Polynomial root-finding using companion matrices

Fernando De Terán

Departamento de Matemáticas
Universidad Carlos III de Madrid
(Spain)

TUBerlin

February 2017

Outline

Outline

Goal

Compute the roots of (scalar) polynomials

$$p(z) = a_n z^n + a_{n-1} z^{n-1} + \cdots + a_1 z + a_0 \qquad (a_k \in \mathbb{C})$$

using companion forms.

We can restrict ourselves to **monic** polynomials (after dividing by a_n, if necessary)

$$p(z) = z^n + a_{n-1} z^{n-1} + \cdots + a_1 z + a_0 \qquad (a_k \in \mathbb{C})$$

...Can we ??? (more on this later)

Goal

Compute the roots of (scalar) polynomials

$$p(z) = a_n z^n + a_{n-1} z^{n-1} + \cdots + a_1 z + a_0 \qquad (a_k \in \mathbb{C})$$

using companion forms.

We can restrict ourselves to **monic** polynomials (after dividing by a_n, if necessary)

$$p(z) = z^n + a_{n-1} z^{n-1} + \cdots + a_1 z + a_0 \qquad (a_k \in \mathbb{C})$$

...Can we ??? (more on this later)

Goal

Compute the roots of (scalar) polynomials

$$p(z) = a_n z^n + a_{n-1} z^{n-1} + \cdots + a_1 z + a_0 \qquad (a_k \in \mathbb{C})$$

using companion forms.

We can restrict ourselves to **monic** polynomials (after dividing by a_n, if necessary)

$$p(z) = z^n + a_{n-1} z^{n-1} + \cdots + a_1 z + a_0 \qquad (a_k \in \mathbb{C})$$

...Can we ??? (more on this later)

Companion matrix

Companion matrix

$A(a_0, a_1, \ldots, a_{n-1})$ such that

$$p_A(z) = \det(zI - A) = z^n + a_{n-1}z^{n-1} + \cdots + a_1 z + a_0 = p(z)$$

(Only for monic polynomials).

Roots of $p(z)$ = Eigenvalues of A (i.e.: $p(z) = 0 \Leftrightarrow \det(zI - A) = 0$)

Theoretically:

Polynomial root-finding $\overset{\text{Companion matrices}}{\Longleftrightarrow}$ Standard eigenvalue problem

But numerically, they are different problems !!!

Companion matrix

Companion matrix

$A(a_0, a_1, \ldots, a_{n-1})$ such that

$$p_A(z) = \det(zI - A) = z^n + a_{n-1}z^{n-1} + \cdots + a_1 z + a_0 = p(z)$$

(Only for monic polynomials).

Roots of $p(z)$ = Eigenvalues of A (i.e.: $p(z) = 0 \Leftrightarrow \det(zI - A) = 0$)

Theoretically:

Polynomial root-finding $\overset{\text{Companion matrices}}{\Longleftrightarrow}$ Standard eigenvalue problem

But numerically, they are different problems !!!

Companion matrix

Companion matrix

$A(a_0, a_1, \ldots, a_{n-1})$ such that

$$p_A(z) = \det(zI - A) = z^n + a_{n-1}z^{n-1} + \cdots + a_1 z + a_0 = p(z)$$

(Only for monic polynomials).

Roots of $p(z)$ = Eigenvalues of A (i.e.: $p(z) = 0 \Leftrightarrow \det(zI - A) = 0$)

Theoretically:

Polynomial root-finding $\overset{\text{Companion matrices}}{\Longleftrightarrow}$ Standard eigenvalue problem

But numerically, they are different problems !!!

Companion matrix

Companion matrix

$A(a_0, a_1, \ldots, a_{n-1})$ such that

$$p_A(z) = \det(zI - A) = z^n + a_{n-1}z^{n-1} + \cdots + a_1 z + a_0 = p(z)$$

(Only for monic polynomials).

Roots of $p(z)$ = Eigenvalues of A (i.e.: $p(z) = 0 \Leftrightarrow \det(zI - A) = 0$)

Theoretically:

Polynomial root-finding $\overset{\text{Companion matrices}}{\Longleftrightarrow}$ Standard eigenvalue problem

But numerically, they are different problems !!!

Motivation

Frobenius companion matrices:

$$C_1 = \begin{bmatrix} -a_{n-1} & -a_{n-2} & \cdots & -a_0 \\ 1 & 0 & \cdots & 0 \\ & \ddots & \ddots & \vdots \\ 0 & & 1 & 0 \end{bmatrix}, \qquad C_2 = C_1^\top$$

MATLAB's command `roots`: QR algorithm on C_2

Companion forms

Companion form: Valid for non-monic polynomials.

Companion form

$A(a_0, a_1, \ldots, a_{n-1}, a_n) \quad \rightsquigarrow \quad$ May have entries of the form $a + bz$

(More on this later)

Companion forms

Companion form: Valid for non-monic polynomials.

Companion form

$A(a_0, a_1, \ldots, a_{n-1}, a_n) \quad \rightsquigarrow$ May have entries of the form $a + bz$

(More on this later)

Outline

Backward stability

Problem:	Algorithm:
$f : \underbrace{X}_{data} \longrightarrow \underbrace{Y}_{solution}$	$\widetilde{f} : \underbrace{X}_{data} \longrightarrow \underbrace{Y}_{solution}$

$\widetilde{f}$ is backward stable if $\widetilde{f}(x) = f(x + \delta x), \quad \|\delta x\| = O(u)\|x\|$

($u =$ unit roundoff)

☞ Different approaches for **poly root-finding** using **companion matrices**:

1. B'stability on the companion matrix (e-vals):

   $\widetilde{f} =$ e-val algorithm, $f =$ e-val problem, $x =$ companion matrix

   (the computed roots (e-vals) are the e-vals of a nearby matrix (not necessarily companion!!!))

2. B'stability on the polynomial (roots):

   $\widetilde{f} =$ e-val algorithm, $f =$ polynomial root-finding, $x =$ polynomial

   (the computed roots (e-vals) are the roots of a nearby polynomial)

Backward stability

Problem:	Algorithm:
$f : \underbrace{X}_{data} \longrightarrow \underbrace{Y}_{solution}$	$\widetilde{f} : \underbrace{X}_{data} \longrightarrow \underbrace{Y}_{solution}$

$\widetilde{f}$ is backward stable if

($u =$ unit roundoff)

$$\widetilde{f}(x) = f(x + \delta x), \qquad \|\delta x\| = O(u)\|x\|$$

☞ Different approaches for **poly root-finding** using **companion matrices**:

1. B'stability on the companion matrix (e-vals):

   $\widetilde{f} =$ e-val algorithm, $f =$ e-val problem, $x =$ companion matrix

   (the computed roots (e-vals) are the e-vals of a nearby matrix (not necessarily companion!!!))

2. B'stability on the polynomial (roots):

   $\widetilde{f} =$ e-val algorithm, $f =$ polynomial root-finding, $x =$ polynomial

   (the computed roots (e-vals) are the roots of a nearby polynomial)

Backward stability

Problem:	Algorithm:
$f: \underbrace{X}_{data} \longrightarrow \underbrace{Y}_{solution}$	$\widetilde{f}: \underbrace{X}_{data} \longrightarrow \underbrace{Y}_{solution}$

$\widetilde{f}$ is backward stable if (u = unit roundoff)

$$\widetilde{f}(x) = f(x+\delta x), \qquad \|\delta x\| = O(u)\|x\|$$

☞ Different approaches for **poly root-finding** using **companion matrices**:

1. B'stability on the companion matrix (e-vals):

   $\widetilde{f}$ = e-val algorithm, f = e-val problem, x = companion matrix

   (the computed roots (e-vals) are the e-vals of a nearby matrix (not necessarily companion!!!))

2. B'stability on the polynomial (roots):

   $\widetilde{f}$ = e-val algorithm, f = polynomial root-finding, x = polynomial

   (the computed roots (e-vals) are the roots of a nearby polynomial)

Backward stability

Problem:	Algorithm:
$f : \underbrace{X}_{data} \longrightarrow \underbrace{Y}_{solution}$	$\widetilde{f} : \underbrace{X}_{data} \longrightarrow \underbrace{Y}_{solution}$

$\widetilde{f}$ is backward stable if ($u =$ unit roundoff)

$$\widetilde{f}(x) = f(x + \delta x), \qquad \|\delta x\| = O(u)\|x\|$$

☞ Different approaches for **poly root-finding** using **companion matrices**:

1. B'stability on the companion matrix (e-vals):

   $\widetilde{f} =$ e-val algorithm, $f =$ e-val problem, $x =$ companion matrix

   (the computed roots (e-vals) are the e-vals of a nearby matrix (not necessarily companion!!!))

2. B'stability on the polynomial (roots):

   $\widetilde{f} =$ e-val algorithm, $f =$ polynomial root-finding, $x =$ polynomial

   (the computed roots (e-vals) are the roots of a nearby polynomial)

Complexity

Complexity (Computational cost)

- Computation time (number of flops)
- Storage

$$p(z): \begin{array}{cc} n & \text{coefficients} \\ n & \text{roots} \end{array} \Bigg\} \rightsquigarrow$$ **Desideratum**: $O(n^2)$ flops + $O(n)$ storage

(I won't pay much attention to this!)

Complexity

Complexity (Computational cost)

- Computation time (number of flops)
- Storage

$$p(z): \quad \left.\begin{array}{cc} n & \text{coefficients} \\ n & \text{roots} \end{array}\right\} \rightsquigarrow$$ **Desideratum**: $O(n^2)$ flops + $O(n)$ storage

(I won't pay much attention to this!)

Complexity

Complexity (Computational cost)

- Computation time (number of flops)
- Storage

$$p(z): \left.\begin{array}{ll} n & \text{coefficients} \\ n & \text{roots} \end{array}\right\} \rightsquigarrow \textbf{Desideratum}: O(n^2) \text{ flops} + O(n) \text{ storage}$$

(I won't pay much attention to this!)

Outline

MATLAB'S command `roots`

- Uses a Frobenius companion matrix.

MATLAB'S command `roots`

- Uses a Frobenius companion matrix.

☞ Is b'stable in the matrix sense.

MATLAB'S command `roots`

- Uses a Frobenius companion matrix.

👍 Is b'stable in the matrix sense.

👎 Too expensive: $O(n^3)$ computations and $O(n^2)$ storage.

MATLAB’S command `roots`

- Uses a Frobenius companion matrix.

👍 Is b’stable in the matrix sense.

👎 Too expensive: $O(n^3)$ computations and $O(n^2)$ storage.

👎 Not b’stable in the polynomial sense.

MATLAB'S command `roots`

- Uses a Frobenius companion matrix.

👍 Is b'stable in the matrix sense.

👎 Too expensive: $O(n^3)$ computations and $O(n^2)$ storage.

👎 Not b'stable in the polynomial sense.

👍 In practice, it can be considered b'stable (in the poly sense) $\rightsquigarrow$ balancing!!

MATLAB'S command `roots`

- Uses a Frobenius companion matrix.

☞ Too expensive: $O(n^3)$ computations and $O(n^2)$ storage.

☞ Not b'stable in the polynomial sense.

Faster algorithms

- Many variants of the QR algorithm: [Calvetti-etal'02], [Bini-etal'04, '05, '10], [Gemignani'07], [Chandrasekharan-etal'08], [Van Barel-etal'10], [Aurentz-etal'13]
  - $O(n^2)$ computations and $O(n)$ storage.
  - Not b'stable (in the matrix sense)!!!
- Variants of C_1, C_2 ([Brugnano-Trigiante'95], [Niu-Sakurai'03]): Improve the accuracy of multiple roots.

☞ [Aurentz-Mach-Vandebril-Watkins'15]: Fast and b'stable method.

  - $O(n^2)$ computations and $O(n)$ storage.
  - B'stable (in the matrix sense).

Faster algorithms

- Many variants of the QR algorithm: [Calvetti-etal'02], [Bini-etal'04, '05, '10], [Gemignani'07], [Chandrasekharan-etal'08], [Van Barel-etal'10], [Aurentz-etal'13]
  - $O(n^2)$ computations and $O(n)$ storage.
  - Not b'stable (in the matrix sense)!!!
- Variants of C_1, C_2 ([Brugnano-Trigiante'95], [Niu-Sakurai'03]): Improve the accuracy of multiple roots.

☞ [Aurentz-Mach-Vandebril-Watkins'15]: Fast and b'stable method.

- $O(n^2)$ computations and $O(n)$ storage.
- B'stable (in the matrix sense).

Faster algorithms

- Many variants of the QR algorithm: [Calvetti-etal'02], [Bini-etal'04, '05, '10], [Gemignani'07], [Chandrasekharan-etal'08], [Van Barel-etal'10], [Aurentz-etal'13]
  - $O(n^2)$ computations and $O(n)$ storage.
  - Not b'stable (in the matrix sense)!!!
- Variants of C_1, C_2 ([Brugnano-Trigiante'95], [Niu-Sakurai'03]): Improve the accuracy of multiple roots.

☞ [Aurentz-Mach-Vandebril-Watkins'15]: Fast and b'stable method.

- $O(n^2)$ computations and $O(n)$ storage.
- B'stable (in the matrix sense).

Faster algorithms

- Many variants of the QR algorithm: [Calvetti-etal'02], [Bini-etal'04, '05, '10], [Gemignani'07], [Chandrasekharan-etal'08], [Van Barel-etal'10], [Aurentz-etal'13]
  - $O(n^2)$ computations and $O(n)$ storage.
  - Not b'stable (in the matrix sense)!!!
- Variants of C_1, C_2 ([Brugnano-Trigiante'95], [Niu-Sakurai'03]): Improve the accuracy of multiple roots.

☞ [Aurentz-Mach-Vandebril-Watkins'15]: Fast and b'stable method.

  - $O(n^2)$ computations and $O(n)$ storage.
  - B'stable (in the matrix sense).

Faster algorithms

- Many variants of the QR algorithm: [Calvetti-etal'02], [Bini-etal'04, '05, '10], [Gemignani'07], [Chandrasekharan-etal'08], [Van Barel-etal'10], [Aurentz-etal'13]
  - $O(n^2)$ computations and $O(n)$ storage.
  - Not b'stable (in the matrix sense)!!!
- Variants of C_1, C_2 ([Brugnano-Trigiante'95], [Niu-Sakurai'03]): Improve the accuracy of multiple roots.

☞ [Aurentz-Mach-Vandebril-Watkins'15]: Fast and b'stable method.

- $O(n^2)$ computations and $O(n)$ storage.
- B'stable (in the matrix sense).

Faster algorithms

- Many variants of the QR algorithm: [Calvetti-etal'02], [Bini-etal'04, '05, '10], [Gemignani'07], [Chandrasekharan-etal'08], [Van Barel-etal'10], [Aurentz-etal'13]
  - $O(n^2)$ computations and $O(n)$ storage.
  - Not b'stable (in the matrix sense)!!!
- Variants of C_1, C_2 ([Brugnano-Trigiante'95], [Niu-Sakurai'03]): Improve the accuracy of multiple roots.

☞ [Aurentz-Mach-Vandebril-Watkins'15]: Fast and b'stable method.

  - $O(n^2)$ computations and $O(n)$ storage.
  - B'stable (in the matrix sense).

Faster algorithms

- Many variants of the QR algorithm: [Calvetti-etal'02], [Bini-etal'04, '05, '10], [Gemignani'07], [Chandrasekharan-etal'08], [Van Barel-etal'10], [Aurentz-etal'13]
  - $O(n^2)$ computations and $O(n)$ storage.
  - Not b'stable (in the matrix sense)!!!
- Variants of C_1, C_2 ([Brugnano-Trigiante'95], [Niu-Sakurai'03]): Improve the accuracy of multiple roots.

☞ [Aurentz-Mach-Vandebril-Watkins'15]: Fast and b'stable method.

  - $O(n^2)$ computations and $O(n)$ storage.
  - B'stable (in the matrix sense).

Faster algorithms

- Many variants of the QR algorithm: [Calvetti-etal'02], [Bini-etal'04, '05, '10], [Gemignani'07], [Chandrasekharan-etal'08], [Van Barel-etal'10], [Aurentz-etal'13]
  - $O(n^2)$ computations and $O(n)$ storage.
  - Not b'stable (in the matrix sense)!!!
- Variants of C_1, C_2 ([Brugnano-Trigiante'95], [Niu-Sakurai'03]): Improve the accuracy of multiple roots.

☞ [Aurentz-Mach-Vandebril-Watkins'15]: Fast and b'stable method.

  - $O(n^2)$ computations and $O(n)$ storage.
  - B'stable (in the matrix sense).

Q: B'stability in the polynomial sense ???

B'err of polynomial root-finding using companion matrices

Given $p(z)$ → Choose A such that $p(z) = \det(zI - A)$ → Compute the e-vals of A = e-vals of $A + E$, $\|E\| = O(u)\|A\|$

(if we use a backward stable algorithm, like QR)

Set $\widetilde{p}(z) = \det(zI - (A + E))$

Question: Is $\widetilde{p}(z)$ close to $p(z)$?

$$\frac{\|p - \widetilde{p}\|}{\|p\|} = O(u) \ ?$$

$\frac{\|p - \widetilde{p}\|}{\|p\|}$: b'err of polynomial root-finding as an eigenvalue problem (using A).

Goal:

Analyze $\frac{\|p - \widetilde{p}\|}{\|p\|}$, for A a **Fiedler matrix**.

B'err of polynomial root-finding using companion matrices

Given $p(z)$ $\rightarrow$ Choose A such that $p(z) = \det(zI - A)$ $\rightarrow$ Compute the e-vals of A $=$ e-vals of $A+E$, $\|E\| = O(u)\|A\|$

(if we use a backward stable algorithm, like QR)

Set $\widetilde{p}(z) = \det(zI - (A+E))$

Question: Is $\widetilde{p}(z)$ close to $p(z)$?

$$\frac{\|p-\widetilde{p}\|}{\|p\|} = O(u) \ ?$$

$\frac{\|p-\widetilde{p}\|}{\|p\|}$: b'err of polynomial root-finding as an eigenvalue problem (using A).

Goal:

Analyze $\frac{\|p-\widetilde{p}\|}{\|p\|}$, for A a **Fiedler matrix**.

B'err of polynomial root-finding using companion matrices

Given $p(z)$ → Choose A such that $p(z) = \det(zI - A)$ → Compute the e-vals of A = e-vals of $A + E$, $\|E\| = O(u)\|A\|$

(if we use a backward stable algorithm, like QR)

Set $\widetilde{p}(z) = \det(zI - (A + E))$

Question: Is $\widetilde{p}(z)$ close to $p(z)$?

$$\frac{\|p - \widetilde{p}\|}{\|p\|} = O(u) \ ?$$

$\frac{\|p - \widetilde{p}\|}{\|p\|}$: b'err of polynomial root-finding as an eigenvalue problem (using A).

Goal:

Analyze $\frac{\|p - \widetilde{p}\|}{\|p\|}$, for A a **Fiedler matrix**.

B'err of polynomial root-finding using companion matrices

Given $p(z)$ → Choose A such that $p(z) = \det(zI - A)$ → Compute the e-vals of A = e-vals of $A + E$, $\|E\| = O(u)\|A\|$

(if we use a backward stable algorithm, like QR)

Set $\widetilde{p}(z) = \det(zI - (A + E))$

Question: Is $\widetilde{p}(z)$ close to $p(z)$?

$$\frac{\|p - \widetilde{p}\|}{\|p\|} = O(u) \text{ ?}$$

$\frac{\|p - \widetilde{p}\|}{\|p\|}$: b'err of polynomial root-finding as an eigenvalue problem (using A).

Goal:

Analyze $\frac{\|p - \widetilde{p}\|}{\|p\|}$, for A a **Fiedler matrix**.

B'err of polynomial root-finding using companion matrices

Given $p(z)$ → Choose A such that $p(z) = \det(zI - A)$ → Compute the e-vals of A = e-vals of $A + E$, $\|E\| = O(u)\|A\|$

(if we use a backward stable algorithm, like QR)

Set $\widetilde{p}(z) = \det(zI - (A + E))$

Question: Is $\widetilde{p}(z)$ close to $p(z)$?

$$\frac{\|p - \widetilde{p}\|}{\|p\|} = O(u) \ ?$$

$\frac{\|p - \widetilde{p}\|}{\|p\|}$: b'err of polynomial root-finding as an eigenvalue problem (using A).

Goal:

Analyze $\frac{\|p - \widetilde{p}\|}{\|p\|}$, for A a **Fiedler matrix**.

B'err of polynomial root-finding using companion matrices

Given $p(z)$ → Choose A such that $p(z) = \det(zI - A)$ → Compute the e-vals of A = e-vals of $A + E$, $\|E\| = O(u)\|A\|$

(if we use a backward stable algorithm, like QR)

Set $\widetilde{p}(z) = \det(zI - (A + E))$

Question: Is $\widetilde{p}(z)$ close to $p(z)$?

$$\frac{\|p - \widetilde{p}\|}{\|p\|} = O(u) \text{ ?}$$

$\frac{\|p - \widetilde{p}\|}{\|p\|}$: b'err of polynomial root-finding as an eigenvalue problem (using A).

Goal:

Analyze $\frac{\|p - \widetilde{p}\|}{\|p\|}$, for A a **Fiedler matrix**.

B'err of polynomial root-finding using companion matrices

Given $p(z)$ → Choose A such that $p(z) = \det(zI - A)$ → Compute the e-vals of A = e-vals of $A+E$, $\|E\| = O(u)\|A\|$

(if we use a backward stable algorithm, like QR)

Set $\widetilde{p}(z) = \det(zI - (A+E))$

Question: Is $\widetilde{p}(z)$ close to $p(z)$?

$$\boxed{\frac{\|p - \widetilde{p}\|}{\|p\|} = O(u) \ \textbf{?}}$$

$\frac{\|p - \widetilde{p}\|}{\|p\|}$: b'err of polynomial root-finding as an eigenvalue problem (using A).

Goal:

Analyze $\frac{\|p - \widetilde{p}\|}{\|p\|}$, for A a **Fiedler matrix**.

Fiedler matrices: definition

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

$$M_0 = \begin{bmatrix} I_{n-1} & \\ & -a_0 \end{bmatrix}, \quad M_k = \begin{bmatrix} I_{n-k-1} & & \\ & \boxed{\begin{matrix} -a_k & 1 \\ 1 & 0 \end{matrix}} & \\ & & I_{k-1} \end{bmatrix}, \quad k = 1, \ldots, n-1.$$

Let $\sigma : \{0, 1, \ldots, n-1\} \to \{1, \ldots, n\}$ be a bijection. Then:

$$M_\sigma := M_{\sigma^{-1}(1)} \cdots M_{\sigma^{-1}(n)}$$

Fiedler matrix of p associated with the bijection σ

- Introduced by **Fiedler** in 2003.

Fiedler matrices: definition

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

$$M_0 = \begin{bmatrix} I_{n-1} & \\ & -a_0 \end{bmatrix}, \quad M_k = \begin{bmatrix} I_{n-k-1} & & \\ & \boxed{\begin{matrix} -a_k & 1 \\ 1 & 0 \end{matrix}} & \\ & & I_{k-1} \end{bmatrix}, \quad k = 1, \ldots, n-1.$$

Let $\sigma : \{0, 1, \ldots, n-1\} \rightarrow \{1, \ldots, n\}$ be a bijection. Then:

$$M_\sigma := M_{\sigma^{-1}(1)} \cdots M_{\sigma^{-1}(n)}$$

Fiedler matrix of p associated with the bijection σ

- Introduced by **Fiedler** in 2003.

Fiedler matrices: definition

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

$$M_0 = \begin{bmatrix} I_{n-1} & \\ & -a_0 \end{bmatrix}, \quad M_k = \begin{bmatrix} I_{n-k-1} & & \\ & \boxed{\begin{matrix} -a_k & 1 \\ 1 & 0 \end{matrix}} & \\ & & I_{k-1} \end{bmatrix}, \quad k = 1, \ldots, n-1.$$

Let $\sigma : \{0, 1, \ldots, n-1\} \to \{1, \ldots, n\}$ be a bijection. Then:

$$M_\sigma := M_{\sigma^{-1}(1)} \cdots M_{\sigma^{-1}(n)}$$

Fiedler matrix of p associated with the bijection σ

- Introduced by **Fiedler** in 2003.

Fiedler matrices: some examples

- Frobenius companion matrices:

$$C_1 = M_{n-1} \cdots M_1 M_0 = \begin{bmatrix} -a_{n-1} & -a_{n-2} & \cdots & -a_0 \\ 1 & 0 & \cdots & 0 \\ & \ddots & \ddots & \vdots \\ 0 & & 1 & 0 \end{bmatrix}$$

$$C_2 = M_0 M_1 \cdots M_{n-1} = C_1^\top$$

- $$M_{n-1} \cdots M_2 M_0 M_1 = \begin{bmatrix} -a_{n-1} & -a_{n-2} & \cdots & 1 \\ 1 & 0 & \cdots & 0 \\ & \ddots & \ddots & \vdots \\ 0 & & -a_0 & 0 \end{bmatrix}$$

- $$M_6(M_4 M_5)(M_2 M_3)(M_0 M_1) = \begin{bmatrix} -a_5 & 1 & 0 & 0 & 0 & 0 \\ -a_4 & 0 & -a_3 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -a_2 & 0 & -a_1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -a_0 & 0 \end{bmatrix} \qquad (n = 6)$$

Fiedler matrices: some examples

- Frobenius companion matrices:

$$C_1 = M_{n-1} \cdots M_1 M_0 = \begin{bmatrix} -a_{n-1} & -a_{n-2} & \cdots & -a_0 \\ 1 & 0 & \cdots & 0 \\ & \ddots & \ddots & \vdots \\ 0 & & 1 & 0 \end{bmatrix}$$

$$C_2 = M_0 M_1 \cdots M_{n-1} = C_1^\top$$

- $$M_{n-1} \cdots M_2 M_0 M_1 = \begin{bmatrix} -a_{n-1} & -a_{n-2} & \cdots & 1 \\ 1 & 0 & \cdots & 0 \\ & \ddots & \ddots & \vdots \\ 0 & & -a_0 & 0 \end{bmatrix}$$

- $$M_6(M_4 M_5)(M_2 M_3)(M_0 M_1) = \begin{bmatrix} -a_5 & 1 & 0 & 0 & 0 & 0 \\ -a_4 & 0 & -a_3 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -a_2 & 0 & -a_1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -a_0 & 0 \end{bmatrix} \quad (n = 6)$$

Fiedler matrices: some examples

- Frobenius companion matrices:

$$C_1 = M_{n-1} \cdots M_1 M_0 = \begin{bmatrix} -a_{n-1} & -a_{n-2} & \cdots & -a_0 \\ 1 & 0 & \cdots & 0 \\ & \ddots & \ddots & \vdots \\ 0 & & 1 & 0 \end{bmatrix}$$

$$C_2 = M_0 M_1 \cdots M_{n-1} = C_1^\top$$

- $$M_{n-1} \cdots M_2 \color{red}{M_0 M_1} = \begin{bmatrix} -a_{n-1} & -a_{n-2} & \cdots & \color{red}{1} \\ 1 & 0 & \cdots & 0 \\ & \ddots & \ddots & \vdots \\ 0 & & \color{red}{-a_0} & 0 \end{bmatrix}$$

- $$M_6(M_4 M_5)(M_2 M_3)(M_0 M_1) = \begin{bmatrix} -a_5 & 1 & 0 & 0 & 0 & 0 \\ -a_4 & 0 & -a_3 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -a_2 & 0 & -a_1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -a_0 & 0 \end{bmatrix} \qquad (n = 6)$$

Fiedler matrices: Basic properties

- All M_σ contain the same entries (located in different positions):

$$-a_0, \ldots, -a_{n-1} \quad \& \quad \overbrace{1, \ldots, 1}^{n-1} \quad \& \quad 0's$$

- M_σ is a companion matrix ($\det(zI - M_\sigma) = p(z)$).

- There are 2^{n-1} different Fiedler matrices.

Fiedler matrices: Basic properties

- All M_σ contain the same entries (located in different positions):

$$-a_0, \dots, -a_{n-1} \quad \& \quad \overbrace{1, \dots, 1}^{n-1} \quad \& \quad 0's$$

- M_σ is a companion matrix ($\det(zI - M_\sigma) = p(z)$).

- There are 2^{n-1} different Fiedler matrices.

Fiedler matrices: Basic properties

- All M_σ contain the same entries (located in different positions):

$$-a_0, \dots, -a_{n-1} \quad \& \quad \overbrace{1, \dots, 1}^{n-1} \quad \& \quad 0's$$

- M_σ is a companion matrix ($\det(zI - M_\sigma) = p(z)$).
- There are 2^{n-1} different Fiedler matrices.

Outline

Perturbation of the characteristic polynomial: first order term

Using Jacobi's formula:

$$\widetilde{p}(z) - p(z) = \det(zI - (A + E)) - \det(zI - A) = -\operatorname{tr}(\operatorname{adj}(zI - A)E) + O(\|E\|^2)$$

$$\operatorname{adj}(zI - A) = \sum_{k=0}^{n-1} z^k A_k \text{ (matrix polynomial of degree } n-1)$$

Hence, if we set: $\det(zI - X) = z^n + \sum_{k=0}^{n-1} a_k(X) z^k$, then, to **first order** in E:

$$\boxed{a_k(A+E) - a_k(A) = -\operatorname{tr}(A_k E) = -\operatorname{vec}(A_k{}^\top)^\top \operatorname{vec}(E).}$$

$(\operatorname{vec}(M) := [m_{11} \ldots m_{m1}\ m_{12} \ldots m_{m2} \ldots m_{1n} \ldots m_{mn}]^\top)$

Perturbation of the characteristic polynomial: first order term

Using Jacobi's formula:

$$\widetilde{p}(z) - p(z) = \det(zI - (A+E)) - \det(zI - A) = -\,\mathrm{tr}(\mathrm{adj}(zI - A)E) + O(\|E\|^2)$$

$$\mathrm{adj}(zI - A) = \sum_{k=0}^{n-1} z^k A_k \text{ (matrix polynomial of degree } n-1)$$

Hence, if we set: $\det(zI - X) = z^n + \sum_{k=0}^{n-1} a_k(X) z^k$, then, to **first order** in E:

$$a_k(A+E) - a_k(A) = -\mathrm{tr}(A_k E) = -\mathrm{vec}({A_k}^\top)^\top \mathrm{vec}(E).$$

$(\mathrm{vec}(M) := [m_{11} \dots m_{m1}\ m_{12} \dots m_{m2} \dots m_{1n} \dots m_{mn}]^\top)$

Perturbation of the characteristic polynomial: first order term

Using Jacobi's formula:

$$\widetilde{p}(z) - p(z) = \det(zI - (A+E)) - \det(zI - A) = -\,\mathrm{tr}(\mathrm{adj}(zI - A)E) + O(\|E\|^2)$$

$$\mathrm{adj}(zI - A) = \sum_{k=0}^{n-1} z^k A_k \text{ (matrix polynomial of degree } n-1)$$

Hence, if we set: $\det(zI - X) = z^n + \sum_{k=0}^{n-1} a_k(X) z^k$, then, to **first order** in E:

$$\boxed{a_k(A+E) - a_k(A) = -\mathrm{tr}(A_k E) = -\mathrm{vec}(A_k{}^\top)^\top \mathrm{vec}(E).}$$

$(\mathrm{vec}(M) := [m_{11} \dots m_{m1}\, m_{12} \dots m_{m2} \dots m_{1n} \dots m_{mn}]^\top)$

- Formula for $[(M_\sigma)_k]_{ij}$ **??**

Perturbation of the characteristic polynomial: first order term

Using Jacobi's formula:

$$\widetilde{p}(z) - p(z) = \det(zI - (A+E)) - \det(zI - A) = -\,\mathrm{tr}(\mathrm{adj}(zI - A)E) + O(\|E\|^2)$$

$$\mathrm{adj}(zI - A) = \sum_{k=0}^{n-1} z^k A_k \text{ (matrix polynomial of degree } n-1)$$

Hence, if we set: $\det(zI - X) = z^n + \sum_{k=0}^{n-1} a_k(X) z^k$, then, to **first order** in E:

$$\boxed{a_k(A+E) - a_k(A) = -\mathrm{tr}(A_k E) = -\mathrm{vec}({A_k}^\top)^\top \mathrm{vec}(E).}$$

$(\mathrm{vec}(M) := [m_{11} \dots m_{m1}\, m_{12} \dots m_{m2} \dots m_{1n} \dots m_{mn}]^\top)$

- Formula for $[(M_\sigma)_k]_{ij}$: Is a polynomial on a_i with degree ≤ 2

Explicit formula for the adjugate matrix

Theorem

$\mathrm{PCIS}(\sigma) = (v_0, v_1, \ldots, v_{n-1})$. The (**nonzero**) kth coefficients of the (j,i) entry of $\mathrm{adj}(zI - M_\sigma)$ are:

(a) if $v_{n-i} = v_{n-j} = 0$:

- $a_{k+\mathfrak{i}_\sigma(n-j:n-i)}$, if $j \geq i$ and $n-k-i+1 \leq \mathfrak{i}_\sigma(n-j:n-i) \leq n-k$;
- $-a_{k+1-\mathfrak{i}_\sigma(n-i:n-j-1)}$, if $j < i$ and $k+1+i-n \leq \mathfrak{i}_\sigma(n-i:n-j-1) \leq k+1$;

(b) if $v_{n-i} = v_{n-j} = 1$:

- $a_{k+\mathfrak{c}_\sigma(n-i:n-j)}$, if $j \leq i$ and $n-k-j+1 \leq \mathfrak{c}_\sigma(n-i:n-j) \leq n-k$;
- $-a_{k+1-\mathfrak{c}_\sigma(n-j:n-i-1)}$, if $j > i$ and $k+1+j-n \leq \mathfrak{c}_\sigma(n-j:n-i-1) \leq k+1$;

(c) if $v_{n-i} = 1$ and $v_{n-j} = 0$:

- 1, if $\mathfrak{i}_\sigma(0:n-j-1) + \mathfrak{c}_\sigma(0:n-i-1) = k$,

(d) if $v_{n-i} = 0$ and $v_{n-j} = 1$:

- $$\sum_{l=\max\{0,k+1+j-\mathfrak{c}_\sigma(n-j:n-i-1)-n\}}^{l=\min\{k+1-\mathfrak{c}_\sigma(n-j:n-i-1),i-1\}} -(a_{n+1-i+l}\, a_{k+1-\mathfrak{c}_\sigma(n-j:n-i-1)-l}),$$
  if $j > i$ and $k+2+j-i-n \leq \mathfrak{c}_\sigma(n-j:n-i-1) \leq k+1$;
- $$\sum_{l=\max\{0,k+1+i-\mathfrak{i}_\sigma(n-i:n-j-1)-n\}}^{l=\min\{k+1-\mathfrak{i}_\sigma(n-i:n-j-1),j-1\}} -(a_{n+1-j+l}\, a_{k+1-\mathfrak{i}_\sigma(n-i:n-j-1)-l}),$$
  if $j < i$ and $k+2+i-j-n \leq \mathfrak{i}_\sigma(n-i:n-j-1) \leq k+1$;

(where we set $a_n := 1$, and $v_{n-1} = v_{n-2}$).

Formula for the adjugate: main features

To first order in E:

$$a_k(A+E) - a_k(A) = -\sum_{i,j=1}^{n} p_{ij}^{(\sigma,k)}(a_0, a_1, \ldots, a_{n-1})\, E_{ij}, \qquad k = 0, 1, \ldots, n-1,$$

where:

- $p_{ij}^{(\sigma,k)}(a_0, a_1, \ldots, a_{n-1})$ is a polynomial in a_i with **degree at most** 2.
- If $M_\sigma = C_1, C_2$, then all $p_{ij}^{(\sigma,k)}(a_0, a_1, \ldots, a_{n-1})$ have **degree** 1.
- If $M_\sigma \neq C_1, C_2$, then there is at least one k and some (i,j) such that $p_{ij}^{(\sigma,k)}(a_0, a_1, \ldots, a_{n-1})$ has **degree** 2.

Formula for the adjugate: main features

To first order in E:

$$a_k(A+E) - a_k(A) = -\sum_{i,j=1}^{n} p_{ij}^{(\sigma,k)}(a_0, a_1, \ldots, a_{n-1})\, E_{ij}, \qquad k = 0, 1, \ldots, n-1,$$

where:

- $p_{ij}^{(\sigma,k)}(a_0, a_1, \ldots, a_{n-1})$ is a polynomial in a_i with **degree at most** 2.
- If $M_\sigma = C_1, C_2$, then all $p_{ij}^{(\sigma,k)}(a_0, a_1, \ldots, a_{n-1})$ have **degree** 1.
- If $M_\sigma \neq C_1, C_2$, then there is at least one k and some (i,j) such that $p_{ij}^{(\sigma,k)}(a_0, a_1, \ldots, a_{n-1})$ has **degree** 2.

Formula for the adjugate: main features

To first order in E:

$$a_k(A+E) - a_k(A) = -\sum_{i,j=1}^{n} p_{ij}^{(\sigma,k)}(a_0, a_1, \dots, a_{n-1})\, E_{ij}, \qquad k = 0, 1, \dots, n-1,$$

where:

- $p_{ij}^{(\sigma,k)}(a_0, a_1, \dots, a_{n-1})$ is a polynomial in a_i with **degree at most** 2.
- If $M_\sigma = C_1, C_2$, then all $p_{ij}^{(\sigma,k)}(a_0, a_1, \dots, a_{n-1})$ have **degree** 1.
- If $M_\sigma \neq C_1, C_2$, then there is at least one k and some (i,j) such that $p_{ij}^{(\sigma,k)}(a_0, a_1, \dots, a_{n-1})$ has **degree** 2.

Recursive formula for the adjugate

$p(z) = z^n + \sum_{k=0}^{n-1} a_k z^k$

Proposition [Gantmacher, 1959]

Set:

$$\begin{cases} A_{n-1} = I, \quad \text{and} \\ A_k = A \cdot A_{k+1} + a_k I, \quad \text{for } k = n-2, \ldots, 1, 0. \end{cases}$$

Then,

$$\operatorname{adj}(zI - A) = \sum_{k=0}^{n-1} z^k A_k.$$

Note:

$$A_{k-1} = p_{n-k}(A) = A^{n-k} + a_{n-1} A^{n-k-1} + \cdots + a_{k+1} A + a_k I.$$

$((n-k)$th Horner shift of $p(z)$ evaluated at $A)$

Hence: $p_{n-k-1}(A)$ encodes the information on the variation $a_k(A+E) - a_k(A)$:

$$a_k(A+E) - a_k(A) = -\sum_{i,j} (p_{n-k-1}(A))_{ji} E_{ij} + O(\|E\|^2).$$

Recursive formula for the adjugate

$p(z) = z^n + \sum_{k=0}^{n-1} a_k z^k$

Proposition [Gantmacher, 1959]

Set:

$$\begin{cases} A_{n-1} = I, \quad \text{and} \\ A_k = A \cdot A_{k+1} + a_k I, \quad \text{for } k = n-2, \ldots, 1, 0. \end{cases}$$

Then,

$$\operatorname{adj}(zI - A) = \sum_{k=0}^{n-1} z^k A_k.$$

Note:

$$A_{k-1} = p_{n-k}(A) = A^{n-k} + a_{n-1} A^{n-k-1} + \cdots + a_{k+1} A + a_k I.$$

($(n-k)$th Horner shift of $p(z)$ evaluated at A)

Hence: $p_{n-k-1}(A)$ encodes the information on the variation $a_k(A+E) - a_k(A)$:

$$a_k(A+E) - a_k(A) = -\sum_{i,j} (p_{n-k-1}(A))_{ji} E_{ij} + O(\|E\|^2).$$

Recursive formula for the adjugate

$p(z) = z^n + \sum_{k=0}^{n-1} a_k z^k$

Proposition [Gantmacher, 1959]

Set:

$$\begin{cases} A_{n-1} = I, \quad \text{and} \\ A_k = A \cdot A_{k+1} + a_k I, \quad \text{for } k = n-2, \ldots, 1, 0. \end{cases}$$

Then,

$$\operatorname{adj}(zI - A) = \sum_{k=0}^{n-1} z^k A_k.$$

Note:

$$A_{k-1} = p_{n-k}(A) = A^{n-k} + a_{n-1} A^{n-k-1} + \cdots + a_{k+1} A + a_k I.$$

$((n-k)$th Horner shift of $p(z)$ evaluated at $A)$

Hence: $p_{n-k-1}(A)$ encodes the information on the variation $a_k(A+E) - a_k(A)$:

$$a_k(A+E) - a_k(A) = -\sum_{i,j} (p_{n-k-1}(A))_{ji} E_{ij} + O(\|E\|^2).$$

Some particular examples

Frobenius companion matrices:

$$p_{n-k-1}(C_1^\top) = p_{n-k-1}(C_2) = \left[\begin{array}{cccc|cccc} 0 & \dots & 0 & & 1 & & & 0 \\ -a_k & & & & a_{n-1} & 1 & & \\ \vdots & \ddots & & & \vdots & a_{n-1} & \ddots & \\ -a_1 & \ddots & -a_k & & a_{k+1} & \vdots & \ddots & 1 \\ -a_0 & \ddots & \vdots & & & a_{k+1} & \ddots & a_{n-1} \\ & \ddots & -a_1 & & & & \ddots & \vdots \\ 0 & & -a_0 & & 0 & & & a_{k+1} \end{array}\right].$$

These are the **only** Fiedler matrices M_σ for which **all** $p_k(M_\sigma)$ have entries of **degree 1** !!!!

Some particular examples

Frobenius companion matrices:

$$p_{n-k-1}(C_1^\top) = p_{n-k-1}(C_2) = \left[\begin{array}{cccc|cccc} 0 & \dots & 0 & & 1 & & & 0 \\ -a_k & & & & a_{n-1} & 1 & & \\ \vdots & \ddots & & & \vdots & a_{n-1} & \ddots & \\ -a_1 & \ddots & -a_k & & a_{k+1} & \vdots & \ddots & 1 \\ -a_0 & \ddots & \vdots & & & a_{k+1} & \ddots & a_{n-1} \\ & \ddots & -a_1 & & & & \ddots & \vdots \\ 0 & & -a_0 & & 0 & & & a_{k+1} \end{array}\right].$$

These are the **only** Fiedler matrices M_σ for which **all** $p_k(M_\sigma)$ have entries of **degree 1** !!!!

Some particular examples (II)

$$F = M_{n-1} \cdots M_2 M_0 M_1$$

$$p_{n-k-1}(F) = \left[\begin{array}{cccc|cccc} 0 & & & & 1 & & & 0 \\ -a_k & & & & a_{n-1} & \ddots & & \vdots \\ \vdots & \ddots & & & \vdots & \ddots & 1 & 0 \\ -a_1 & & -a_k & & a_{k+2} & & a_{n-1} & -a_0 \\ -a_0 & \ddots & \vdots & -a_k & a_{k+1} & \ddots & \vdots & -a_0 a_{n-1} \\ & \ddots & -a_1 & \vdots & & \ddots & a_{k+2} & \vdots \\ & & -a_0 & -a_1 & & & a_{k+1} & -a_0 a_{k+2} \\ & & & 1 & & & & a_{k+1} \end{array}\right], \quad \text{for } k = 0 : n-3,$$

$$p_1(F) = \begin{bmatrix} 0 & & & & & 0 \\ -a_{n-2} & 1 & & & & \\ -a_{n-3} & a_{n-1} & 1 & & & \\ \vdots & & a_{n-1} & \ddots & & \\ \vdots & & & \ddots & 1 & \\ -a_1 & & & & a_{n-1} & -a_0 \\ 1 & & & & 0 & a_{n-1} \end{bmatrix}, \quad \text{and} \quad p_0(F) = I.$$

Outline

Backward error

Theorem

If the roots of $p(z)$ are computed as the e-vals of M_σ with a backward stable algorithm, the computed roots are the exact roots of a polynomial $\widetilde{p}(z)$ with:

(a) If $M_\sigma = C_1, C_2$:

$$\frac{\|\widetilde{p}-p\|_\infty}{\|p\|_\infty} = O(u)\|p\|_\infty,$$

(b) if $M_\sigma \neq C_1, C_2$:

$$\frac{\|\widetilde{p}-p\|_\infty}{\|p\|_\infty} = O(u)\|p\|_\infty^2.$$

(u is the machine precision)

Proof (idea): $|\widetilde{a}_k - a_k| = \left|\sum_{i,j=1}^n p_{ij}^{(\sigma,k)}(a_0,a_1,\ldots,a_{n-1})E_{ij}\right| \leq \sum_{i,j=1}^n \left|p_{ij}^{(\sigma,k)}(a_0,a_1,\ldots,a_{n-1})\right| \cdot |E_{ij}| \leq (\max_{1\leq i,j\leq n}|E_{ij}|)\cdot\left(\sum_{i,j=1}^n |p_{ij}^{(\sigma,k)}(a_0,a_1,\ldots,a_{n-1})|\right)$. Therefore,

$$\|\widetilde{p}-p\|_\infty = \max_{k=0,1,\ldots,n-1}|\widetilde{a}_k - a_k| = O(u)\|M_\sigma\|_\infty\|p\|_\infty^2 = O(u)\|p\|_\infty^3,$$

using: $\max_{i,j=1,2,\ldots,n}|E_{ij}| = O(u)\|M_\sigma\|_\infty$ and $\|M_\sigma\|_\infty = O(1)\|p\|_\infty$ [D., Dopico, Pérez, 2013]. □

Backward error

Theorem

If the roots of $p(z)$ are computed as the e-vals of M_σ with a backward stable algorithm, the computed roots are the exact roots of a polynomial $\widetilde{p}(z)$ with:

(a) If $M_\sigma = C_1, C_2$:

$$\frac{\|\widetilde{p}-p\|_\infty}{\|p\|_\infty} = O(u)\|p\|_\infty,$$

(b) if $M_\sigma \neq C_1, C_2$:

$$\frac{\|\widetilde{p}-p\|_\infty}{\|p\|_\infty} = O(u)\|p\|_\infty^2.$$

(u is the machine precision)

Proof (idea): $|\widetilde{a}_k - a_k| = \left|\sum_{i,j=1}^n p_{ij}^{(\sigma,k)}(a_0,a_1,\ldots,a_{n-1})E_{ij}\right| \leq \sum_{i,j=1}^n \left|p_{ij}^{(\sigma,k)}(a_0,a_1,\ldots,a_{n-1})\right| \cdot |E_{ij}| \leq (\max_{1\leq i,j\leq n}|E_{ij}|)\cdot\left(\sum_{i,j=1}^n |p_{ij}^{(\sigma,k)}(a_0,a_1,\ldots,a_{n-1})|\right)$. Therefore,

$$\|\widetilde{p}-p\|_\infty = \max_{k=0,1,\ldots,n-1}|\widetilde{a}_k - a_k| = O(u)\|M_\sigma\|_\infty\|p\|_\infty^2 = O(u)\|p\|_\infty^3,$$

using: $\max_{i,j=1,2,\ldots,n}|E_{ij}| = O(u)\|M_\sigma\|_\infty$ and $\|M_\sigma\|_\infty = O(1)\|p\|_\infty$ [D., Dopico, Pérez, 2013]. □

Some remarks

(Recall: $\|p\|_\infty \geq 1$, since p is monic).

- For $\|p\|_\infty$ moderate, backward stability of polynomial root-finding is guaranteed using **any Fiedler matrix**.
- Then, particular features of some Fiedler matrices (like low bandwidth) can make them preferable than C_1 and C_2.
- When $\|p\|_\infty$ is large, C_1 and C_2 are expected to give smaller b'err than any other Fiedler.
- **Coefficient-wise backward stability** is **not** guaranteed for any Fiedler matrix, even when $\|p\|_\infty = 1$.
- However, when all $|a_i| = \Theta(1)$ (i.e: moderate and not too close to zero), then: $\max_{k=0,1,\ldots,n-1} \frac{|\widetilde{a}_k - a_k|}{|a_k|} = O(u)$.

Some remarks

(Recall: $\|p\|_\infty \geq 1$, since p is monic).

- For $\|p\|_\infty$ moderate, backward stability of polynomial root-finding is guaranteed using **any Fiedler matrix**.
- Then, particular features of some Fiedler matrices (like low bandwidth) can make them preferable than C_1 and C_2.
- When $\|p\|_\infty$ is large, C_1 and C_2 are expected to give smaller b'err than any other Fiedler.
- **Coefficient-wise backward stability** is **not** guaranteed for any Fiedler matrix, even when $\|p\|_\infty = 1$.
- However, when all $|a_i| = \Theta(1)$ (i.e: moderate and not too close to zero), then: $\max_{k=0,1,\dots,n-1} \frac{|\widetilde{a}_k - a_k|}{|a_k|} = O(u)$.

Some remarks

(Recall: $\|p\|_\infty \geq 1$, since p is monic).

- For $\|p\|_\infty$ moderate, backward stability of polynomial root-finding is guaranteed using **any Fiedler matrix**.
- Then, particular features of some Fiedler matrices (like low bandwidth) can make them preferable than C_1 and C_2.
- When $\|p\|_\infty$ is large, C_1 and C_2 are expected to give smaller b'err than any other Fiedler.
- **Coefficient-wise backward stability** is **not** guaranteed for any Fiedler matrix, even when $\|p\|_\infty = 1$.
- However, when all $|a_i| = \Theta(1)$ (i.e: moderate and not too close to zero), then: $\max_{k=0,1,\dots,n-1} \frac{|\widetilde{a}_k - a_k|}{|a_k|} = O(u)$.

Some remarks

(Recall: $\|p\|_\infty \geq 1$, since p is monic).

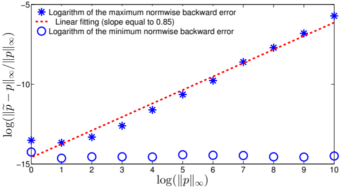
- For $\|p\|_\infty$ moderate, backward stability of polynomial root-finding is guaranteed using **any Fiedler matrix**.
- Then, particular features of some Fiedler matrices (like low bandwidth) can make them preferable than C_1 and C_2.
- When $\|p\|_\infty$ is large, C_1 and C_2 are expected to give smaller b'err than any other Fiedler.
- **Coefficient-wise backward stability** is **not** guaranteed for any Fiedler matrix, even when $\|p\|_\infty = 1$.
- However, when all $|a_i| = \Theta(1)$ (i.e: moderate and not too close to zero), then: $\max_{k=0,1,\ldots,n-1} \frac{|\widetilde{a}_k - a_k|}{|a_k|} = O(u)$.

Some remarks

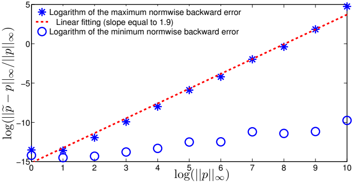
(Recall: $\|p\|_\infty \geq 1$, since p is monic).

- For $\|p\|_\infty$ moderate, backward stability of polynomial root-finding is guaranteed using **any Fiedler matrix**.
- Then, particular features of some Fiedler matrices (like low bandwidth) can make them preferable than C_1 and C_2.
- When $\|p\|_\infty$ is large, C_1 and C_2 are expected to give smaller b'err than any other Fiedler.
- **Coefficient-wise backward stability** is **not** guaranteed for any Fiedler matrix, even when $\|p\|_\infty = 1$.
- However, when all $|a_i| = \Theta(1)$ (i.e: moderate and not too close to zero), then: $\max_{k=0,1,\ldots,n-1} \frac{|\widetilde{a}_k - a_k|}{|a_k|} = O(u)$.

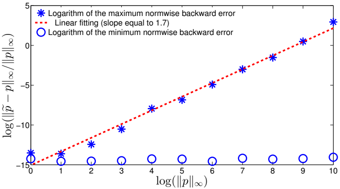
Random polynomials, $n = 20$



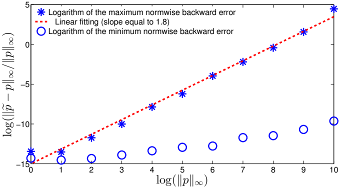
(a) C_2



(b) Pentadiagonal



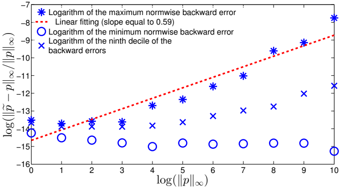
(c) F



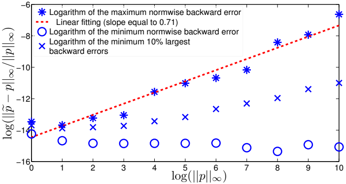
(d) M_σ

Figure: 11 samples, 500 random polys, $\|p\|_\infty = 10^k$ ($k = 0:10$), $a_i = a \cdot 10^c$, $a \in [-1,1]$, $c \in [-k,k]$, $a_0 = 10^k$.

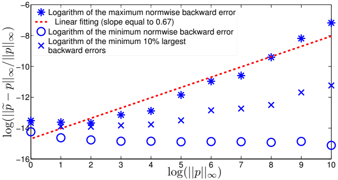
Random polynomials, $n = 20$ (with balancing)



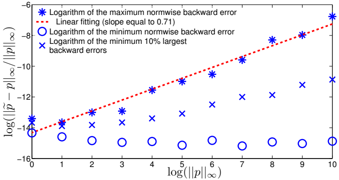
(a) C_2



(b) Pentadiagonal



(c) F



(d) M_σ

Figure: 11 samples, 500 random polys, $\|p\|_\infty = 10^k$ $(k = 0:10)$, $a_i = a \cdot 10^c$, $a \in [-1,1]$, $c \in [-k,k]$, $a_0 = 10^k$.

Outline

Other companion matrices?

Q: Are there any other companion matrices than Fiedlers?

Other companion matrices?

Q: Are there any other companion matrices than Fiedlers?

YES: Infinitely many!

Just multiply: $PM_\sigma P^{-1}$ (P invertible)

Other companion matrices?

Q: Are there any other companion matrices than Fiedlers?

YES: Infinitely many!

Just multiply: $PM_\sigma P^{-1}$ (P invertible)$\rightsquigarrow$ In general, not sparse (**exception**: P is a **permutation** matrix)

☞ We look for sparse companion matrices

Sparse companion matrices (I)

Sparse: It has the smallest number of nonzero entries

☞ For companion matrices, this number is $2n - 1$ [Ma-Zhan'13]

(we focus on: $\underbrace{1, \ldots, 1}_{n-1}, a_0, \ldots, a_{n-1}$)

Q: How many non-permutationally similar sparse companion matrices are there and how do they look like?

Sparse companion matrices (I)

Sparse: It has the smallest number of nonzero entries

☞ For companion matrices, this number is $2n-1$ [Ma-Zhan'13]

(we focus on: $\underbrace{1,\ldots,1}_{n-1}, a_0, \ldots, a_{n-1}$)

Q: How many non-permutationally similar sparse companion matrices are there and how do they look like?

Sparse companion matrices (I)

Sparse: It has the smallest number of nonzero entries

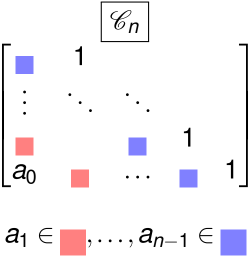
☞ For companion matrices, this number is $2n - 1$ [Ma-Zhan'13]

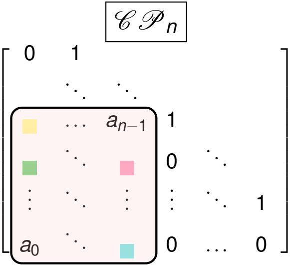
(we focus on: $\underbrace{1, \ldots, 1}_{n-1}, a_0, \ldots, a_{n-1}$)

Q: How many non-permutationally similar sparse companion matrices are there and how do they look like?

Sparse companion matrices (II)

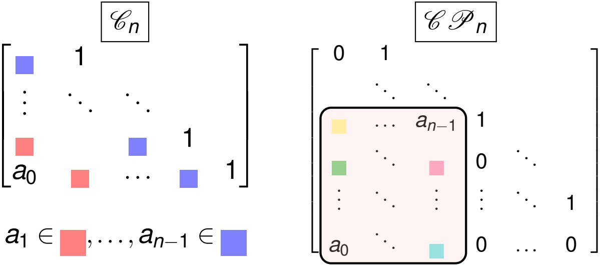
We define the following (lower Hessenberg) classes of matrices:





Sparse companion matrices (II)

We define the following (lower Hessenberg) classes of matrices:

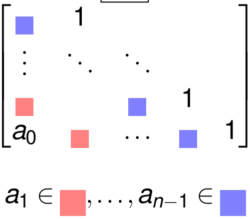


$(\mathscr{C}\mathscr{P}_n \subseteq \mathscr{C}_n)$

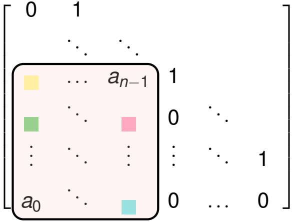
Sparse companion matrices (II)

We define the following (lower Hessenberg) classes of matrices:

$\mathscr{C}_n$



$\mathscr{C}\mathscr{P}_n$



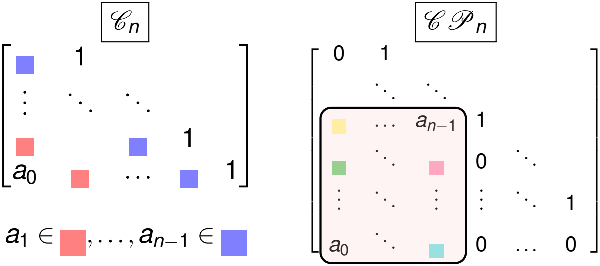
$(\mathscr{C}\mathscr{P}_n \subseteq \mathscr{C}_n)$

Theorem [Eastman-etal'14]

Any sparse companion matrix is permutationally similar to a matrix in $\mathscr{C}_n$.

Sparse companion matrices (II)

We define the following (lower Hessenberg) classes of matrices:



$(\mathcal{CP}_n \subseteq \mathcal{C}_n)$

Theorem [Eastman-etal'14]

Any sparse companion matrix is permutationally similar to a matrix in $\mathcal{C}_n$.

Theorem [Eastman-etal'14]

$A(a_0, \ldots, a_{n-1}) \in \mathcal{C}_n$ is a sparse companion matrix $\Leftrightarrow A(a_0, \ldots, a_{n-1}) \in \mathcal{CP}_n$.

Why monic polynomials?

If $q(z) = a_n z^n + a_{n-1} z^{n-1} + \cdots + a_1 z + a_0$ (not necessarily monic) $(a_n \neq 0)$

Why monic polynomials?

If $q(z) = a_n z^n + a_{n-1} z^{n-1} + \cdots + a_1 z + a_0$ (not necessarily monic) $(a_n \neq 0)$

Then $p(z) = \frac{1}{a_n} q(z) = z^n + \frac{a_{n-1}}{a_n} z^{n-1} + \cdots + \frac{a_1}{a_n} z + \frac{a_0}{a_n}$ is monic

Why monic polynomials?

If $q(z) = a_n z^n + a_{n-1} z^{n-1} + \cdots + a_1 z + a_0$ (not necessarily monic) $(a_n \neq 0)$

Then $p(z) = \frac{1}{a_n} q(z) = z^n + \frac{a_{n-1}}{a_n} z^{n-1} + \cdots + \frac{a_1}{a_n} z + \frac{a_0}{a_n}$ is monic

If the method is b'stable for monic polys:

$$\frac{\|p - \widetilde{p}\|}{\|p\|} = O(u) \qquad \text{(for some } \widetilde{p}\text{).}$$

Why monic polynomials?

If $q(z) = a_n z^n + a_{n-1} z^{n-1} + \cdots + a_1 z + a_0$ (not necessarily monic) $(a_n \neq 0)$

Then $p(z) = \frac{1}{a_n} q(z) = z^n + \frac{a_{n-1}}{a_n} z^{n-1} + \cdots + \frac{a_1}{a_n} z + \frac{a_0}{a_n}$ is monic

If the method is b'stable for monic polys:

$$\frac{\|p - \widetilde{p}\|}{\|p\|} = O(u) \qquad \text{(for some } \widetilde{p}).$$

Then (setting $\widetilde{q} := a_n \widetilde{p}$):

$$\frac{\|q - \widetilde{q}\|}{\|q\|} = \frac{\|\frac{q}{a_n} - \frac{\widetilde{q}}{a_n}\|}{\|\frac{q}{a_n}\|} = \frac{\|p - \widetilde{p}\|}{\|p\|} = O(u).$$

Why monic polynomials?

If $q(z) = a_n z^n + a_{n-1} z^{n-1} + \cdots + a_1 z + a_0$ (not necessarily monic) $(a_n \neq 0)$

Then $p(z) = \frac{1}{a_n} q(z) = z^n + \frac{a_{n-1}}{a_n} z^{n-1} + \cdots + \frac{a_1}{a_n} z + \frac{a_0}{a_n}$ is monic

If the method is b'stable for monic polys:

$$\frac{\|p - \widetilde{p}\|}{\|p\|} = O(u) \qquad \text{(for some } \widetilde{p}\text{)}.$$

Then (setting $\widetilde{q} := a_n \widetilde{p}$):

$$\frac{\|q - \widetilde{q}\|}{\|q\|} = \frac{\|\frac{q}{a_n} - \frac{\widetilde{q}}{a_n}\|}{\|\frac{q}{a_n}\|} = \frac{\|p - \widetilde{p}\|}{\|p\|} = O(u).$$

$\Rightarrow$ It is enough to prove b'stability for monic polys

However...

- B'stability (in the poly sense) is only guaranteed if $\|p\|$ is moderate.
- The QZ algorithm on the Frobenius companion form (non-monic) gives b'stability if $\|p\|_\infty \approx 1$ ([Van Dooren-Dewilde'83]).
- If we divide by $\|p\|_\infty \rightsquigarrow$ the polynomial may become non-monic!

However...

- B'stability (in the poly sense) is only guaranteed if $\|p\|$ is moderate.
- The QZ algorithm on the Frobenius companion form (non-monic) gives b'stability if $\|p\|_\infty \approx 1$ ([Van Dooren-Dewilde'83]).
- If we divide by $\|p\|_\infty \rightsquigarrow$ the polynomial may become non-monic!

However...

- B'stability (in the poly sense) is only guaranteed if $\|p\|$ is moderate.
- The QZ algorithm on the Frobenius companion form (non-monic) gives b'stability if $\|p\|_\infty \approx 1$ ([Van Dooren-Dewilde'83]).
- If we divide by $\|p\|_\infty \rightsquigarrow$ the polynomial may become non-monic!

Fiedler companion forms

Frobenius companion forms

$$F_i(z) = z\,\mathrm{diag}(a_n, 1, \ldots, 1) - C_i \qquad i = 1, 2$$

Fiedler companion forms

Frobenius companion forms

$$F_i(z) = z\,\mathrm{diag}(a_n, 1, \dots, 1) - C_i \qquad i = 1, 2$$

Fiedler companion forms

$$F_\sigma(z) = z\,\mathrm{diag}(a_n, 1, \dots, 1) - M_\sigma$$

Fiedler companion forms

Frobenius companion forms

$$F_i(z) = z\,\mathrm{diag}(a_n, 1, \ldots, 1) - C_i \qquad i = 1, 2$$

Fiedler companion forms

$$F_\sigma(z) = z\,\mathrm{diag}(a_n, 1, \ldots, 1) - M_\sigma$$

Examples: $F_1 = \begin{bmatrix} a_n z + a_{n-1} & a_{n-2} & \cdots & a_0 \\ -1 & z & \cdots & 0 \\ & \ddots & \ddots & \vdots \\ 0 & & -1 & z \end{bmatrix} \qquad F_2 = F_1^\top$

$$F = \begin{bmatrix} a_6 z + a_5 & -1 & 0 & 0 & 0 & 0 \\ a_4 & z & a_3 & -1 & 0 & 0 \\ -1 & 0 & z & 0 & 0 & 0 \\ 0 & 0 & a_2 & z & a_1 & -1 \\ 0 & 0 & -1 & 0 & z & 0 \\ 0 & 0 & 0 & 0 & a_0 & z \end{bmatrix} \qquad (n = 6)$$

Other companion forms

Companion form

A matrix $A(a_0, a_1, \ldots, a_{n-1}, a_n; z)$ such that:

- The entries are linear polynomials in z.
- $\det A(a_0, a_1, \ldots, a_{n-1}, a_n; z) = a_n z^n + a_{n-1} z^{n-1} + \cdots + a_1 z + a_0$

Other companion forms

Companion form

A matrix $A(a_0, a_1, \ldots, a_{n-1}, a_n; z)$ such that:

- The entries are linear polynomials in z.
- $\det A(a_0, a_1, \ldots, a_{n-1}, a_n; z) = a_n z^n + a_{n-1} z^{n-1} + \cdots + a_1 z + a_0$

☞ ~~Similarity~~ Equivalence

Other companion forms

Companion form

A matrix $A(a_0, a_1, \ldots, a_{n-1}, a_n; z)$ such that:

- The entries are linear polynomials in z.
- $\det A(a_0, a_1, \ldots, a_{n-1}, a_n; z) = a_n z^n + a_{n-1} z^{n-1} + \cdots + a_1 z + a_0$

☞ ~~Similarity~~ Equivalence

Fiedler-like:
$$\begin{bmatrix} 0 & 0 & 0 & z & a_0 + za_1 \\ 0 & 0 & 1 & 0 & -z \\ 0 & z & a_2 + za_3 & -1 & 0 \\ 1 & 0 & -z & 0 & 0 \\ a_4 + za_5 & -1 & 0 & 0 & 0 \end{bmatrix} \qquad (n = 5)$$

Other companion forms

Companion form

A matrix $A(a_0, a_1, \dots, a_{n-1}, a_n; z)$ such that:

- The entries are linear polynomials in z.
- $\det A(a_0, a_1, \dots, a_{n-1}, a_n; z) = a_n z^n + a_{n-1} z^{n-1} + \cdots + a_1 z + a_0$

☞ ~~Similarity~~ Equivalence

Fiedler-like: $$\begin{bmatrix} 0 & 0 & 0 & z & a_0 + za_1 \\ 0 & 0 & 1 & 0 & -z \\ 0 & z & a_2 + za_3 & -1 & 0 \\ 1 & 0 & -z & 0 & 0 \\ a_4 + za_5 & -1 & 0 & 0 & 0 \end{bmatrix} \qquad (n = 5)$$

☞ There are many others [Dopico-Lawrence-Pérez-VanDooren]:

- Permutationally equivalent to companion forms in some "extended $\mathscr{C}\mathscr{P}_n$".
- Most of them are not sparse.

Open questions for companion forms

- How many non-permutationally similar companion matrices are there in $\mathscr{C}\mathscr{P}_n$?
- Which is the smallest number of nonzero entries (sparse)?

Open questions for companion forms

- How many non-permutationally similar companion matrices are there in $\mathcal{CP}_n$?
- Which is the smallest number of nonzero entries (sparse)?

$$\begin{bmatrix} 0 & 0 & 0 & z & a_0+za_1 \\ 0 & 0 & 1 & 0 & -z \\ 0 & z & a_2+za_3 & -1 & 0 \\ 1 & 0 & -z & 0 & 0 \\ a_4+za_5 & -1 & 0 & 0 & 0 \end{bmatrix} \qquad \begin{matrix} (n=5, \\ \#(nonzero)=11) \end{matrix}$$

Open questions for companion forms

- How many non-permutationally similar companion matrices are there in $\mathscr{C}\mathscr{P}_n$?
- Which is the smallest number of nonzero entries (sparse)?
- Are all sparse companion forms permutationally equivalent to a companion form in an “extended $\mathscr{C}_n$"?

Open questions for companion forms

- How many non-permutationally similar companion matrices are there in $\mathscr{C}\mathscr{P}_n$?
- Which is the smallest number of nonzero entries (sparse)?
- Are all sparse companion forms permutationally equivalent to a companion form in an "extended $\mathscr{C}_n$"?
- Do all sparse companion forms in this $\mathscr{C}_n$ belong to an "extended $\mathscr{C}\mathscr{P}_n$"?

Open questions for companion forms

- How many non-permutationally similar companion matrices are there in $\mathscr{CP}_n$?
- Which is the smallest number of nonzero entries (sparse)?
- Are all sparse companion forms permutationally equivalent to a companion form in an “extended $\mathscr{C}_n$"?
- Do all sparse companion forms in this $\mathscr{C}_n$ belong to an “extended $\mathscr{CP}_n$"?
- Is there any companion form that provides a smaller b’err than Frobenius ones?

Outline

Conclusions

- B'stability on the e-val problem $\not\Rightarrow$ B'stability on the poly root-finding problem.
- When $\|p\|_\infty$ is moderate, a b'stable e-val algorithm implies poly b'stability for any Fiedler matrix.
- When $\|p\|_\infty$ is large, Frobenius companion matrices are expected to give **less b'err** than any other Fiedlers.
- Though `roots` is b'stable in practice, it could give non-satisfactory results.
- Characterization of all sparse companion **matrices** is known (only for monic polynomials!).
- Looking at monic polynomials is **not enough** to guarantee b'stability.
- Still more room to look for other companion **forms** and to describe all sparse ones.

Conclusions

- B'stability on the e-val problem $\not\Rightarrow$ B'stability on the poly root-finding problem.
- When $\|p\|_\infty$ is moderate, a b'stable e-val algorithm implies poly b'stability for any Fiedler matrix.
- When $\|p\|_\infty$ is large, Frobenius companion matrices are expected to give **less b'err** than any other Fiedlers.
- Though `roots` is b'stable in practice, it could give non-satisfactory results.
- Characterization of all sparse companion **matrices** is known (only for monic polynomials!).
- Looking at monic polynomials is **not enough** to guarantee b'stability.
- Still more room to look for other companion **forms** and to describe all sparse ones.

Conclusions

- B'stability on the e-val problem $\not\Rightarrow$ B'stability on the poly root-finding problem.
- When $\|p\|_\infty$ is moderate, a b'stable e-val algorithm implies poly b'stability for any Fiedler matrix.
- When $\|p\|_\infty$ is large, Frobenius companion matrices are expected to give **less b'err** than any other Fiedlers.
- Though `roots` is b'stable in practice, it could give non-satisfactory results.
- Characterization of all sparse companion **matrices** is known (only for monic polynomials!).
- Looking at monic polynomials is **not enough** to guarantee b'stability.
- Still more room to look for other companion **forms** and to describe all sparse ones.

Conclusions

- B'stability on the e-val problem $\not\Rightarrow$ B'stability on the poly root-finding problem.
- When $\|p\|_\infty$ is moderate, a b'stable e-val algorithm implies poly b'stability for any Fiedler matrix.
- When $\|p\|_\infty$ is large, Frobenius companion matrices are expected to give **less b'err** than any other Fiedlers.
- Though `roots` is b'stable in practice, it could give non-satisfactory results.
- Characterization of all sparse companion **matrices** is known (only for monic polynomials!).
- Looking at monic polynomials is **not enough** to guarantee b'stability.
- Still more room to look for other companion **forms** and to describe all sparse ones.

Conclusions

- B'stability on the e-val problem $\not\Rightarrow$ B'stability on the poly root-finding problem.
- When $\|p\|_\infty$ is moderate, a b'stable e-val algorithm implies poly b'stability for any Fiedler matrix.
- When $\|p\|_\infty$ is large, Frobenius companion matrices are expected to give **less b'err** than any other Fiedlers.
- Though `roots` is b'stable in practice, it could give non-satisfactory results.
- Characterization of all sparse companion **matrices** is known (only for monic polynomials!).
- Looking at monic polynomials is **not enough** to guarantee b'stability.
- Still more room to look for other companion **forms** and to describe all sparse ones.

Conclusions

- B'stability on the e-val problem $\not\Rightarrow$ B'stability on the poly root-finding problem.
- When $\|p\|_\infty$ is moderate, a b'stable e-val algorithm implies poly b'stability for any Fiedler matrix.
- When $\|p\|_\infty$ is large, Frobenius companion matrices are expected to give **less b'err** than any other Fiedlers.
- Though `roots` is b'stable in practice, it could give non-satisfactory results.
- Characterization of all sparse companion **matrices** is known (only for monic polynomials!).
- Looking at monic polynomials is **not enough** to guarantee b'stability.
- Still more room to look for other companion **forms** and to describe all sparse ones.

Conclusions

- B'stability on the e-val problem $\not\Rightarrow$ B'stability on the poly root-finding problem.
- When $\|p\|_\infty$ is moderate, a b'stable e-val algorithm implies poly b'stability for any Fiedler matrix.
- When $\|p\|_\infty$ is large, Frobenius companion matrices are expected to give **less b'err** than any other Fiedlers.
- Though `roots` is b'stable in practice, it could give non-satisfactory results.
- Characterization of all sparse companion **matrices** is known (only for monic polynomials!).
- Looking at monic polynomials is **not enough** to guarantee b'stability.
- Still more room to look for other companion **forms** and to describe all sparse ones.

Aurentz, Mach, Vandebril, Watkins.
Fast and stable computation of roots of polynomials.
SIAM J. Matrix Anal. Appl., 36 (2015) 942–973.

DT., Dopico, Pérez.
Backward stability of polynomial root-finding using Fiedler companion matrices.
IMA J. Numer. Analysis, 36 (2016) 133–173.

Eastman, Kim, Shader, Vander Meulen.
Companion matrix patterns.
Linear Algebra Appl., 463 (2014) 255–272.

Edelman, Murakami.
Polynomial roots from companion matrix eigenvalues.
Math. Comp., 64 (1995) 763–776.

Fiedler.
A note on companion matrices.
Linear Algebra Appl., 372 (2003) 325–331.

Ma, Zhan.
Extremal sparsity of the companion matrix of a polynomial.
Linear Algebra Appl., 438 (2013) 621–625.

Van Dooren, Dewilde.
The eigenstructure of an arbitrary polynomial matrix: computational aspects
Linear Algebra Appl., 50 (1983) 545–579.

DANKE !!