Eigenvalue Problems

Anton Evgrafov

October 29, 2014

Problem:

Find $\lambda \in \mathbb{C}$, $u : \Omega \to \mathbb{C}$:

$$\begin{cases} -\Delta u(z) = \lambda u(z), & z \in \Omega, \\ u(z) = 0, & z \in \partial\Omega \end{cases}$$

$$\Downarrow$$

discretization

$$\Downarrow$$

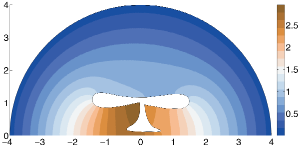
Find $\lambda \in \mathbb{C}$, $u \in \mathbb{C}^n$:

$$Au = \lambda u$$

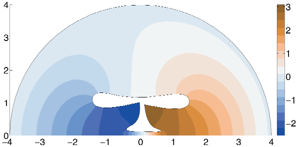
Eigenvalue problem!

Motivation: antennas

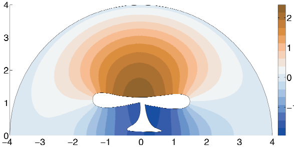
$f_1 \approx 28.5\text{MHz}$



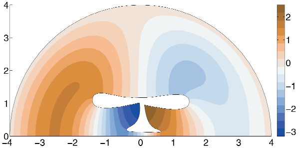
$f_2 \approx 34.6\text{MHz}$



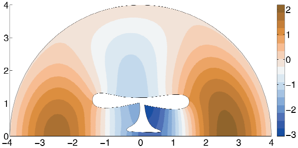
$f_3 \approx 42.6\text{MHz}$



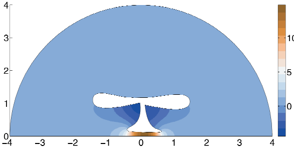
$f_4 \approx 59.6\text{MHz}$



$f_5 \approx 63.2\text{MHz}$



$f_{13} \approx 114.98\text{MHz}$



Motivation: mechanics

Motivation: SVD

$$A = U\Sigma V^*$$

$$\Downarrow$$

$$AA^* = U\Sigma\Sigma^{\mathrm{T}} U^*$$

$$A^*A = V\Sigma^{\mathrm{T}}\Sigma V^*$$

σ - SVD of A iff σ^2-eigenvalue of AA^* (or A^*A)

Straightforward algorithm for finding eigenvalues:

1. Compute coefficients of
$$p_A(\lambda) = \det(A - \lambda I)$$
2. Find roots
$$p_A(\lambda) = 0$$
3. Optionally/if needed: find eigenvectors

Expensive; numerically unstable!

Similar matrices

Definition: A and B-similar if $\exists X$, $\det(X) \neq 0$:

$$A = XBX^{-1}$$

Same eigenvalues (incl. algebraic multiplicity):

$$\begin{aligned} p_A(\lambda) &= \det(A - \lambda I) = \det[X(B - \lambda I)X^{-1}] \\ &= \det(X)\det(B - \lambda I)\det(X^{-1}) = \det(B - \lambda I) = p_B(\lambda) \end{aligned}$$

Similar matrices

Definition: A and B-similar if $\exists X$, $\det(X) \neq 0$:

$$A = XBX^{-1}$$

Same number of lin. indep. eigenvectors (geometric multiplicity):

$$\begin{aligned} Av &= \lambda v \\ XBX^{-1}v &= \lambda v \\ B[X^{-1}v] &= \lambda[X^{-1}v] \end{aligned}$$

Eigenvalue-revealing factorizations:

If $A = X\Lambda X^{-1}$, Λ-diagonal

$$AX = X\Lambda$$
$$AX_{*,j} = \Lambda_{j,j}X_{*,j}$$

Columns of X-eigenvectors, diagonal of Λ-eigenvalues!

Does not exist for defective matrices!

Eigenvalue-revealing factorizations:

Schur factorization: $A = QRQ^*$, $Q^*Q = I$, R-(upper) triangular
(Remember: $A^*A = AA^* \implies R$-diagonal.)

Exists for all matrices!

A similar to $R \implies$

$$\lambda_i(A) = \lambda_i(R) = R_{i,i}$$

Is existence proof constructive?

Not in the sense of numerical analysis...

Start with $Aq_1 = \lambda_1 q_1$, complete q to ON basis $\tilde{Q} = [q_1, \ldots, q_n]$

$$\begin{aligned}
\tilde{Q}^* A \tilde{Q} &= \tilde{Q}^* [Aq_1, Aq_2, \ldots, Aq_n] \\
&= \tilde{Q}^* [\lambda_1 q_1, Aq_2, \ldots, Aq_n] \\
&= [\lambda_1 e_1, \tilde{Q}^* Aq_2, \ldots, \tilde{Q}^* Aq_n] \\
&= \begin{bmatrix} \lambda_1 & w \\ 0 & \tilde{A} \end{bmatrix} = \begin{bmatrix} \lambda_1 & w \\ 0 & \hat{Q}\hat{R}\hat{Q}^* \end{bmatrix}
\end{aligned}$$

-used induction hypothesis on a smaller submatrix $\tilde{A}$.

Is existence proof constructive?

Not in the sense of numerical analysis...

Start with $Aq_1 = \lambda_1 q_1$, complete q to ON basis $\tilde{Q} = [q_1, \ldots, q_n]$

$$\tilde{Q}^* A \tilde{Q} = \begin{bmatrix} \lambda_1 & w \\ 0 & \hat{Q}\hat{R}\hat{Q}^* \end{bmatrix}$$

-used induction hypothesis on a smaller submatrix $\tilde{A}$.

$$A = \underbrace{\tilde{Q} \begin{bmatrix} 1 & 0 \\ 0 & \hat{Q} \end{bmatrix}}_{=:Q} \underbrace{\begin{bmatrix} \lambda_1 & w\hat{Q} \\ 0 & \hat{R} \end{bmatrix}}_{=:R} \underbrace{\begin{bmatrix} 1 & 0 \\ 0 & \hat{Q}^* \end{bmatrix} \tilde{Q}^*}_{=:Q^*}$$

Recall Householder QR-factorization algorithm

$$Q_1^* \begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \end{bmatrix} = \begin{bmatrix} \times & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & \times & \times & \times & \times \end{bmatrix}$$

Recall Householder QR-factorization algorithm

$$Q_2^* Q_1^* \begin{bmatrix} \times & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & \times & \times & \times & \times \end{bmatrix} = \begin{bmatrix} \times & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & & \times & \times & \times \\ & & \times & \times & \times \\ & & \times & \times & \times \end{bmatrix}$$

Could we do something similar for Schur?

$$Q_1^* \begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \end{bmatrix} Q_1 = \begin{bmatrix} \times & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & \times & \times & \times & \times \end{bmatrix}$$

$$\vdots$$

$$Q^* \begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \end{bmatrix} Q = \begin{bmatrix} \times & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & & \times & \times & \times \\ & & & \times & \times \\ & & & & \times \end{bmatrix}$$

Not possible!

Connexion between polynomial roots and eigenvalues

Arbitrary polynomial:

$$p(x) = x^n + a_{n-1}x^{n-1} + \cdots + a_1 x + a_0$$

Define:

$$A = \begin{bmatrix} 0 & & & & & -a_0 \\ 1 & 0 & & & & -a_1 \\ & 1 & 0 & & & -a_2 \\ & & 1 & \ddots & & \vdots \\ & & & \ddots & 0 & -a_{n-2} \\ & & & & 1 & -a_{n-1} \end{bmatrix}$$

Then $p(\lambda) = \det[\lambda I - A]$!

No formula for polynomial roots (Abel, 1842) $\implies$ no **finite** algorithm for eigenvalues!

Remember: all eigenvalue algorithms are inherently iterative!
(But in practice, only a few iterations are needed for good algorithms.)

Phase 1/phase 2 approach

Phase 1: finite factorization

$$A = Q\tilde{A}Q^*$$

$Q^*Q = I$, $\tilde{A}$-significantly simpler than A

Phase 2: iterative computation of eigenvalues of $\tilde{A}$ (hence also A)

Phase 1

$$A = QHQ^*$$

$Q^*Q = I$, H-(upper) Hessenberg
Algorithm: similar to Householder-QR

$$Q_1^* \begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \end{bmatrix} = \begin{bmatrix} \times & \times & \times & \times & \times \\ \color{blue}{\times} & \color{blue}{\times} & \color{blue}{\times} & \color{blue}{\times} & \color{blue}{\times} \\ & \color{blue}{\times} & \color{blue}{\times} & \color{blue}{\times} & \color{blue}{\times} \\ & \color{blue}{\times} & \color{blue}{\times} & \color{blue}{\times} & \color{blue}{\times} \\ & \color{blue}{\times} & \color{blue}{\times} & \color{blue}{\times} & \color{blue}{\times} \end{bmatrix}$$

Phase 1

$$A = QHQ^*$$

$Q^*Q = I$, H-(upper) Hessenberg
Algorithm: similar to Householder-QR

$$Q_1^* \begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \end{bmatrix} = \begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & \times & \times & \times & \times \end{bmatrix}$$

$$\begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & \times & \times & \times & \times \end{bmatrix} Q_1 = \begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & \times & \times & \times & \times \end{bmatrix}$$

Phase 1

$$A = QHQ^*$$

$Q^*Q = I$, H-(upper) Hessenberg
Algorithm: similar to Householder-QR

$$Q_2^* \begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & \times & \times & \times & \times \end{bmatrix} = \begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & & \times & \times & \times \\ & & \times & \times & \times \end{bmatrix}$$

Phase 1

$$A = QHQ^*$$

$Q^*Q = I$, H-(upper) Hessenberg
Algorithm: similar to Householder-QR

$$Q_2^* \begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & \times & \times & \times & \times \end{bmatrix} = \begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & & \times & \times & \times \\ & & \times & \times & \times \end{bmatrix}$$

$$\begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & & \times & \times & \times \\ & & \times & \times & \times \end{bmatrix} Q_2^* = \begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & & \times & \times & \times \\ & & \times & \times & \times \end{bmatrix}$$

Full algorithm: Householder Hessenberg

1: **for** $k = 1$ to $n - 2$ **do**
2: $\quad x := A_{k+1:n,k}$
3: $\quad v_k := \text{sign}(x_1)\|x\|_2 e_1 + x$
4: $\quad v_k := v_k / \|v_k\|_2$
5: $\quad A_{k+1:n,k:n} = A_{k+1:n,k:n} - 2v_k(v_k^* A_{k+1:n,k:n})$
6: $\quad A_{1:n,k+1:n} = A_{1:n,k+1:n} - 2(A_{1:n,k+1:n} v_k) v_k^*$
7: **end for**

Complexity:

$$\sim \sum_{k=1}^{n-2} \left[\underbrace{4(n-k)^2}_{\text{QR,line 5}} + \underbrace{4n(n-k)}_{\text{line 6}} \right] \approx \underbrace{4n^3/3}_{\text{QR}} + 4n^3/2$$

$$= 10n^3/3 = 2.5 \times \text{QR}$$

Phase 1, $A^* = A$

$$A = QHQ^*$$
$$A^* = QH^*Q^*$$
$$\Downarrow$$

H— tri-diagonal

Householder Hessenberg complexity: same as QR

Phase 2

Given: Hessenberg (or tri-diagonal) matrix
Construct iterative process for computing eigenvalues

Rayleigh quotient

Suppose $x \in \mathbb{C}^n \setminus \{0\}$ is a given approximation of an eigenvector.
Task: find eigenvalue!
I.e., find $\lambda \in \mathbb{C}$:

$$x\lambda \approx Ax$$

Rayleigh quotient

Suppose $x \in \mathbb{C}^n \setminus \{0\}$ is a given approximation of an eigenvector.
Task: find eigenvalue!
I.e., find $\lambda \in \mathbb{C}$:

$$x\lambda \approx Ax$$

Solve as least-squares problem! Normal equations:

$$\begin{aligned}(x^*x)\lambda &= x^*Ax \\ \lambda &= \frac{x^*Ax}{x^*x} =: r(x)\end{aligned}$$

Properties:

1. $r(\alpha x) = r(x)$, $\forall \alpha \neq 0$, $x \neq 0$
2. r: smooth function of x [for $x \in \mathbb{C}^n$: consider real and complex components]
3. $\|Ax - r(x)x\| = 0$ iff x-eigenvector; then $r(x)$ is eigenvalue.

Possible algorithm: solve a system of non-linear equations
$Ax - r(x)x = 0$, $\|x\| = 1$.

Derivatives:

Assume $x \in \mathbb{R}^n \setminus \{0\}$, $A \in \mathbb{R}^{n \times n}$:

$$
\begin{aligned}
\partial_i r(x) &= \frac{[\partial_i (x^{\mathrm{T}} A x)] x^{\mathrm{T}} x - x^{\mathrm{T}} A x \partial_i (x^{\mathrm{T}} x)}{(x^{\mathrm{T}} x)^2} \\
&= \frac{([Ax]_i + [A^{\mathrm{T}} x]_i) x^{\mathrm{T}} x - 2(x^{\mathrm{T}} A x) x_i}{(x^{\mathrm{T}} x)^2} \\
&= \frac{2}{x^{\mathrm{T}} x} \left[\frac{[Ax]_i + [A^{\mathrm{T}} x]_i}{2} - r(x) x_i \right]
\end{aligned}
$$

$$
\nabla r(x) = \frac{2}{x^{\mathrm{T}} x} \left[\frac{A + A^{\mathrm{T}}}{2} x - r(x) x \right]
$$

Derivatives:

Additionally assume $A = A^{\mathrm{T}}$. Then

$$\nabla r(x) = \frac{2}{x^{\mathrm{T}}x}[Ax - r(x)x]$$

Therefore $\nabla r(x) = 0$ iff $x \neq 0$ – eigenvector.

In particular, if $A\bar{x} = \lambda\bar{x}$ then

$$r(x) - \lambda = O(\|x - \bar{x}\|^2),$$

in the vicinity of $\bar{x}$.

(Note: in non-Hermitian case $r(x) - \lambda = O(\|x - \bar{x}\|)$ only.)