(k-1)\mathsf{H}} A X^{(k-1)} = X^{(k-1)\mathsf{H}} X^{(k)} R^{(k)} \qquad \text{(QR decomposition of } AX^{(k-1)}\text{)} \\
T^{(k)} &= X^{(k)\mathsf{H}} A X^{(k)} = X^{(k)\mathsf{H}} A X^{(k-1)} X^{(k-1)\mathsf{H}} X^{(k)} = X^{(k)\mathsf{H}} X^{(k)} R^{(k)} X^{(k-1)\mathsf{H}} X^{(k)} \\
&= R^{(k)} X^{(k-1)\mathsf{H}} X^{(k)},
\end{aligned}
$$

Reformulate:

$$
\text{(a) } T^{(k-1)} = Q^{(k)} R^{(k)}, \qquad \text{(b) } T^{(k)} = R^{(k)} Q^{(k)}, \qquad Q^{(k)} := X^{(k-1)\mathsf{H}} X^{(k)} \text{ unitary.}
$$

Algorithm 14 Basic QR iterations

1: $A \in \mathbb{K}^{n\times n}$ Hermitian, $T^{(0)} = A$ (initialization)
2: **for** $k = 1, 2, \ldots$ **do**
3: $\quad Q^{(k)} R^{(k)} = T^{(k-1)}$ (compute QR decomposition of $T^{(k-1)} \in \mathbb{K}^{n\times p}$)
4: $\quad T^{(k)} = R^{(k)} Q^{(k)}$ (update $T^{(k)}$)
5: $\quad$ **Stop** if convergence, $\mathrm{diag}(T^{(k)})$ contains the eigenvalues of A
6: **end for**

Computation of matrix spectra: shortcomings of basic QR iterations

Basic QR iterations are workable (in particular backward stable) but lack efficiency:

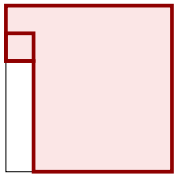
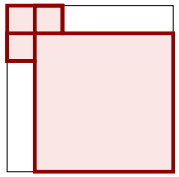
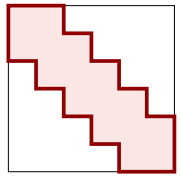
- Each QR factorization costs $O(n^3)$ operations (see lecture 2)
- Expect $O(n)$ QR iterations needed, so $O(n^4)$ computing work overall.
- Rate of convergence of eigenvalues depends on their distribution
- Convergence may fail if $|\lambda_j| = |\lambda_{j+1}|$ for some j.

Two major improvements adresss these issues:

- First put A in *tridiagonal form* (needs $O(n^3)$ work).
  QR decompositions of a tridiagonal matrix then take $O(n^2)$ work $\implies$ $O(n^3)$ overall work.
- Accelerate convergence using a shifted form of QR algorithm

Computation of matrix spectra: reduction to tridiagonal form

Method 1: symmetric application of Householder reflectors

$F_1 A =$  $F_1 A F_1^{\mathrm{H}} =$  $\ldots F A F^{\mathrm{H}} =$ 

- Requires storage of A
- Proved stability

Method 2: Lanczos orthogonalization iterations

- Recall Arnoldi iterations (used for GMRES):

  $\boxed{A = QHQ^{\mathrm{H}}}$ Q unitary, H upper Hessenberg

  Here, A Hermitian $\implies$ H tridiagonal.
- Specialize Arnoldi iterations to A Hermitian (so H tridiagonal) $\rightarrow$ Lanczos iterations
- Only requires matrix-vector products $q \mapsto Aq$, i.e. suitable for large sparse matrices

Lanczos iterations

Column-by-column enforcement of equality (starting with q_1 arbitrary unit vector)

$$A[q_1 \ \dots \ q_k] = [q_1 \ \dots \ q_k, q_{k+1}] \begin{bmatrix} \alpha_1 & \beta_1 & 0 & \dots & 0 \\ \beta_1 & \alpha_2 & & & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & & & \ddots & \beta_{n-1} \\ 0 & \dots & 0 & \beta_{n-1} & \alpha_n \end{bmatrix}$$

column 1: $Aq_1 = \alpha_1 q_1 + \beta_1 q_2 \quad \Longrightarrow \alpha_1, \beta_1, q_2$

$(q_1^{\mathsf{H}} q_2 = 0, \ \|q_2\| = 1)$

column 2: $Aq_2 = \beta_1 q_1 + \alpha_2 q_2 + \beta_2 q_3 \quad \Longrightarrow \alpha_2, \beta_2, q_3$

$(q_1^{\mathsf{H}} q_3 = q_2^{\mathsf{H}} q_3 = 0, \ \|q_3\| = 1)$

column k: $Aq_k = \beta_{k-1} q_{k-1} + \alpha_k q_k + \beta_k q_{k+1} \quad \Longrightarrow \alpha_k, \beta_k, q_{k+1}$

$(q_{k-1}^{\mathsf{H}} q_{k+1} = q_k^{\mathsf{H}} q_{k+1} = 0, \ \|q_{k+1}\| = 1) \dots$

column n: $Aq_n = \beta_{n-k1} q_{n-1} + \alpha_n q_n \quad \Longrightarrow \alpha_n$

By induction: $q_k \in \mathrm{span}(q_1, Aq_1, \dots, A^{k-1}q_1)$ for $k = 1, 2, 3 \dots$

$\mathrm{span}(q_1, q_2, \dots, q_k) = \mathrm{span}(q_1, Aq_1, \dots, A^{k-1}q_1) = \mathcal{K}_k(A, b)$

Inverse-iteration interpretation of the QR algorithm

- Generic orthogonal iteration at root of basic QR algorithm:
  $$X^{(k)}R^{(k)} = AX^{(k-1)\mathrm{H}} \implies A = X^{(k)}R^{(k)}X^{(k-1)\mathrm{H}}.$$
- Evaluate $A^{-1} = (A^{-1})^{\mathrm{H}}$ (since A Hermitian):
  $$A^{-1} = X^{(k-1)}(R^{(k)})^{-1}X^{(k)\mathrm{H}} = X^{(k)}(R^{(k)})^{-\mathrm{H}}X^{(k-1)\mathrm{H}}$$
  Rewrite using "flipped identity" P (properties: $P^2 = I$, $P[x_1, \ldots, x_n] = [x_n, \ldots, x_1]$):
  $$\boxed{A^{-1} = \left(X^{(k)}P\right)\left(P(R^{(k)})^{-\mathrm{H}}P\right)\left(X^{(k-1)}P\right)^{\mathrm{H}},} \qquad P := \begin{bmatrix} 0 & \cdots & & 1 \\ \vdots & & \iddots & \\ & \iddots & & \vdots \\ 1 & & \cdots & 0 \end{bmatrix}$$
- Observe (i) $X^{(k-1)}P$, $X^{(k)}P$ unitary; (ii) $P(R^{(k)})^{-\mathrm{H}}P$ upper triangular.

Orthogonal iteration for A on $X^{(k)}$ equivalent to orthogonal iteration for A^{-1} on $X^{(k)}P$

In particular, 1st column of $X^{(k)}P$, i.e. $x_n^{(k)}$, undergoes inverse iteration (without shift).

Computation of matrix spectra: shifted QR algorithm

Inverse-iteration interpretation suggests using a shift $\mu^{(k)}$; main steps become

$$\text{(a) } T^{(k-1)} - \mu^{(k)} I = Q^{(k)} R^{(k)} \quad \text{and} \quad \text{(b) } T^{(k)} = R^{(k)} Q^{(k)} + \mu^{(k)} I.$$

How to (adaptively) choose shifts?

- $\rightarrow$ $\mu^{(k)} := t_{nn}^{(k-1)}$ natural choice (Rayleigh quotient for $q_n^{(k-1)}$), but known to fail on some "nice" matrices.
- $\rightarrow$ Wilkinson shift (eigenvalue of bottom rightmost 2×2 block of $T^{(k-1)}$ closest to $t_{nn}^{(k-1)}$)

Algorithm 15 Shifted QR iterations

1: $A \in \mathbb{K}^{n \times n}$ Hermitian (data)
2: $(Q^{(0)})^{\mathsf{H}} T^{(0)} Q^{(0)} = A$ (Tridiagonalization of A)
3: **for** $k = 1, 2, \ldots$ **do**
4: Choose $\mu^{(k)}$ (shift value, e.g. use the *Wilkinson shift*)
5: $Q^{(k)} R^{(k)} = T^{(k-1)} - \mu^{(k)} I$ (compute QR factorization of $T^{(k-1)} - \mu^{(k)} I \in \mathbb{K}^{n \times p}$)
6: $T^{(k)} = R^{(k)} Q^{(k)} + \mu^{(k)} I$ (update $T^{(k)}$)
7: If any off-diagonal entry $t_{j,j+1}^{(k)}$ is sufficiently small,

set $t_{j,j+1}^{(k)} = t_{j+1,j}^{(k)} = 0$ to obtain $T^{(k)} = \begin{bmatrix} T_1^{(k)} & 0 \\ 0 & T_2^{(k)} \end{bmatrix}$.

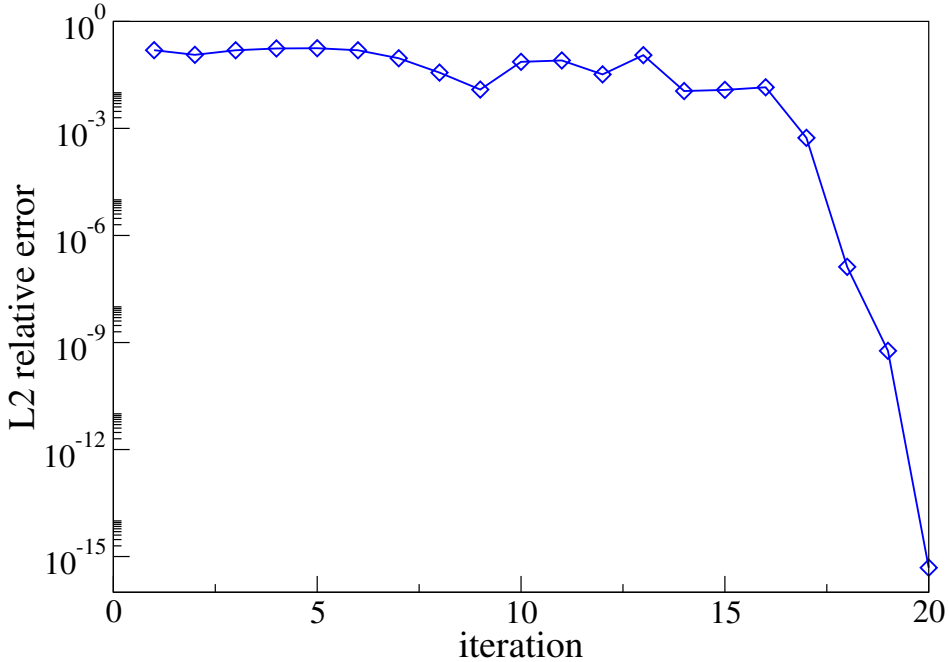
From now, apply the QR algorithm separately to $T_1^{(k)}$ and $T_2^{(k)}$ ("deflation")
8: **Stop** if convergence, $\mathrm{diag}(T^{(k)})$ contains the eigenvalues of A
9: **end for**

Computation of matrix spectra: example

$A \in \mathbb{R}^{n \times n}$ a random real symmetric matrix.

Stopping criterion of QR iterations: $\dfrac{\| T^{(k+1)} - T^{(k)} \|}{\| T^{(k)} \|} \leq 10^{-12}$

n	iterations (deflation)	iterations (no deflation)	spectrum error
10	17	254	5.1954e-16
20	37	1275	1.0795e-15
50	96	22365	1.1e-15
100	185		8.6751e-16
200	370		1.6564e-15
500	880		1.5776e-15



Spectrum error: $e_\lambda = \dfrac{\| \lambda - \lambda_{QR} \|}{\| \lambda \|}$, $\lambda = \{ \lambda_1, \ldots, \lambda_n \}$

- Deflation (here not fully implemented) dramatically reduces iteration count
- Very high accuracy on whole spectrum achievable (note however: comparison spectrum also numerical $\leftarrow$ `eig(A)`)

$\boxed{(A-\lambda I)x=0}$, A unsymmetric

- Set A in upper Hessenberg form (e.g. using Householder reflectors):

$$A = QHQ^{\mathsf{H}}, \qquad H = \begin{bmatrix} \times & \times & \times & \dots & \times \\ \times & \times & & & \times \\ 0 & \times & \times & & \times \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & \times & \times \end{bmatrix}$$

  Spectra of A and H coincide.

- (shifted) QR iterations applicable to H:

$$T^{(0)} = H, \text{ then } \quad \text{(a) } T^{(k-1)} - \mu^{(k)} I = Q^{(k)} R^{(k)} \quad \text{and} \quad \text{(b) } T^{(k)} = R^{(k)} Q^{(k)} + \mu^{(k)} I.$$

- If $A \in \mathbb{C}^{n\times n}$, $T^{(k)} \to T$ upper triangular;
- If $A \in \mathbb{R}^{n\times n}$ and QR algorithm in real arithmetic, $T^{(k)} \to T$ "almost upper triangular" (2×2 diagonal blocks $\to$ pairs of conjugate complex eigenvalues);