Backward stability of polynomial root-finding using Fiedler companion matrices and pencils

Froilán M. Dopico
joint work with
F. De Terán (UC3M), J. Pérez (U. Manchester, UK),
P. Lawrence (KU Leuven, Belgium), P. Van Dooren (UC Louvain, Belgium)

Department of Mathematics, Universidad Carlos III de Madrid, Spain

Talk at Departamento de Matemáticas, Estadística y Computación
Universidad de Cantabria. Santander
November 16, 2015

Introduction (I): The problem and the rules

- To compute all the roots of a scalar polynomial

$$q(z) = b_n\, z^n + b_{n-1}\, z^{n-1} + \cdots + b_1\, z + \, b_0, \quad b_i \in \mathbb{C}\,,$$

- with an algorithm which uses only floating point arithmetic (with unit roundoff u, $u \approx 10^{-16}$ in IEEE double precision),
- is efficient, that is, it has cost at most $O(n^3)$ operations (flops) and ideally much less, and
- enjoys guaranteed backward stability.

- To compute all the roots of a scalar polynomial

$$q(z) = b_n\, z^n + b_{n-1}\, z^{n-1} + \cdots + b_1\, z + \, b_0, \quad b_i \in \mathbb{C}\,,$$

- with an algorithm which uses only floating point arithmetic (with unit roundoff u, $u \approx 10^{-16}$ in IEEE double precision),
- is efficient, that is, it has cost at most $O(n^3)$ operations (flops) and ideally much less, and
- enjoys guaranteed backward stability.

- To compute all the roots of a scalar polynomial

$$q(z) = b_n \, z^n + b_{n-1} \, z^{n-1} + \cdots + b_1 \, z + \, b_0, \quad b_i \in \mathbb{C} \, ,$$

- with an algorithm which uses only floating point arithmetic (with unit roundoff u, $u \approx 10^{-16}$ in IEEE double precision),
- is efficient, that is, it has cost at most $O(n^3)$ operations (flops) and ideally much less, and
- enjoys guaranteed backward stability.

- To compute all the roots of a scalar polynomial

$$q(z) = b_n\, z^n + b_{n-1}\, z^{n-1} + \cdots + b_1\, z + b_0, \quad b_i \in \mathbb{C}\,,$$

- with an algorithm which uses only floating point arithmetic (with unit roundoff u, $u \approx 10^{-16}$ in IEEE double precision),
- is efficient, that is, it has cost at most $O(n^3)$ operations (flops) and ideally much less, and
- enjoys guaranteed backward stability.

Problem: Compute all the roots of a scalar polynomial

$$q(z) = b_n\, z^n + b_{n-1}\, z^{n-1} + \cdots + b_1\, z + b_0, \quad b_i \in \mathbb{C}.$$

- Loosely speaking: the computed roots are the exact roots of a nearby polynomial

$$\widetilde{q}(z) = \widetilde{b}_n\, z^n + \widetilde{b}_{n-1}\, z^{n-1} + \cdots + \widetilde{b}_1\, z + \widetilde{b}_0, \quad \widetilde{b}_i \in \mathbb{C}\,.$$

- **Rigorous meaning**:

  1. The whole ensemble of computed roots is the whole ensemble of roots of $\widetilde{q}(z)$ and

$$\|q(z) - \widetilde{q}(z)\|_\infty = O(u)\, \|q(z)\|_\infty,$$

  where $\|q(z)\|_\infty := \max\{|b_n|, |b_{n-1}|, \ldots, |b_1|, |b_0|\}$, and the constant involved in $O(u)$ is a moderate low degree polynomial in n.

What does "guaranteed backward stability" mean? (I)

Problem: Compute all the roots of a scalar polynomial

$$q(z) = b_n\, z^n + b_{n-1}\, z^{n-1} + \cdots + b_1\, z + b_0, \quad b_i \in \mathbb{C}.$$

- Loosely speaking: the computed roots are the exact roots of a nearby polynomial

$$\widetilde{q}(z) = \widetilde{b}_n\, z^n + \widetilde{b}_{n-1}\, z^{n-1} + \cdots + \widetilde{b}_1\, z + \widetilde{b}_0, \quad \widetilde{b}_i \in \mathbb{C}\,.$$

- **Rigorous meaning**:

  1. The whole ensemble of computed roots is the whole ensemble of roots of $\widetilde{q}(z)$ and

$$\|q(z) - \widetilde{q}(z)\|_\infty = O(u)\, \|q(z)\|_\infty,$$

  where $\|q(z)\|_\infty := \max\{|b_n|, |b_{n-1}|, \ldots, |b_1|, |b_0|\}$, and the constant involved in $O(u)$ is a moderate low degree polynomial in n.

What does "guaranteed backward stability" mean? (II)

Problem: Compute all the roots of a scalar polynomial

$$q(z) = b_n\, z^n + b_{n-1}\, z^{n-1} + \cdots + b_1\, z + b_0, \quad b_i \in \mathbb{C}\,.$$

Several other possible **rigorous meanings (not used in this talk)**:

2. The whole ensemble of computed roots is the whole ensemble of roots of $\widetilde{q}(z) = \widetilde{b}_n\, z^n + \widetilde{b}_{n-1}\, z^{n-1} + \cdots + \widetilde{b}_1\, z + \widetilde{b}_0$ and

   $$|b_i - \widetilde{b}_i| = O(u)\, |b_i|, \quad i = 1, \ldots, n\,.$$

   It has been proved by *Mastronardi and Van Dooren, ETNA, 2015* that there does not exist any algorithm that get this coefficient-wise backward stability for quadratic polynomials $\longrightarrow$ too strict!!

3. Each computed root $\widehat{\lambda}$ is the exact root of a nearby polynomial $\widetilde{q}_\lambda(z)$ (different for each $\widehat{\lambda}$!!!!) and

   $$|b_i - (\widetilde{b}_\lambda)_i| = O(u)\, |b_i|, \quad i = 1, \ldots, n\,.$$

   At present, algorithms with this type of coefficient-wise backward stability are only known for cubic polynomials (Su, Lu, ICIAM, 2015).

Problem: Compute all the roots of a scalar polynomial

$$q(z) = b_n\, z^n + b_{n-1}\, z^{n-1} + \cdots + b_1\, z + b_0, \quad b_i \in \mathbb{C}\,.$$

Several other possible **rigorous meanings (not used in this talk)**:

2. The whole ensemble of computed roots is the whole ensemble of roots of $\widetilde{q}(z) = \widetilde{b}_n\, z^n + \widetilde{b}_{n-1}\, z^{n-1} + \cdots + \widetilde{b}_1\, z + \widetilde{b}_0$ and

$$|b_i - \widetilde{b}_i| = O(u)\,|b_i|, \quad i = 1, \ldots, n\,.$$

   It has been proved by *Mastronardi and Van Dooren, ETNA, 2015* that there does not exist any algorithm that get this coefficient-wise backward stability for quadratic polynomials $\longrightarrow$ too strict!!

3. Each computed root $\widehat{\lambda}$ is the exact root of a nearby polynomial $\widetilde{q}_\lambda(z)$ (different for each $\widehat{\lambda}$!!!!) and

$$|b_i - (\widetilde{b}_\lambda)_i| = O(u)\,|b_i|, \quad i = 1, \ldots, n\,.$$

   At present, algorithms with this type of coefficient-wise backward stability are only known for cubic polynomials (Su, Lu, ICIAM, 2015).

What does "guaranteed backward stability" mean? (II)

Problem: Compute all the roots of a scalar polynomial

$$q(z) = b_n\, z^n + b_{n-1}\, z^{n-1} + \cdots + b_1\, z + b_0, \quad b_i \in \mathbb{C}\,.$$

Several other possible **rigorous meanings (not used in this talk)**:

2. The whole ensemble of computed roots is the whole ensemble of roots of $\widetilde{q}(z) = \widetilde{b}_n\, z^n + \widetilde{b}_{n-1}\, z^{n-1} + \cdots + \widetilde{b}_1\, z + \widetilde{b}_0$ and

$$|b_i - \widetilde{b}_i| = O(u)\,|b_i|, \quad i = 1, \ldots, n\,.$$

It has been proved by *Mastronardi and Van Dooren, ETNA, 2015* that there does not exist any algorithm that get this coefficient-wise backward stability for quadratic polynomials $\longrightarrow$ too strict!!

3. Each computed root $\widehat{\lambda}$ is the exact root of a nearby polynomial $\widetilde{q}_\lambda(z)$ (different for each $\widehat{\lambda}$!!!!) and

$$|b_i - (\widetilde{b}_\lambda)_i| = O(u)\,|b_i|, \quad i = 1, \ldots, n\,.$$

At present, algorithms with this type of coefficient-wise backward stability are only known for cubic polynomials (Su, Lu, ICIAM, 2015).

Problem: Compute all the roots of a scalar polynomial

$$q(z) = b_n\, z^n + b_{n-1}\, z^{n-1} + \cdots + b_1\, z + b_0, \quad b_i \in \mathbb{C}\,.$$

Several other possible **rigorous meanings (not used in this talk)**:

2. The whole ensemble of computed roots is the whole ensemble of roots of $\widetilde{q}(z) = \widetilde{b}_n\, z^n + \widetilde{b}_{n-1}\, z^{n-1} + \cdots + \widetilde{b}_1\, z + \widetilde{b}_0$ and

   $$|b_i - \widetilde{b}_i| = O(u)\,|b_i|, \quad i = 1, \ldots, n\,.$$

   It has been proved by *Mastronardi and Van Dooren, ETNA, 2015* that there does not exist any algorithm that get this coefficient-wise backward stability for quadratic polynomials $\longrightarrow$ too strict!!

3. Each computed root $\widehat{\lambda}$ is the exact root of a nearby polynomial $\widetilde{q}_\lambda(z)$ (different for each $\widehat{\lambda}$!!!!) and

   $$|b_i - (\widetilde{b}_\lambda)_i| = O(u)\,|b_i|, \quad i = 1, \ldots, n\,.$$

   At present, algorithms with this type of coefficient-wise backward stability are only known for cubic polynomials (Su, Lu, ICIAM, 2015).

How does MATLAB compute all the roots of a polynomial? (simplified)

- **Step 1.** Make the polynomial monic

$$q(z) = b_n\, z^n + b_{n-1}\, z^{n-1} + \cdots + b_1\, z + b_0 \longrightarrow p(z) := q(z)/b_n\,.$$

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

- **Step 2.** Construct the first Frobenius Companion matrix of $p(z)$

$$C = \begin{bmatrix} -a_{n-1} & \cdots & -a_1 & -a_0 \\ 1 & & & \\ & \ddots & & \\ & & 1 & \end{bmatrix} \in \mathbb{C}^{n\times n}\,.$$

- **Step 3.** Compute all the eigenvalues of C using the Francis-QR algorithm.

Remark: C is the best known example of a companion matrix of $p(z)$, that is, a matrix easily constructible from $p(z)$ and whose characteristic polynomial is $p(z)$. There are many other companion matrices, some developed very recently.

How does MATLAB compute all the roots of a polynomial? (simplified)

- **Step 1.** Make the polynomial monic

$$q(z) = b_n\, z^n + b_{n-1}\, z^{n-1} + \cdots + b_1\, z + b_0 \longrightarrow p(z) := q(z)/b_n\,.$$

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

- **Step 2.** Construct the first Frobenius Companion matrix of $p(z)$

$$C = \begin{bmatrix} -a_{n-1} & \cdots & -a_1 & -a_0 \\ 1 & & & \\ & \ddots & & \\ & & 1 & \end{bmatrix} \in \mathbb{C}^{n\times n}\,.$$

- **Step 3.** Compute all the eigenvalues of C using the Francis-QR algorithm.

Remark: C is the best known example of a companion matrix of $p(z)$, that is, a matrix easily constructible from $p(z)$ and whose characteristic polynomial is $p(z)$. There are many other companion matrices, some developed very recently.

How does MATLAB compute all the roots of a polynomial? (simplified)

- **Step 1.** Make the polynomial monic

$$q(z) = b_n\, z^n + b_{n-1}\, z^{n-1} + \cdots + b_1\, z + b_0 \longrightarrow p(z) := q(z)/b_n\,.$$

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

- **Step 2.** Construct the first Frobenius Companion matrix of $p(z)$

$$C = \begin{bmatrix} -a_{n-1} & \cdots & -a_1 & -a_0 \\ 1 & & & \\ & \ddots & & \\ & & 1 & \end{bmatrix} \in \mathbb{C}^{n\times n}\,.$$

- **Step 3.** Compute all the eigenvalues of C using the Francis-QR algorithm.

Remark: C is the best known example of a companion matrix of $p(z)$, that is, a matrix easily constructible from $p(z)$ and whose characteristic polynomial is $p(z)$. There are many other companion matrices, some developed very recently.

- **Step 1.** Make the polynomial monic

$$q(z) = b_n\, z^n + b_{n-1}\, z^{n-1} + \cdots + b_1\, z + b_0 \longrightarrow p(z) := q(z)/b_n\,.$$

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

- **Step 2.** Construct the first Frobenius Companion matrix of $p(z)$

$$C = \begin{bmatrix} -a_{n-1} & \cdots & -a_1 & -a_0 \\ 1 & & & \\ & \ddots & & \\ & & 1 & \end{bmatrix} \in \mathbb{C}^{n\times n}\,.$$

- **Step 3.** Compute all the eigenvalues of C using the Francis-QR algorithm.

Remark: C is the best known example of a companion matrix of $p(z)$, that is, a matrix easily constructible from $p(z)$ and whose characteristic polynomial is $p(z)$. There are many other companion matrices, some developed very recently.

- **Step 1.** Make the polynomial monic

$$q(z) = b_n\, z^n + b_{n-1}\, z^{n-1} + \cdots + b_1\, z + b_0 \longrightarrow p(z) := q(z)/b_n\,.$$

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

- **Step 2.** Construct the first Frobenius Companion matrix of $p(z)$

$$C = \begin{bmatrix} -a_{n-1} & \cdots & -a_1 & -a_0 \\ 1 & & & \\ & \ddots & & \\ & & 1 & \end{bmatrix} \in \mathbb{C}^{n\times n}\,.$$

- **Step 3.** Compute all the eigenvalues of C using the Francis-QR algorithm.

Remark: C is the best known example of a companion matrix of $p(z)$, that is, a matrix easily constructible from $p(z)$ and whose characteristic polynomial is $p(z)$. There are many other companion matrices, some developed very recently.

Compute with Francis-QR algorithm all the eigenvalues of

$$C = \begin{bmatrix} -a_{n-1} & \cdots & -a_1 & -a_0 \\ 1 & & & \\ & \ddots & & \\ & & 1 & \end{bmatrix} \in \mathbb{C}^{n\times n}.$$

- $O(n^3)$ computational cost and $O(n^2)$ storage for only n input data.
- TOO MUCH!!, though in practice MATLAB covers most of the interesting cases, since the degrees often are not huge.
- **Long time dream** started by C. Moler, Mathworks Newsletter, (1991):

  > *"An algorithm designed specifically for polynomial roots might use order n storage and n^2 time"*

  and (community adds) to be as stable as MATLAB's command `roots`.
- After many tries the dream has been realized by Aurentz, Mach, Vandebril, Watkins, SIMAX, 2015 via a highly structured version of Francis-QR algorithm adapted to C (Will it be in MATLAB??).

Main drawback of MATLAB's approach

Compute with Francis-QR algorithm all the eigenvalues of

$$C = \begin{bmatrix} -a_{n-1} & \cdots & -a_1 & -a_0 \\ 1 & & & \\ & \ddots & & \\ & & 1 & \end{bmatrix} \in \mathbb{C}^{n\times n}.$$

- $O(n^3)$ computational cost and $O(n^2)$ storage for only n input data.
- TOO MUCH!!, though in practice MATLAB covers most of the interesting cases, since the degrees often are not huge.
- **Long time dream** started by C. Moler, Mathworks Newsletter, (1991):

  "An algorithm designed specifically for polynomial roots might use order n storage and n^2 time"

  and (community adds) to be as stable as MATLAB's command `roots`.
- After many tries the dream has been realized by Aurentz, Mach, Vandebril, Watkins, SIMAX, 2015 via a highly structured version of Francis-QR algorithm adapted to C (Will it be in MATLAB??).

Main drawback of MATLAB's approach

Compute with Francis-QR algorithm all the eigenvalues of

$$C = \begin{bmatrix} -a_{n-1} & \cdots & -a_1 & -a_0 \\ 1 & & & \\ & \ddots & & \\ & & 1 & \end{bmatrix} \in \mathbb{C}^{n\times n}.$$

- $O(n^3)$ computational cost and $O(n^2)$ storage for only n input data.
- TOO MUCH!!, though in practice MATLAB covers most of the interesting cases, since the degrees often are not huge.
- **Long time dream** started by C. Moler, Mathworks Newsletter, (1991):

  "An algorithm designed specifically for polynomial roots might use order n storage and n^2 time"

  and (community adds) to be as stable as MATLAB's command `roots`.
- After many tries the dream has been realized by Aurentz, Mach, Vandebril, Watkins, SIMAX, 2015 via a highly structured version of Francis-QR algorithm adapted to C (Will it be in MATLAB??).

Main drawback of MATLAB's approach

Compute with Francis-QR algorithm all the eigenvalues of

$$C = \begin{bmatrix} -a_{n-1} & \cdots & -a_1 & -a_0 \\ 1 & & & \\ & \ddots & & \\ & & 1 & \end{bmatrix} \in \mathbb{C}^{n\times n}.$$

- $O(n^3)$ computational cost and $O(n^2)$ storage for only n input data.
- TOO MUCH!!, though in practice MATLAB covers most of the interesting cases, since the degrees often are not huge.
- **Long time dream** started by C. Moler, Mathworks Newsletter, (1991):

  "An algorithm designed specifically for polynomial roots might use order n storage and n^2 time"

  and (community adds) to be as stable as MATLAB's command `roots`.
- After many tries the dream has been realized by Aurentz, Mach, Vandebril, Watkins, SIMAX, 2015 via a highly structured version of Francis-QR algorithm adapted to C (Will it be in MATLAB??).

Compute with Francis-QR algorithm all the eigenvalues of

$$C = \begin{bmatrix} -a_{n-1} & \cdots & -a_1 & -a_0 \\ 1 & & & \\ & \ddots & & \\ & & 1 & \end{bmatrix} \in \mathbb{C}^{n \times n}.$$

- $O(n^3)$ computational cost and $O(n^2)$ storage for only n input data.
- TOO MUCH!!, though in practice MATLAB covers most of the interesting cases, since the degrees often are not huge.
- **Long time dream** started by C. Moler, Mathworks Newsletter, (1991):

  "An algorithm designed specifically for polynomial roots might use order n storage and n^2 time"

  and (community adds) to be as stable as MATLAB's command `roots`.
- After many tries the dream has been realized by Aurentz, Mach, Vandebril, Watkins, SIMAX, 2015 via a highly structured version of Francis-QR algorithm adapted to C (Will it be in MATLAB??).

Reliability in two senses.

1. **Francis QR-algorithm is extremely robust**. It enjoys "guaranteed practical" convergence for all eigenvalues (roots).
2. **Francis QR-algorithm is extremely stable**. It enjoys **perfect MATRIX backward stability**, that is, the computed roots of $p(z)$ are the exact eigenvalues of

$$C + E, \quad \text{with} \quad \|E\|_2 = O(u)\|C\|_2,$$

where $u (\approx 10^{-16})$ is the unit roundoff.

- What kind of **polynomial backward stability** is provided by this **perfect matrix backward stability**?
- Note that for our monic poly $p(z) = z^n + a_{n-1}\, z^{n-1} + \cdots + a_1\, z + \, a_0$,

$$c_n \, \|C\|_2 \leq \|p\|_\infty \leq d_n \, \|C\|_2,$$

  for c_n, d_n low powers of n (note also $\|p\|_\infty \geq 1$).
- So, **MATLAB computed roots of $p(z)$ are** the exact eigenvalues of

$$C + E, \quad \text{with} \quad \|E\|_2 = O(u)\|C\|_2 = O(u)\|p\|_\infty,$$

  or **the exact roots of**

$$\widetilde{p}(z) = \det(zI - (C + E)).$$

- Van Dooren & DeWilde (1983), Edelman & Murakami (1995), Lemmonier & Van Dooren (2003) proved

$$\widetilde{p}(z) = p(z) + e(z), \quad \text{with} \quad \|e(z)\|_\infty = O(u)\|p(z)\|_\infty^2,$$

- which means that **perfect matrix backward stability DOES NOT** imply perfect **polynomial backward stability** $\Longrightarrow$ **there is a penalty.**

Is this "the stability desired" for polynomial root-finding?

- What kind of **polynomial backward stability** is provided by this **perfect matrix backward stability**?
- Note that for our monic poly $p(z) = z^n + a_{n-1}\, z^{n-1} + \cdots + a_1\, z + \, a_0$,

$$c_n \, \|C\|_2 \leq \|p\|_\infty \leq d_n \, \|C\|_2,$$

for c_n, d_n low powers of n (note also $\|p\|_\infty \geq 1$).

- So, **MATLAB computed roots of** $p(z)$ **are** the exact eigenvalues of

$$C + E, \quad \text{with} \quad \|E\|_2 = O(u)\|C\|_2 = O(u)\|p\|_\infty,$$

or **the exact roots of**

$$\widetilde{p}(z) = \det(zI - (C + E)).$$

- Van Dooren & DeWilde (1983), Edelman & Murakami (1995), Lemmonier & Van Dooren (2003) proved

$$\widetilde{p}(z) = p(z) + e(z), \quad \text{with} \quad \|e(z)\|_\infty = O(u)\|p(z)\|_\infty^2,$$

- which means that **perfect matrix backward stability DOES NOT** imply perfect **polynomial backward stability** $\Longrightarrow$ **there is a penalty.**

Is this "the stability desired" for polynomial root-finding?

- What kind of **polynomial backward stability** is provided by this **perfect matrix backward stability**?
- Note that for our monic poly $p(z) = z^n + a_{n-1}\, z^{n-1} + \cdots + a_1\, z + \, a_0$,

$$c_n\, \|C\|_2 \leq \|p\|_\infty \leq d_n\, \|C\|_2,$$

for c_n, d_n low powers of n (note also $\|p\|_\infty \geq 1$).

- So, **MATLAB computed roots of $p(z)$ are** the exact eigenvalues of

$$C + E, \quad \text{with} \quad \|E\|_2 = O(u)\|C\|_2 = O(u)\|p\|_\infty,$$

or **the exact roots of**

$$\widetilde{p}(z) = \det(zI - (C + E)).$$

- Van Dooren & DeWilde (1983), Edelman & Murakami (1995), Lemmonier & Van Dooren (2003) proved

$$\widetilde{p}(z) = p(z) + e(z), \quad \text{with} \quad \|e(z)\|_\infty = O(u)\|p(z)\|_\infty^2,$$

- which means that **perfect matrix backward stability DOES NOT** imply perfect **polynomial backward stability** $\Longrightarrow$ **there is a penalty.**

Is this "the stability desired" for polynomial root-finding?

- What kind of **polynomial backward stability** is provided by this **perfect matrix backward stability**?
- Note that for our monic poly $p(z) = z^n + a_{n-1}\, z^{n-1} + \cdots + a_1\, z + a_0$,

$$c_n\, \|C\|_2 \leq \|p\|_\infty \leq d_n\, \|C\|_2,$$

for c_n, d_n low powers of n (note also $\|p\|_\infty \geq 1$).

- So, **MATLAB computed roots of $p(z)$ are** the exact eigenvalues of

$$C + E, \quad \text{with} \quad \|E\|_2 = O(u)\|C\|_2 = O(u)\|p\|_\infty,$$

or **the exact roots of**

$$\widetilde{p}(z) = \det(zI - (C + E)).$$

- Van Dooren & DeWilde (1983), Edelman & Murakami (1995), Lemmonier & Van Dooren (2003) proved

$$\widetilde{p}(z) = p(z) + e(z), \quad \text{with} \quad \|e(z)\|_\infty = O(u)\|p(z)\|_\infty^2,$$

- which means that **perfect matrix backward stability DOES NOT** imply perfect **polynomial backward stability** $\Longrightarrow$ **there is a penalty.**

Is this "the stability desired" for polynomial root-finding?

- What kind of **polynomial backward stability** is provided by this **perfect matrix backward stability**?
- Note that for our monic poly $p(z) = z^n + a_{n-1}\, z^{n-1} + \cdots + a_1\, z + \, a_0$,

$$c_n \, \|C\|_2 \leq \|p\|_\infty \leq d_n \, \|C\|_2,$$

  for c_n, d_n low powers of n (note also $\|p\|_\infty \geq 1$).

- So, **MATLAB computed roots of $p(z)$ are** the exact eigenvalues of

$$C + E, \quad \text{with} \quad \|E\|_2 = O(u)\|C\|_2 = O(u)\|p\|_\infty,$$

  or **the exact roots of**

$$\widetilde{p}(z) = \det(zI - (C + E)).$$

- Van Dooren & DeWilde (1983), Edelman & Murakami (1995), Lemmonier & Van Dooren (2003) proved

$$\widetilde{p}(z) = p(z) + e(z), \quad \text{with} \quad \|e(z)\|_\infty = O(u)\|p(z)\|_\infty^2,$$

- which means that **perfect matrix backward stability DOES NOT** imply perfect **polynomial backward stability** $\Longrightarrow$ there is a penalty.

Is this "the stability desired" for polynomial root-finding?

- What kind of **polynomial backward stability** is provided by this **perfect matrix backward stability**?
- Note that for our monic poly $p(z) = z^n + a_{n-1}\, z^{n-1} + \cdots + a_1\, z + a_0$,

$$c_n \, \|C\|_2 \leq \|p\|_\infty \leq d_n \, \|C\|_2,$$

for c_n, d_n low powers of n (note also $\|p\|_\infty \geq 1$).

- So, **MATLAB computed roots of $p(z)$ are** the exact eigenvalues of

$$C + E, \quad \text{with} \quad \|E\|_2 = O(u)\|C\|_2 = O(u)\|p\|_\infty,$$

or **the exact roots of**

$$\widetilde{p}(z) = \det(zI - (C + E)).$$

- Van Dooren & DeWilde (1983), Edelman & Murakami (1995), Lemmonier & Van Dooren (2003) proved

$$\widetilde{p}(z) = p(z) + e(z), \quad \text{with} \quad \|e(z)\|_\infty = O(u)\|p(z)\|_\infty^2,$$

- which means that **perfect matrix backward stability DOES NOT** imply perfect **polynomial backward stability** $\Longrightarrow$ **there is a penalty**.

Reflections on this penalty

This **penalty** in the **polynomial** backward error is an **intrinsic matrix perturbation phenomenon**, **independent of the algorithm**, and is determined by

1. the particular properties of the Frobenius companion matrix C,
2. the magnitude of $\|E\|_2 = O(u)\|C\|_2 (= O(u)\|p\|_\infty)$,
3. and the magnitude of

$$\|\widetilde{p}(z) - p(z)\|_\infty = \|\det(zI - (C + E)) - \det(zI - C)\|_\infty .$$

A key reason for this penalty is that E is dense and does not respect the structure of C.

Reflections on this penalty

This **penalty** in the **polynomial** backward error is an **intrinsic matrix perturbation phenomenon**, **independent of the algorithm**, and is determined by

1. the particular properties of the Frobenius companion matrix C,
2. the magnitude of $\|E\|_2 = O(u)\|C\|_2 (= O(u)\|p\|_\infty)$,
3. and the magnitude of

$$\|\widetilde{p}(z) - p(z)\|_\infty = \|\det(zI - (C + E)) - \det(zI - C)\|_\infty .$$

A key reason for this penalty is that E is dense and does not respect the structure of C.

Reflections on this penalty

This **penalty** in the **polynomial** backward error is an **intrinsic matrix perturbation phenomenon**, **independent of the algorithm**, and is determined by

1. the particular properties of the Frobenius companion matrix C,
2. the magnitude of $\|E\|_2 = O(u)\|C\|_2 (= O(u)\|p\|_\infty)$,
3. and the magnitude of

$$\|\widetilde{p}(z) - p(z)\|_\infty = \|\det(zI - (C+E)) - \det(zI - C)\|_\infty .$$

A key reason for this penalty is that E is dense and does not respect the structure of C.

Reflections on this penalty

This **penalty** in the **polynomial** backward error is an **intrinsic matrix perturbation phenomenon**, **independent of the algorithm**, and is determined by

1. the particular properties of the Frobenius companion matrix C,
2. the magnitude of $\|E\|_2 = O(u)\|C\|_2 (= O(u)\|p\|_\infty)$,
3. and the magnitude of

$$\|\widetilde{p}(z) - p(z)\|_\infty = \|\det(zI - (C+E)) - \det(zI - C)\|_\infty .$$

A key reason for this penalty is that E is dense and does not respect the structure of C.

Reflections on this penalty

This **penalty** in the **polynomial** backward error is an **intrinsic matrix perturbation phenomenon**, **independent of the algorithm**, and is determined by

1. the particular properties of the Frobenius companion matrix C,
2. the magnitude of $\|E\|_2 = O(u)\|C\|_2 (= O(u)\|p\|_\infty)$,
3. and the magnitude of

$$\|\widetilde{p}(z) - p(z)\|_\infty = \|\det(zI - (C + E)) - \det(zI - C)\|_\infty .$$

A key reason for this penalty is that E is dense and does not respect the structure of C.

But there are other companion matrices for $p(z)$!!

- In the last years many new classes of companion matrices have been developed.
- This intense activity has been mainly motivated by the numerical solution of polynomial eigenvalue problems.
- One of the most relevant among these new families are the **Fiedler companion matrices**, since they can be constructed very easily.
- In this scenario, we have solved a similar perturbation problem for the wider class of **Fiedler companion matrices** of $p(z)$ (with the hope of improving!!) and,
- if M_σ is a Fiedler matrix, we consider more general perturbations of M_σ

$$\|E\|_2 = O(u)\,\alpha(p)\,\|M_\sigma\|_2,$$

  where $\alpha(p)$ can be larger than one for backward errors of eigenvalue algorithms faster than traditional Francis-QR, but which may NOT be perfectly backward stable.
- **Goal of the talk:** To present our recent backward stability results on polynomial root-finding solved via eigenvalue algorithms applied on Fiedler matrices.

But there are other companion matrices for $p(z)$!!

- In the last years many new classes of companion matrices have been developed.
- This intense activity has been mainly motivated by the numerical solution of polynomial eigenvalue problems.
- One of the most relevant among these new families are the **Fiedler companion matrices**, since they can be constructed very easily.
- In this scenario, we have solved a similar perturbation problem for the wider class of **Fiedler companion matrices** of $p(z)$ (with the hope of improving!!) and,
- if M_σ is a Fiedler matrix, we consider more general perturbations of M_σ

$$\|E\|_2 = O(u)\, \alpha(p)\, \|M_\sigma\|_2,$$

where $\alpha(p)$ can be larger than one for backward errors of eigenvalue algorithms faster than traditional Francis-QR, but which may NOT be perfectly backward stable.
- **Goal of the talk:** To present our recent backward stability results on polynomial root-finding solved via eigenvalue algorithms applied on Fiedler matrices.

But there are other companion matrices for $p(z)$!!

- In the last years many new classes of companion matrices have been developed.
- This intense activity has been mainly motivated by the numerical solution of polynomial eigenvalue problems.
- One of the most relevant among these new families are the **Fiedler companion matrices**, since they can be constructed very easily.
- In this scenario, we have solved a similar perturbation problem for the wider class of **Fiedler companion matrices** of $p(z)$ (with the hope of improving!!) and,
- if M_σ is a Fiedler matrix, we consider more general perturbations of M_σ

$$\|E\|_2 = O(u)\,\alpha(p)\,\|M_\sigma\|_2,$$

  where $\alpha(p)$ can be larger than one for backward errors of eigenvalue algorithms faster than traditional Francis-QR, but which may NOT be perfectly backward stable.
- **Goal of the talk:** To present our recent backward stability results on polynomial root-finding solved via eigenvalue algorithms applied on Fiedler matrices.

But there are other companion matrices for $p(z)$!!

- In the last years many new classes of companion matrices have been developed.
- This intense activity has been mainly motivated by the numerical solution of polynomial eigenvalue problems.
- One of the most relevant among these new families are the **Fiedler companion matrices**, since they can be constructed very easily.
- In this scenario, we have solved a similar perturbation problem for the wider class of **Fiedler companion matrices** of $p(z)$ (with the hope of improving!!) and,
- if M_σ is a Fiedler matrix, we consider more general perturbations of M_σ

$$\|E\|_2 = O(u)\,\alpha(p)\,\|M_\sigma\|_2,$$

  where $\alpha(p)$ can be larger than one for backward errors of eigenvalue algorithms faster than traditional Francis-QR, but which may NOT be perfectly backward stable.
- **Goal of the talk:** To present our recent backward stability results on polynomial root-finding solved via eigenvalue algorithms applied on Fiedler matrices.

But there are other companion matrices for $p(z)$!!

- In the last years many new classes of companion matrices have been developed.
- This intense activity has been mainly motivated by the numerical solution of polynomial eigenvalue problems.
- One of the most relevant among these new families are the **Fiedler companion matrices**, since they can be constructed very easily.
- In this scenario, we have solved a similar perturbation problem for the wider class of **Fiedler companion matrices** of $p(z)$ (with the hope of improving!!) and,
- if M_σ is a Fiedler matrix, we consider more general perturbations of M_σ

$$\|E\|_2 = O(u)\,\alpha(p)\,\|M_\sigma\|_2,$$

  where $\alpha(p)$ can be larger than one for backward errors of eigenvalue algorithms faster than traditional Francis-QR, but which may NOT be perfectly backward stable.
- **Goal of the talk:** To present our recent backward stability results on polynomial root-finding solved via eigenvalue algorithms applied on Fiedler matrices.

But there are other companion matrices for $p(z)$!!

- In the last years many new classes of companion matrices have been developed.
- This intense activity has been mainly motivated by the numerical solution of polynomial eigenvalue problems.
- One of the most relevant among these new families are the **Fiedler companion matrices**, since they can be constructed very easily.
- In this scenario, we have solved a similar perturbation problem for the wider class of **Fiedler companion matrices** of $p(z)$ (with the hope of improving!!) and,
- if M_σ is a Fiedler matrix, we consider more general perturbations of M_σ

$$\|E\|_2 = O(u)\,\alpha(p)\,\|M_\sigma\|_2,$$

  where $\alpha(p)$ can be larger than one for backward errors of eigenvalue algorithms faster than traditional Francis-QR, but which may NOT be perfectly backward stable.
- **Goal of the talk:** To present our recent backward stability results on polynomial root-finding solved via eigenvalue algorithms applied on Fiedler matrices.

A summary of the main result for Fiedler matrices

- Fiedler matrices also satisfy $\tilde{c}_n \, \|M_\sigma\|_2 \leq \|p\|_\infty \leq \tilde{d}_n \, \|M_\sigma\|_2$,
- and we have proved that if

$$\|E\|_2 = O(u)\alpha(p)\|M_\sigma\|_2 \; (= O(u)\alpha(p)\|p\|_\infty),$$

then

$$\|\widetilde{p}(z) - p(z)\|_\infty = \|\det(zI - (M_\sigma + E)) - \det(zI - M_\sigma)\|_\infty$$
$$= O(u)\,\alpha(p)\,\|p(z)\|_\infty^3,$$

if M_σ is not a Frobenius companion matrix.

- So, **the penalty in the transition from matrix to polynomial backward errors is larger than for the classical Frobenius companion matrix,**
- but, note that all are satisfactory if $\|p\|_\infty$ is moderate and none is if $\|p\|_\infty$ is large.

- Fiedler matrices also satisfy $\tilde{c}_n \, \|M_\sigma\|_2 \leq \|p\|_\infty \leq \tilde{d}_n \, \|M_\sigma\|_2$,
- and we have proved that if

$$\|E\|_2 = O(u)\alpha(p)\|M_\sigma\|_2 \; (= O(u)\alpha(p)\|p\|_\infty),$$

then

$$\begin{aligned} \|\widetilde{p}(z) - p(z)\|_\infty &= \|\det(zI - (M_\sigma + E)) - \det(zI - M_\sigma)\|_\infty \\ &= O(u)\, \alpha(p)\, \|p(z)\|_\infty^3, \end{aligned}$$

if M_σ is not a Frobenius companion matrix.

- So, **the penalty in the transition from matrix to polynomial backward errors is larger than for the classical Frobenius companion matrix,**
- but, note that all are satisfactory if $\|p\|_\infty$ is moderate and none is if $\|p\|_\infty$ is large.

A summary of the main result for Fiedler matrices

- Fiedler matrices also satisfy $\tilde{c}_n \, \|M_\sigma\|_2 \leq \|p\|_\infty \leq \tilde{d}_n \, \|M_\sigma\|_2$,
- and we have proved that if

$$\|E\|_2 = O(u)\alpha(p)\|M_\sigma\|_2 \; (= O(u)\alpha(p)\|p\|_\infty),$$

then

$$\|\widetilde{p}(z) - p(z)\|_\infty = \|\det(zI - (M_\sigma + E)) - \det(zI - M_\sigma)\|_\infty$$
$$= O(u)\, \alpha(p)\, \|p(z)\|_\infty^3,$$

if M_σ is not a Frobenius companion matrix.

- So, **the penalty in the transition from matrix to polynomial backward errors is larger than for the classical Frobenius companion matrix,**
- but, note that all are satisfactory if $\|p\|_\infty$ is moderate and none is if $\|p\|_\infty$ is large.

A summary of the main result for Fiedler matrices

- Fiedler matrices also satisfy $\tilde{c}_n \, \|M_\sigma\|_2 \leq \|p\|_\infty \leq \tilde{d}_n \, \|M_\sigma\|_2$,
- and we have proved that if

$$\|E\|_2 = O(u)\alpha(p)\|M_\sigma\|_2 \; (= O(u)\alpha(p)\|p\|_\infty),$$

then

$$\|\widetilde{p}(z) - p(z)\|_\infty = \|\text{det}(zI - (M_\sigma + E)) - \text{det}(zI - M_\sigma)\|_\infty$$

$$= \; O(u) \, \alpha(p) \, \|p(z)\|_\infty^3,$$

if M_σ is not a Frobenius companion matrix.

- So, **the penalty in the transition from matrix to polynomial backward errors is larger than for the classical Frobenius companion matrix,**
- but, note that all are satisfactory if $\|p\|_\infty$ is moderate and none is if $\|p\|_\infty$ is large.

“...a general principle: a numerical process is more likely to be backward stable when the number of outputs is small compared with the number of inputs, so that there is an abundance of data onto which to “throw the backward error”...”

N. Higham, *Accuracy and Stability of Numerical Algorithms*, 2nd ed., SIAM, 2002, p.65.

A final surprise: companion pencils and QZ algorithm (I)

Let us go back to the original nonmonic problem, i.e., to compute all the roots of a scalar polynomial

$$q(z) = b_n\, z^n + b_{n-1}\, z^{n-1} + \cdots + b_1\, z + b_0, \quad b_i \in \mathbb{C}\,,$$

- and define the first Frobenius companion pencil

$$C(z) = z \begin{bmatrix} b_n & & & \\ & 1 & & \\ & & \ddots & \\ & & & 1 \end{bmatrix} - \begin{bmatrix} -b_{n-1} & \cdots & -b_1 & -b_0 \\ 1 & & & \\ & \ddots & & \\ & & 1 & \end{bmatrix},$$

- or any other Fiedler companion pencil

$$F_\sigma(z) = z \begin{bmatrix} b_n & & & \\ & 1 & & \\ & & \ddots & \\ & & & 1 \end{bmatrix} - M_\sigma\,.$$

Both satisfy: $q(z) = \det(C(z)) = \det(F_\sigma(z))$.

Let us go back to the original nonmonic problem, i.e., to compute all the roots of a scalar polynomial

$$q(z) = b_n\, z^n + b_{n-1}\, z^{n-1} + \cdots + b_1\, z + b_0, \quad b_i \in \mathbb{C}\,,$$

- and define the first Frobenius companion pencil

$$C(z) = z \begin{bmatrix} b_n & & & \\ & 1 & & \\ & & \ddots & \\ & & & 1 \end{bmatrix} - \begin{bmatrix} -b_{n-1} & \cdots & -b_1 & -b_0 \\ 1 & & & \\ & \ddots & & \\ & & 1 & \end{bmatrix},$$

- or any other Fiedler companion pencil

$$F_\sigma(z) = z \begin{bmatrix} b_n & & & \\ & 1 & & \\ & & \ddots & \\ & & & 1 \end{bmatrix} - M_\sigma\,.$$

Both satisfy: $q(z) = \det(C(z)) = \det(F_\sigma(z))$.

A final surprise: companion pencils and QZ algorithm (I)

Let us go back to the original nonmonic problem, i.e., to compute all the roots of a scalar polynomial

$$q(z) = b_n\, z^n + b_{n-1}\, z^{n-1} + \cdots + b_1\, z + b_0, \quad b_i \in \mathbb{C}\,,$$

- and define the first Frobenius companion pencil

$$C(z) = z \begin{bmatrix} b_n & & & \\ & 1 & & \\ & & \ddots & \\ & & & 1 \end{bmatrix} - \begin{bmatrix} -b_{n-1} & \cdots & -b_1 & -b_0 \\ 1 & & & \\ & \ddots & & \\ & & 1 & \end{bmatrix},$$

- or any other Fiedler companion pencil

$$F_\sigma(z) = z \begin{bmatrix} b_n & & & \\ & 1 & & \\ & & \ddots & \\ & & & 1 \end{bmatrix} - M_\sigma\,.$$

Both satisfy: $q(z) = \det(C(z)) = \det(F_\sigma(z))$.

A final surprise: companion pencils and QZ algorithm (I)

Let us go back to the original nonmonic problem, i.e., to compute all the roots of a scalar polynomial

$$q(z) = b_n\, z^n + b_{n-1}\, z^{n-1} + \cdots + b_1\, z + b_0, \quad b_i \in \mathbb{C}\,,$$

- and define the first Frobenius companion pencil

$$C(z) = z \begin{bmatrix} b_n & & & \\ & 1 & & \\ & & \ddots & \\ & & & 1 \end{bmatrix} - \begin{bmatrix} -b_{n-1} & \cdots & -b_1 & -b_0 \\ 1 & & & \\ & \ddots & & \\ & & 1 & \end{bmatrix},$$

- or any other Fiedler companion pencil

$$F_\sigma(z) = z \begin{bmatrix} b_n & & & \\ & 1 & & \\ & & \ddots & \\ & & & 1 \end{bmatrix} - M_\sigma\,.$$

Both satisfy: $q(z) = \det(C(z)) = \det(F_\sigma(z))$.

Alternative algorithms.

- **Step 1. Normalize the polynomial**

$$q(z) = b_n\, z^n + b_{n-1}\, z^{n-1} + \cdots + b_1\, z + \, b_0 \longrightarrow \; s(z) := q(z)/\|q(z)\|_\infty \, .$$

- **Step 2.** Construct the Frobenius or any other Fiedler companion pencil for $s(z)$.
- **Step 3.** Compute all the eigenvalues of the pencil using the QZ algorithm for pencils.

Remark: This seems at a first glance a great way to WASTE CPU-time because the number of flops used by the standard QZ algorithm is three times the number of flops used by the QR algorithm,

but...

Alternative algorithms.

- **Step 1. Normalize the polynomial**

$$q(z) = b_n\, z^n + b_{n-1}\, z^{n-1} + \cdots + b_1\, z + b_0 \longrightarrow s(z) := q(z)/\|q(z)\|_\infty \,.$$

- **Step 2.** Construct the Frobenius or any other Fiedler companion pencil for $s(z)$.

- **Step 3.** Compute all the eigenvalues of the pencil using the QZ algorithm for pencils.

Remark: This seems at a first glance a great way to WASTE CPU-time because the number of flops used by the standard QZ algorithm is three times the number of flops used by the QR algorithm,

but...

Alternative algorithms.

- **Step 1. Normalize the polynomial**

$$q(z) = b_n \, z^n + b_{n-1} \, z^{n-1} + \cdots + b_1 \, z + \, b_0 \longrightarrow \; s(z) := q(z)/\|q(z)\|_\infty \, .$$

- **Step 2.** Construct the Frobenius or any other Fiedler companion pencil for $s(z)$.
- **Step 3.** Compute all the eigenvalues of the pencil using the QZ algorithm for pencils.

Remark: This seems at a first glance a great way to WASTE CPU-time because the number of flops used by the standard QZ algorithm is three times the number of flops used by the QR algorithm,

but...

Alternative algorithms.

- **Step 1. Normalize the polynomial**

$$q(z) = b_n\, z^n + b_{n-1}\, z^{n-1} + \cdots + b_1\, z + b_0 \longrightarrow s(z) := q(z)/\|q(z)\|_\infty \,.$$

- **Step 2.** Construct the Frobenius or any other Fiedler companion pencil for $s(z)$.
- **Step 3.** Compute all the eigenvalues of the pencil using the QZ algorithm for pencils.

Remark: This seems at a first glance a great way to WASTE CPU-time because the number of flops used by the standard QZ algorithm is three times the number of flops used by the QR algorithm,

but...

Alternative algorithms.

- **Step 1. Normalize the polynomial**

$$q(z) = b_n\, z^n + b_{n-1}\, z^{n-1} + \cdots + b_1\, z + \, b_0 \longrightarrow \; s(z) := q(z)/\|q(z)\|_\infty \, .$$

- **Step 2.** Construct the Frobenius or any other Fiedler companion pencil for $s(z)$.
- **Step 3.** Compute all the eigenvalues of the pencil using the QZ algorithm for pencils.

Remark: This seems at a first glance a great way to WASTE CPU-time because the number of flops used by the standard QZ algorithm is three times the number of flops used by the QR algorithm,

but...

A final surprise: companion pencils and QZ algorithm (III)

- using the backward stability of the QZ algorithm applied on any regular pencil $zA - B$, that is, the computed eigenvalues are the exact eigenvalues of

$$z(A+E_A)-(B+E_B), \quad \text{with} \quad \|E_A\|_2 = O(u)\|A\|_2, \; \|E_B\|_2 = O(u)\|B\|_2,$$

- and the normalization of the polynomial, $\|s(z)\|_\infty = 1$, which implies that, for the pencils we are considering,

$$\|A\|_2 \leq \sqrt{2\,n} \quad \text{and} \quad \|B\|_2 \leq \sqrt{2\,n},$$

- one can prove with a careful analysis that the whole ensemble of computed roots is the whole ensemble of roots of $\widetilde{q}(z)$ with

$$\|q(z) - \widetilde{q}(z)\|_\infty = O(u)\,\|q(z)\|_\infty,$$

that is, **perfect polynomial backward stability!!!**

A final surprise: companion pencils and QZ algorithm (III)

- using the backward stability of the QZ algorithm applied on any regular pencil $zA - B$, that is, the computed eigenvalues are the exact eigenvalues of

$$z(A+E_A)-(B+E_B), \quad \text{with} \quad \|E_A\|_2 = O(u)\|A\|_2, \; \|E_B\|_2 = O(u)\|B\|_2,$$

- and the normalization of the polynomial, $\|s(z)\|_\infty = 1$, which implies that, for the pencils we are considering,

$$\|A\|_2 \leq \sqrt{2\,n} \quad \text{and} \quad \|B\|_2 \leq \sqrt{2\,n},$$

- one can prove with a careful analysis that the whole ensemble of computed roots is the whole ensemble of roots of $\widetilde{q}(z)$ with

$$\|q(z) - \widetilde{q}(z)\|_\infty = O(u)\,\|q(z)\|_\infty,$$

that is, **perfect polynomial backward stability!!!**

- using the backward stability of the QZ algorithm applied on any regular pencil $zA - B$, that is, the computed eigenvalues are the exact eigenvalues of

$$z(A + E_A) - (B + E_B), \quad \text{with} \quad \|E_A\|_2 = O(u)\|A\|_2, \; \|E_B\|_2 = O(u)\|B\|_2,$$

- and the normalization of the polynomial, $\|s(z)\|_\infty = 1$, which implies that, for the pencils we are considering,

$$\|A\|_2 \leq \sqrt{2\,n} \quad \text{and} \quad \|B\|_2 \leq \sqrt{2\,n},$$

- one can prove with a careful analysis that the whole ensemble of computed roots is the whole ensemble of roots of $\widetilde{q}(z)$ with

$$\|q(z) - \widetilde{q}(z)\|_\infty = O(u)\,\|q(z)\|_\infty,$$

that is, **perfect polynomial backward stability!!!**

Outline

1. Perturbation of characteristic polynomials of general matrices
2. Antecedents: results for Frobenius companion matrices
3. Fiedler matrices: definition and properties
4. Backward errors of poly. root-finding from Fiedler matrices
5. Balancing Fiedler matrices
6. Numerical experiments
7. Backward errors of poly. root-finding from Fiedler PENCILS
8. Conclusions

Outline

Jacobi formula and consequences (I)

Theorem (Jacobi)

Let $A, E \in \mathbb{C}^{n\times n}$. *Then*

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

where $\mathrm{adj}(zI - A)$ *is the adjugate matrix (or classical adjoint) of* $zI - A$*, i.e., the transpose matrix of its cofactors.*

Lemma (Gantmacher, 1959)

Let $A \in \mathbb{C}^{n\times n}$ *and* $p(z) := \det(zI - A) = z^n + a_{n-1}\,z^{n-1} + \cdots + a_1\,z + a_0$. *Then*

$$\mathrm{adj}(zI - A) = \sum_{k=0}^{n-1} z^k A_k, \quad A_k \in \mathbb{C}^{n\times n},$$

and

$$A_{n-1} = I, \quad A_k = A\,A_{k+1} + a_{k+1} I, \quad \textit{for } k = n-2, n-3, \ldots, 0.$$

Theorem (Jacobi)

Let $A, E \in \mathbb{C}^{n\times n}$. *Then*

$$\begin{aligned}\widetilde{p}(z) - p(z) &:= \det(zI - (A+E)) - \det(zI - A)\\ &= -\mathrm{trace}(\,\mathrm{adj}(zI - A)\,E) + O(\|E\|^2),\end{aligned}$$

where $\mathrm{adj}(zI - A)$ *is the adjugate matrix (or classical adjoint) of* $zI - A$, *i.e., the transpose matrix of its cofactors.*

Lemma (Gantmacher, 1959)

Let $A \in \mathbb{C}^{n\times n}$ *and* $p(z) := \det(zI - A) = z^n + a_{n-1}\,z^{n-1} + \cdots + a_1\,z + a_0$. *Then*

$$\mathrm{adj}(zI - A) = \sum_{k=0}^{n-1} z^k A_k, \quad A_k \in \mathbb{C}^{n\times n},$$

and

$$A_{n-1} = I, \quad A_k = A\,A_{k+1} + a_{k+1} I, \quad \textit{for} \;\; k = n-2, n-3, \ldots, 0.$$

Jacobi formula and consequences (II)

Theorem

Let $A, E \in \mathbb{C}^{n\times n}$,

$$p(z) := \det(zI - A) = z^n + a_{n-1}\, z^{n-1} + \cdots + a_1\, z + a_0,$$
$$\widetilde{p}(z) := \det(zI - (A+E)) = z^n + \widetilde{a}_{n-1}\, z^{n-1} + \cdots + \widetilde{a}_1\, z + \widetilde{a}_0,$$

and

$$\mathrm{adj}(zI - A) = \sum_{k=0}^{n-1} z^k A_k, \quad A_k \in \mathbb{C}^{n\times n}.$$

Then

$$\widetilde{a}_k - a_k := -\mathrm{trace}(\, A_k\, E) + O(\|E\|^2), \quad \textit{for } k = 0, 1, \ldots, n-1.$$

Explicit formulas for $\mathrm{trace}(\, A_k\, E)$ obtained for

- $A =$ Frobenius companion matrix of $p(z)$ by Edelman-Murakami (1995),
- $A = M_\sigma$ **any other Fiedler companion matrix of** $p(z)$ **in this talk**.

Jacobi formula and consequences (II)

Theorem

Let $A, E \in \mathbb{C}^{n\times n}$,

$$p(z) := \det(zI - A) = z^n + a_{n-1}\, z^{n-1} + \cdots + a_1\, z + \, a_0,$$
$$\widetilde{p}(z) := \det(zI - (A+E)) = z^n + \widetilde{a}_{n-1}\, z^{n-1} + \cdots + \widetilde{a}_1\, z + \, \widetilde{a}_0,$$

and

$$\operatorname{adj}(zI - A) = \sum_{k=0}^{n-1} z^k A_k, \quad A_k \in \mathbb{C}^{n\times n}.$$

Then

$$\widetilde{a}_k - a_k := -\operatorname{trace}(\, A_k\, E) + O(\|E\|^2), \quad \textit{for } \; k = 0, 1, \ldots, n-1.$$

Explicit formulas for $\operatorname{trace}(\, A_k\, E)$ obtained for

- $A =$ Frobenius companion matrix of $p(z)$ by Edelman-Murakami (1995),
- $A = M_\sigma$ **any other Fiedler companion matrix of $p(z)$ in this talk**.

Outline

The best known companion matrices of a monic polynomial

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

are the first and second Frobenius companion matrices of $p(z)$:

$$C_1 = \begin{bmatrix} -a_{n-1} & \cdots & -a_1 & -a_0 \\ 1 & & & \\ & \ddots & & \\ & & 1 & \end{bmatrix}, \quad C_2 = \begin{bmatrix} -a_{n-1} & 1 & & \\ \vdots & & \ddots & \\ -a_1 & & & 1 \\ -a_0 & & & \end{bmatrix},$$

which have the property that

$$\det(zI - C_1) = \det(zI - C_2) = p(z)\,.$$

Theorem (Edelman, Murakami, 1995)

Let $C_1 \in \mathbb{C}^{n\times n}$ *be the first Frobenius companion matrix of* $p(z)$, $E \in \mathbb{C}^{n\times n}$, *and*

$$p(z) := \det(zI - C_1) = z^n + a_{n-1}\, z^{n-1} + \cdots + a_1\, z + a_0,$$
$$\widetilde{p}(z) := \det(zI - (C_1 + E)) = z^n + \widetilde{a}_{n-1}\, z^{n-1} + \cdots + \widetilde{a}_1\, z + \widetilde{a}_0.$$

Then, to first order in E*:*

$$\widetilde{a}_k - a_k = \sum_{s=0}^{k} \sum_{j=1}^{n-k-1} a_s E_{j-s+k+1,j} - \sum_{s=k+1}^{n} \sum_{j=n-k}^{n} a_s E_{j-s+k+1,j}.$$

Perturbation of the characteristic polynomial of C_1

Theorem (Edelman, Murakami, 1995)

Let $C_1 \in \mathbb{C}^{n\times n}$ be the first Frobenius companion matrix of $p(z)$, $E \in \mathbb{C}^{n\times n}$, and

$$p(z) := \det(zI - C_1) = z^n + a_{n-1}\, z^{n-1} + \cdots + a_1\, z + a_0,$$
$$\widetilde{p}(z) := \det(zI - (C_1 + E)) = z^n + \widetilde{a}_{n-1}\, z^{n-1} + \cdots + \widetilde{a}_1\, z + \widetilde{a}_0.$$

Then, to first order in E:

$$\widetilde{a}_k - a_k = \sum_{s=0}^{k} \sum_{j=1}^{n-k-1} a_s E_{j-s+k+1,j} - \sum_{s=k+1}^{n} \sum_{j=n-k}^{n} a_s E_{j-s+k+1,j}.$$

Corollary

Let $C_1 \in \mathbb{C}^{n\times n}$ be the first Frobenius companion matrix of $p(z)$, $E \in \mathbb{C}^{n\times n}$, and

$$p(z) := \det(zI - C_1) = z^n + a_{n-1}\, z^{n-1} + \cdots + a_1\, z + a_0,$$
$$\widetilde{p}(z) := \det(zI - (C_1 + E)) = z^n + \widetilde{a}_{n-1}\, z^{n-1} + \cdots + \widetilde{a}_1\, z + \widetilde{a}_0.$$

If $\|E\|_2 = O(u)\,\alpha(p)\,\|C_1\|_2$, then

$$\|\widetilde{p}(z) - p(z)\|_\infty = O(u)\,\alpha(p)\,\|p(z)\|_\infty^{\mathbf{2}}.$$

- Even the "superstable" QR-algorithm applied to C_1 does not lead to a backward stable polynomial root-finding method. Yes if $\|p(z)\|_\infty \approx 1$.
- Edelman & Murakami provided numerical evidence that shows that if **balancing** is used before the QR-algorithm is applied to C_1, then

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

Corollary

Let $C_1 \in \mathbb{C}^{n\times n}$ be the first Frobenius companion matrix of $p(z)$, $E \in \mathbb{C}^{n\times n}$, and

$$p(z) := \det(zI - C_1) = z^n + a_{n-1}\, z^{n-1} + \cdots + a_1\, z + a_0,$$
$$\widetilde{p}(z) := \det(zI - (C_1 + E)) = z^n + \widetilde{a}_{n-1}\, z^{n-1} + \cdots + \widetilde{a}_1\, z + \widetilde{a}_0.$$

If $\|E\|_2 = O(u)\,\alpha(p)\,\|C_1\|_2$, *then*

$$\|\widetilde{p}(z) - p(z)\|_\infty = O(u)\,\alpha(p)\,\|p(z)\|_\infty^{\mathbf{2}}.$$

- Even the "superstable" QR-algorithm applied to C_1 does not lead to a backward stable polynomial root-finding method. Yes if $\|p(z)\|_\infty \approx 1$.
- Edelman & Murakami provided numerical evidence that shows that if **balancing** is used before the QR-algorithm is applied to C_1, then

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

Corollary

Let $C_1 \in \mathbb{C}^{n\times n}$ be the first Frobenius companion matrix of $p(z)$, $E \in \mathbb{C}^{n\times n}$, and

$$p(z) := \det(zI - C_1) = z^n + a_{n-1}\, z^{n-1} + \cdots + a_1\, z + \, a_0,$$
$$\widetilde{p}(z) := \det(zI - (C_1 + E)) = z^n + \widetilde{a}_{n-1}\, z^{n-1} + \cdots + \widetilde{a}_1\, z + \, \widetilde{a}_0.$$

If $\|E\|_2 = O(u)\,\alpha(p)\,\|C_1\|_2$*, then*

$$\|\widetilde{p}(z) - p(z)\|_\infty = O(u)\,\alpha(p)\,\|p(z)\|_\infty^{\mathbf{2}}.$$

- Even the "superstable" QR-algorithm applied to C_1 does not lead to a backward stable polynomial root-finding method. Yes if $\|p(z)\|_\infty \approx 1$.
- Edelman & Murakami provided numerical evidence that shows that if **balancing** is used before the QR-algorithm is applied to C_1, then

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

Corollary

Let $C_1 \in \mathbb{C}^{n\times n}$ be the first Frobenius companion matrix of $p(z)$, $E \in \mathbb{C}^{n\times n}$, and

$$p(z) := \det(zI - C_1) = z^n + a_{n-1}\, z^{n-1} + \cdots + a_1\, z + \, a_0,$$
$$\widetilde{p}(z) := \det(zI - (C_1 + E)) = z^n + \widetilde{a}_{n-1}\, z^{n-1} + \cdots + \widetilde{a}_1\, z + \, \widetilde{a}_0.$$

If $\|E\|_2 = O(u)\,\alpha(p)\,\|C_1\|_2$, *then*

$$\|\widetilde{p}(z) - p(z)\|_\infty = O(u)\,\alpha(p)\,\|p(z)\|_\infty^{\mathbf{2}}.$$

- Even the "superstable" QR-algorithm applied to C_1 does not lead to a backward stable polynomial root-finding method. Yes if $\|p(z)\|_\infty \approx 1$.
- Edelman & Murakami provided numerical evidence that shows that if **balancing** is used before the QR-algorithm is applied to C_1, then

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

Penalty in polynomial backward errors from C_1

Corollary

Let $C_1 \in \mathbb{C}^{n\times n}$ *be the first Frobenius companion matrix of* $p(z)$, $E \in \mathbb{C}^{n\times n}$, *and*

$$p(z) := \det(zI - C_1) = z^n + a_{n-1}\, z^{n-1} + \cdots + a_1\, z + a_0,$$
$$\widetilde{p}(z) := \det(zI - (C_1 + E)) = z^n + \widetilde{a}_{n-1}\, z^{n-1} + \cdots + \widetilde{a}_1\, z + \widetilde{a}_0.$$

If $\|E\|_2 = O(u)\,\alpha(p)\,\|C_1\|_2$, *then*

$$\|\widetilde{p}(z) - p(z)\|_\infty = O(u)\,\alpha(p)\,\|p(z)\|_\infty^{\mathbf{2}}.$$

- Even the "superstable" QR-algorithm applied to C_1 does not lead to a backward stable polynomial root-finding method. Yes if $\|p(z)\|_\infty \approx 1$.
- Edelman & Murakami provided numerical evidence that shows that if **balancing** is used before the QR-algorithm is applied to C_1, then

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

Outline

Given $p(z) = z^n + a_{n-1}z^{n-1} + \cdots + a_1 z + a_0$, we define the following matrices

$$M_i := \begin{bmatrix} I_{n-i-1} & & & \\ & -a_i & 1 & \\ & 1 & 0 & \\ & & & I_{i-1} \end{bmatrix} \in \mathbb{C}^{n\times n}, \qquad i = 1, 2, \ldots, n-1$$

$$M_0 := \begin{bmatrix} I_{n-1} & 0 \\ 0 & -a_0 \end{bmatrix} \in \mathbb{C}^{n\times n}$$

For any permutation $\sigma = (i_0, i_1, \ldots, i_{n-1})$ of $(0, 1, \ldots, n-1)$, the Fiedler companion matrix of $p(z)$ associated to σ is

$$M_\sigma = M_{i_0} M_{i_1} \cdots M_{i_{n-1}}$$

Theorem (Fiedler, LAA, 2003)

For any monic polynomial $p(z)$, all associated Fiedler matrices are similar to each other, and their characteristic polynomials are equal to $p(z)$.

Definition of Fiedler matrices (Fiedler, LAA, 2003)

Given $p(z) = z^n + a_{n-1}z^{n-1} + \cdots + a_1 z + a_0$, we define the following matrices

$$M_i := \begin{bmatrix} I_{n-i-1} & & & \\ & -a_i & 1 & \\ & 1 & 0 & \\ & & & I_{i-1} \end{bmatrix} \in \mathbb{C}^{n\times n}, \qquad i = 1, 2, \ldots, n-1$$

$$M_0 := \begin{bmatrix} I_{n-1} & 0 \\ 0 & -a_0 \end{bmatrix} \in \mathbb{C}^{n\times n}$$

For any permutation $\sigma = (i_0, i_1, \ldots, i_{n-1})$ of $(0, 1, \ldots, n-1)$, the Fiedler companion matrix of $p(z)$ associated to σ is

$$M_\sigma = M_{i_0} M_{i_1} \cdots M_{i_{n-1}}$$

Theorem (Fiedler, LAA, 2003)

For any monic polynomial $p(z)$, all associated Fiedler matrices are similar to each other, and their characteristic polynomials are equal to $p(z)$.

Definition of Fiedler matrices (Fiedler, LAA, 2003)

Given $p(z) = z^n + a_{n-1}z^{n-1} + \cdots + a_1 z + a_0$, we define the following matrices

$$M_i := \begin{bmatrix} I_{n-i-1} & & & \\ & -a_i & 1 & \\ & 1 & 0 & \\ & & & I_{i-1} \end{bmatrix} \in \mathbb{C}^{n\times n}, \qquad i = 1, 2, \ldots, n-1$$

$$M_0 := \begin{bmatrix} I_{n-1} & 0 \\ 0 & -a_0 \end{bmatrix} \in \mathbb{C}^{n\times n}$$

For any permutation $\sigma = (i_0, i_1, \ldots, i_{n-1})$ of $(0, 1, \ldots, n-1)$, the Fiedler companion matrix of $p(z)$ associated to σ is

$$M_\sigma = M_{i_0} M_{i_1} \cdots M_{i_{n-1}}$$

Theorem (Fiedler, LAA, 2003)

For any monic polynomial $p(z)$, all associated Fiedler matrices are similar to each other, and their characteristic polynomials are equal to $p(z)$.

Examples of Fiedler matrices

$$p(z) = z^6 + a_5 z^5 + a_4 z^4 + a_3 z^3 + a_2 z^2 + a_1 z + a_0$$

First Frobenius companion matrix: $C_1 = M_5 M_4 M_3 M_2 M_1 M_0$

$$= \begin{bmatrix} -a_5 & -a_4 & -a_3 & -a_2 & -a_1 & -a_0 \\ 1 & & & & & \\ & 1 & & & & \\ & & 1 & & & \\ & & & 1 & & \\ & & & & 1 & \end{bmatrix}$$

Second Frobenius companion matrix: $C_2 = M_0 M_1 M_2 M_3 M_4 M_5$

$$= \begin{bmatrix} -a_5 & 1 & & & & \\ -a_4 & & 1 & & & \\ -a_3 & & & 1 & & \\ -a_2 & & & & 1 & \\ -a_1 & & & & & 1 \\ -a_0 & & & & & \end{bmatrix}$$

Examples of Fiedler matrices

$$p(z) = z^6 + a_5 z^5 + a_4 z^4 + a_3 z^3 + a_2 z^2 + a_1 z + a_0$$

First Frobenius companion matrix: $C_1 = M_5 M_4 M_3 M_2 M_1 M_0$

$$= \begin{bmatrix} -a_5 & -a_4 & -a_3 & -a_2 & -a_1 & -a_0 \\ 1 & & & & & \\ & 1 & & & & \\ & & 1 & & & \\ & & & 1 & & \\ & & & & 1 & \end{bmatrix}$$

Second Frobenius companion matrix: $C_2 = M_0 M_1 M_2 M_3 M_4 M_5$

$$= \begin{bmatrix} -a_5 & 1 & & & & \\ -a_4 & & 1 & & & \\ -a_3 & & & 1 & & \\ -a_2 & & & & 1 & \\ -a_1 & & & & & 1 \\ -a_0 & & & & & \end{bmatrix}$$

Examples of Fiedler matrices

$$p(z) = z^6 + a_5 z^5 + a_4 z^4 + a_3 z^3 + a_2 z^2 + a_1 z + a_0$$

First Frobenius companion matrix: $C_1 = M_5 M_4 M_3 M_2 M_1 M_0$

$$= \begin{bmatrix} -a_5 & -a_4 & -a_3 & -a_2 & -a_1 & -a_0 \\ 1 & & & & & \\ & 1 & & & & \\ & & 1 & & & \\ & & & 1 & & \\ & & & & 1 & \end{bmatrix}$$

Second Frobenius companion matrix: $C_2 = M_0 M_1 M_2 M_3 M_4 M_5$

$$= \begin{bmatrix} -a_5 & 1 & & & & \\ -a_4 & & 1 & & & \\ -a_3 & & & 1 & & \\ -a_2 & & & & 1 & \\ -a_1 & & & & & 1 \\ -a_0 & & & & & \end{bmatrix}$$

Examples of Fiedler matrices

$$p(z) = z^6 + a_5 z^5 + a_4 z^4 + a_3 z^3 + a_2 z^2 + a_1 z + a_0$$

First Frobenius companion matrix: $C_1 = M_5 M_4 M_3 M_2 M_1 M_0$

$$= \begin{bmatrix} -a_5 & -a_4 & -a_3 & -a_2 & -a_1 & -a_0 \\ 1 & & & & & \\ & 1 & & & & \\ & & 1 & & & \\ & & & 1 & & \\ & & & & 1 & \end{bmatrix}$$

Another Fiedler matrix: $M_\sigma = M_0 M_1 M_5 M_4 M_3 M_2$

$$= \begin{bmatrix} -a_5 & -a_4 & -a_3 & -a_2 & 1 & \\ 1 & & & & & \\ & 1 & & & & \\ & & 1 & & & \\ & & & -a_1 & & 1 \\ & & & -a_0 & & \end{bmatrix}$$

Examples of Fiedler matrices

$$p(z) = z^6 + a_5 z^5 + a_4 z^4 + a_3 z^3 + a_2 z^2 + a_1 z + a_0$$

First Frobenius companion matrix: $C_1 = M_5 M_4 M_3 M_2 M_1 M_0$

$$= \begin{bmatrix} -a_5 & -a_4 & -a_3 & -a_2 & -a_1 & -a_0 \\ 1 & & & & & \\ & 1 & & & & \\ & & 1 & & & \\ & & & 1 & & \\ & & & & 1 & \end{bmatrix}$$

Another Fiedler matrix: $M_\sigma = M_0 M_1 M_5 M_4 M_3 M_2$

$$= \begin{bmatrix} -a_5 & -a_4 & -a_3 & -a_2 & 1 & \\ 1 & & & & & \\ & 1 & & & & \\ & & 1 & & & \\ & & & -a_1 & & 1 \\ & & & -a_0 & & \end{bmatrix}$$

Examples of Fiedler matrices

First Frobenius companion matrix: $C_1 = M_5 M_4 M_3 M_2 M_1 M_0$

$$
= \begin{bmatrix} -a_5 & -a_4 & -a_3 & -a_2 & -a_1 & -a_0 \\ 1 & & & & & \\ & 1 & & & & \\ & & 1 & & & \\ & & & 1 & & \\ & & & & 1 & \end{bmatrix}
$$

Another Fiedler matrix: $M_\sigma = M_0 M_1 M_5 M_4 M_3 M_2$

$$
= \begin{bmatrix} -a_5 & -a_4 & -a_3 & -a_2 & 1 & \\ 1 & & & & & \\ & 1 & & & & \\ & & 1 & & & \\ & & & -a_1 & & 1 \\ & & & -a_0 & & \end{bmatrix}
$$

Structural property 1 of Fiedler matrices

Every Fiedler matrix has exactly the **same entries** as the first Frobenius companion matrix (in different positions).

Examples of Fiedler matrices

First Frobenius companion matrix: $C_1 = M_5 M_4 M_3 M_2 M_1 M_0$

$$= \begin{bmatrix} -a_5 & -a_4 & -a_3 & -a_2 & -a_1 & -a_0 \\ 1 & & & & & \\ & 1 & & & & \\ & & 1 & & & \\ & & & 1 & & \\ & & & & 1 & \end{bmatrix}$$

Another Fiedler matrix: $M_\sigma = M_0 M_1 M_5 M_4 M_3 M_2$

$$= \begin{bmatrix} -a_5 & -a_4 & -a_3 & -a_2 & 1 & \\ 1 & & & & & \\ & 1 & & & & \\ & & 1 & & & \\ & & & -a_1 & & 1 \\ & & & -a_0 & & \end{bmatrix}$$

Structural property 1 of Fiedler matrices

Every Fiedler matrix has exactly the **same entries** as the first Frobenius companion matrix (in different positions).

Examples of Fiedler matrices (II)

$$p(z) = z^6 + a_5 z^5 + a_4 z^4 + a_3 z^3 + a_2 z^2 + a_1 z + a_0$$

Special Fiedler matrices: **Pentadiagonal matrices** (there are 4 for each degree n).

$$P_1 = (M_0 M_2 M_4)(M_1 M_3 M_5) = \begin{bmatrix} -a_5 & 1 & 0 & 0 & 0 & 0 \\ -a_4 & 0 & -a_3 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -a_2 & 0 & -a_1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -a_0 & 0 \end{bmatrix}$$

Structural property 2 of Fiedler matrices

Frobenius companion matrices are the Fiedler matrices with largest bandwidth and pentadiagonal Fiedler matrices are the ones with smallest bandwidth.

Examples of Fiedler matrices (II)

$$p(z) = z^6 + a_5 z^5 + a_4 z^4 + a_3 z^3 + a_2 z^2 + a_1 z + a_0$$

Special Fiedler matrices: **Pentadiagonal matrices** (there are 4 for each degree n).

$$P_1 = (M_0 M_2 M_4)(M_1 M_3 M_5) = \begin{bmatrix} -a_5 & 1 & 0 & 0 & 0 & 0 \\ -a_4 & 0 & -a_3 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -a_2 & 0 & -a_1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -a_0 & 0 \end{bmatrix}$$

Structural property 2 of Fiedler matrices

Frobenius companion matrices are the Fiedler matrices with largest bandwidth and pentadiagonal Fiedler matrices are the ones with smallest bandwidth.

Examples of Fiedler matrices (II)

$$p(z) = z^6 + a_5 z^5 + a_4 z^4 + a_3 z^3 + a_2 z^2 + a_1 z + a_0$$

Special Fiedler matrices: **Pentadiagonal matrices** (there are 4 for each degree n).

$$P_1 = (M_0 M_2 M_4)(M_1 M_3 M_5) = \begin{bmatrix} -a_5 & 1 & 0 & 0 & 0 & 0 \\ -a_4 & 0 & -a_3 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -a_2 & 0 & -a_1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -a_0 & 0 \end{bmatrix}$$

Structural property 2 of Fiedler matrices

Frobenius companion matrices are the Fiedler matrices with largest bandwidth and pentadiagonal Fiedler matrices are the ones with smallest bandwidth.

Number of different Fiedler matrices

Recall that the Fiedler matrix M_σ associated with a permutation σ of $(0, 1, \ldots, n-1)$ is

$$M_\sigma = M_{i_0} M_{i_1} \cdots M_{i_{n-1}}$$

But $M_i M_j = M_j M_i$, for $|i-j| \neq 1$, and many permutations lead to the same matrix.

This allows us to prove:

Lemma

There exist 2^{n-1} *different Fiedler matrices associated with a monic polynomial* $p(z)$ *of degree* n.

Recall that the Fiedler matrix M_σ associated with a permutation σ of $(0, 1, \ldots, n-1)$ is

$$M_\sigma = M_{i_0} M_{i_1} \cdots M_{i_{n-1}}$$

But $M_i M_j = M_j M_i$, for $|i - j| \neq 1$, and many permutations lead to the same matrix.

This allows us to prove:

Lemma

There exist 2^{n-1} *different Fiedler matrices associated with a monic polynomial* $p(z)$ *of degree* n.

Outline

Theorem

Let $M_\sigma \in \mathbb{C}^{n\times n}$ *be a Fiedler matrix of* $p(z)$, $E \in \mathbb{C}^{n\times n}$, *and*

$$p(z) := \det(zI - M_\sigma) = z^n + a_{n-1}\, z^{n-1} + \cdots + a_1\, z + \, a_0,$$
$$\widetilde{p}(z) := \det(zI - (M_\sigma + E)) = z^n + \widetilde{a}_{n-1}\, z^{n-1} + \cdots + \widetilde{a}_1\, z + \, \widetilde{a}_0.$$

Then, to first order in E*:*

$$\widetilde{a}_k - a_k = - \sum_{i,j=1}^{n} p_{ij}^{(\sigma,k)}(a_0, a_1, \ldots, a_{n-1})\, E_{ij} \quad \text{for } k = 0, 1, \ldots, n-1,$$

where the functions $p_{ij}^{(\sigma,k)}(a_0, a_1, \ldots, a_{n-1})$ *are* **multivariable polynomials in the coefficients of** $p(z)$ **given by...**

Theorem

Let $M_\sigma \in \mathbb{C}^{n\times n}$ be a Fiedler matrix of $p(z)$, $E \in \mathbb{C}^{n\times n}$, and

$$p(z) := \det(zI - M_\sigma) = z^n + a_{n-1}\, z^{n-1} + \cdots + a_1\, z + a_0,$$
$$\widetilde{p}(z) := \det(zI - (M_\sigma + E)) = z^n + \widetilde{a}_{n-1}\, z^{n-1} + \cdots + \widetilde{a}_1\, z + \widetilde{a}_0.$$

Then, to first order in E:

$$\widetilde{a}_k - a_k = -\sum_{i,j=1}^{n} p_{ij}^{(\sigma,k)}(a_0, a_1, \ldots, a_{n-1})\, E_{ij} \quad \text{for } k = 0, 1, \ldots, n-1,$$

where the functions $p_{ij}^{(\sigma,k)}(a_0, a_1, \ldots, a_{n-1})$ are **multivariable polynomials in the coefficients of $p(z)$ given by...**

(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 $k+1+i-n \leq \mathfrak{i}_\sigma(n-i:n-j-1) \leq k+1$;
- 0, otherwise;

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

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

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

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

- $$\sum_{l=\max\{0,k+1+j-\mathfrak{c}_\sigma(n-j:n-i-1)-n\}}^{l=\min\{k+1-\mathfrak{c}_\sigma(n-j:n-i-1),i-1\}} -\left(a_{n+1-i+l}\, a_{k+1-\mathfrak{c}_\sigma(n-j:n-i-1)-l}\right),$$

  if $j > i$ and $k+2+j-i-n \leq \mathfrak{c}_\sigma(n-j:n-i-1) \leq k+1$;

- $$\sum_{l=\max\{0,k+1+i-\mathfrak{i}_\sigma(n-i:n-j-1)-n\}}^{l=\min\{k+1-\mathfrak{i}_\sigma(n-i:n-j-1),j-1\}} -\left(a_{n+1-j+l}\, a_{k+1-\mathfrak{i}_\sigma(n-i:n-j-1)-l}\right),$$

  if $j < i$ and $k+2+i-j-n \leq \mathfrak{i}_\sigma(n-i:n-j-1) \leq k+1$;

- 0, otherwise.

Perturbation of the characteristic polynomial of a Fiedler matrix (II)

Theorem (Soft version)

Let $M_\sigma \in \mathbb{C}^{n\times n}$ *be a Fiedler matrix of* $p(z)$, $E \in \mathbb{C}^{n\times n}$, *and*

$$p(z) := \det(zI - M_\sigma) = z^n + a_{n-1}\, z^{n-1} + \cdots + a_1\, z + a_0,$$
$$\widetilde{p}(z) := \det(zI - (M_\sigma + E)) = z^n + \widetilde{a}_{n-1}\, z^{n-1} + \cdots + \widetilde{a}_1\, z + \widetilde{a}_0.$$

Then, to first order in E*:*

$$\widetilde{a}_k - a_k = -\sum_{i,j=1}^{n} p_{ij}^{(\sigma,k)}(a_0, a_1, \ldots, a_{n-1})\, E_{ij} \quad \text{for } k = 0, 1, \ldots, n-1,$$

where $p_{ij}^{(\sigma,k)}(a_0, a_1, \ldots, a_{n-1})$ *are* **multivariable polynomials such that**

- $p_{ij}^{(\sigma,k)}(a_0, a_1, \ldots, a_{n-1})$ *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.

Perturbation of the characteristic polynomial of a Fiedler matrix (II)

Theorem (Soft version)

Let $M_\sigma \in \mathbb{C}^{n\times n}$ *be a Fiedler matrix of* $p(z)$, $E \in \mathbb{C}^{n\times n}$, *and*

$$p(z) := \det(zI - M_\sigma) = z^n + a_{n-1}\, z^{n-1} + \cdots + a_1\, z + a_0,$$
$$\widetilde{p}(z) := \det(zI - (M_\sigma + E)) = z^n + \widetilde{a}_{n-1}\, z^{n-1} + \cdots + \widetilde{a}_1\, z + \widetilde{a}_0.$$

Then, to first order in E*:*

$$\widetilde{a}_k - a_k = -\sum_{i,j=1}^{n} p_{ij}^{(\sigma,k)}(a_0, a_1, \ldots, a_{n-1})\, E_{ij} \quad \text{for } k = 0, 1, \ldots, n-1,$$

where $p_{ij}^{(\sigma,k)}(a_0, a_1, \ldots, a_{n-1})$ *are* **multivariable polynomials such that**

- $p_{ij}^{(\sigma,k)}(a_0, a_1, \ldots, a_{n-1})$ *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.

Corollary

Let $M_\sigma \in \mathbb{C}^{n\times n}$ be a Fiedler matrix of $p(z)$, $E \in \mathbb{C}^{n\times n}$, and

$$p(z) := \det(zI - M_\sigma) = z^n + a_{n-1}\, z^{n-1} + \cdots + a_1\, z + a_0,$$
$$\widetilde{p}(z) := \det(zI - (M_\sigma + E)) = z^n + \widetilde{a}_{n-1}\, z^{n-1} + \cdots + \widetilde{a}_1\, z + \widetilde{a}_0.$$

If $\|E\|_2 = O(u)\,\alpha(p)\,\|M_\sigma\|_2$, *then*

- *For M_σ Frobenius companion matrix,*

$$\|\widetilde{p}(z) - p(z)\|_\infty = O(u)\,\alpha(p)\,\|p(z)\|_\infty^{\mathbf{2}}.$$

- *For M_σ NOT Frobenius companion matrix,*

$$\|\widetilde{p}(z) - p(z)\|_\infty = O(u)\,\alpha(p)\,\|p(z)\|_\infty^{\mathbf{3}}.$$

Remark: Only backward stability in polynomial root finding if $\|p(z)\|_\infty \approx 1$.

Penalty in polynomial backward errors from Fiedler matrices

Corollary

Let $M_\sigma \in \mathbb{C}^{n\times n}$ *be a Fiedler matrix of* $p(z)$, $E \in \mathbb{C}^{n\times n}$, *and*

$$p(z) := \det(zI - M_\sigma) = z^n + a_{n-1}\, z^{n-1} + \cdots + a_1\, z + a_0,$$
$$\widetilde{p}(z) := \det(zI - (M_\sigma + E)) = z^n + \widetilde{a}_{n-1}\, z^{n-1} + \cdots + \widetilde{a}_1\, z + \widetilde{a}_0.$$

If $\|E\|_2 = O(u)\,\alpha(p)\,\|M_\sigma\|_2$, *then*

- *For* M_σ *Frobenius companion matrix,*

$$\|\widetilde{p}(z) - p(z)\|_\infty = O(u)\,\alpha(p)\,\|p(z)\|_\infty^{\mathbf{2}}.$$

- *For* M_σ *NOT Frobenius companion matrix,*

$$\|\widetilde{p}(z) - p(z)\|_\infty = O(u)\,\alpha(p)\,\|p(z)\|_\infty^{\mathbf{3}}.$$

Remark: Only backward stability in polynomial root finding if $\|p(z)\|_\infty \approx 1$.

Corollary

Let $M_\sigma \in \mathbb{C}^{n\times n}$ *be a Fiedler matrix of* $p(z)$, $E \in \mathbb{C}^{n\times n}$, *and*

$$p(z) := \det(zI - M_\sigma) = z^n + a_{n-1}\, z^{n-1} + \cdots + a_1\, z + a_0,$$
$$\widetilde{p}(z) := \det(zI - (M_\sigma + E)) = z^n + \widetilde{a}_{n-1}\, z^{n-1} + \cdots + \widetilde{a}_1\, z + \widetilde{a}_0.$$

If $\|E\|_2 = O(u)\,\alpha(p)\,\|M_\sigma\|_2$, *then*

- *For* M_σ *Frobenius companion matrix,*

$$\|\widetilde{p}(z) - p(z)\|_\infty = O(u)\,\alpha(p)\,\|p(z)\|_\infty^{\mathbf{2}}.$$

- *For* M_σ *NOT Frobenius companion matrix,*

$$\|\widetilde{p}(z) - p(z)\|_\infty = O(u)\,\alpha(p)\,\|p(z)\|_\infty^{\mathbf{3}}.$$

Remark: Only backward stability in polynomial root finding if $\|p(z)\|_\infty \approx 1$.

- Let $p(z) = z^n + \sum_{i=0}^{n-1} a_i \, z^i$, with $\|p(z)\|_\infty > 1$.
- Then
$$t(z) := \beta^n \, p\left(\frac{z}{\beta}\right) = z^n + \sum_{i=0}^{n-1} (a_i \, \beta^{n-i}) \, z^i,$$
and it is inmediate to choose β such that $\max_i \left| a_i \, \beta^{n-i} \right| = 1$.
- Moreover,
$$t(z_0) = 0 \iff p\left(\frac{z_0}{\beta}\right) = 0 \, .$$
- But, Vanni Noferini pointed out that this process does not lead to "backward stability" in the original polynomial.
- More precisely,

$$\|\widetilde{t}(z) - t(z)\|_\infty = O(u) \Rightarrow \|\widetilde{p}(z) - p(z)\|_\infty = O(u) \, \max_i |\beta|^{i-n} = O(u) \, \left(\frac{1}{|\beta|}\right)^n .$$

Scaling does not work: a key remark by V. Noferini (2014)

- Let $p(z) = z^n + \sum_{i=0}^{n-1} a_i\, z^i$, with $\|p(z)\|_\infty > 1$.
- Then
$$t(z) := \beta^n\, p\left(\frac{z}{\beta}\right) = z^n + \sum_{i=0}^{n-1} (a_i\, \beta^{n-i})\, z^i,$$
and it is inmediate to choose β such that $\max_i \left|a_i\, \beta^{n-i}\right| = 1$.
- Moreover,
$$t(z_0) = 0 \Longleftrightarrow p\left(\frac{z_0}{\beta}\right) = 0\,.$$
- But, Vanni Noferini pointed out that this process does not lead to “backward stability” in the original polynomial.
- More precisely,

$$\|\widetilde{t}(z) - t(z)\|_\infty = O(u) \Rightarrow \|\widetilde{p}(z) - p(z)\|_\infty = O(u)\, \max_i |\beta|^{i-n} = O(u)\, \left(\frac{1}{|\beta|}\right)^n.$$

Scaling does not work: a key remark by V. Noferini (2014)

- Let $p(z) = z^n + \sum_{i=0}^{n-1} a_i\, z^i$, with $\|p(z)\|_\infty > 1$.
- Then

$$t(z) := \beta^n\, p\left(\frac{z}{\beta}\right) = z^n + \sum_{i=0}^{n-1} (a_i\, \beta^{n-i})\, z^i,$$

  and it is inmediate to choose β such that $\max_i \left|a_i\, \beta^{n-i}\right| = 1$.
- Moreover,

$$t(z_0) = 0 \Longleftrightarrow p\left(\frac{z_0}{\beta}\right) = 0\,.$$

- But, Vanni Noferini pointed out that this process does not lead to "backward stability" in the original polynomial.
- More precisely,

$$\|\widetilde{t}(z) - t(z)\|_\infty = O(u) \Rightarrow \|\widetilde{p}(z) - p(z)\|_\infty = O(u)\, \max_i |\beta|^{i-n} = O(u)\, \left(\frac{1}{|\beta|}\right)^n.$$

Scaling does not work: a key remark by V. Noferini (2014)

- Let $p(z) = z^n + \sum_{i=0}^{n-1} a_i\, z^i$, with $\|p(z)\|_\infty > 1$.
- Then
$$t(z) := \beta^n\, p\left(\frac{z}{\beta}\right) = z^n + \sum_{i=0}^{n-1} (a_i\, \beta^{n-i})\, z^i,$$
and it is inmediate to choose β such that $\max_i \left| a_i\, \beta^{n-i} \right| = 1$.
- Moreover,
$$t(z_0) = 0 \iff p\left(\frac{z_0}{\beta}\right) = 0\,.$$
- But, Vanni Noferini pointed out that this process does not lead to "backward stability" in the original polynomial.
- More precisely,

$$\|\widetilde{t}(z) - t(z)\|_\infty = O(u) \Rightarrow \|\widetilde{p}(z) - p(z)\|_\infty = O(u)\, \max_i |\beta|^{i-n} = O(u) \left(\frac{1}{|\beta|}\right)^n.$$

Scaling does not work: a key remark by V. Noferini (2014)

- Let $p(z) = z^n + \sum_{i=0}^{n-1} a_i\, z^i$, with $\|p(z)\|_\infty > 1$.
- Then

$$t(z) := \beta^n\, p\left(\frac{z}{\beta}\right) = z^n + \sum_{i=0}^{n-1} (a_i\, \beta^{n-i})\, z^i,$$

  and it is inmediate to choose β such that $\max_i \left| a_i\, \beta^{n-i} \right| = 1$.
- Moreover,

$$t(z_0) = 0 \Longleftrightarrow p\left(\frac{z_0}{\beta}\right) = 0\,.$$

- But, Vanni Noferini pointed out that this process does not lead to "backward stability" in the original polynomial.
- More precisely,

$$\|\widetilde{t}(z) - t(z)\|_\infty = O(u) \Rightarrow \|\widetilde{p}(z) - p(z)\|_\infty = O(u)\, \max_i |\beta|^{i-n} = O(u) \left(\frac{1}{|\beta|}\right)^n.$$

Outline

1. Perturbation of characteristic polynomials of general matrices
2. Antecedents: results for Frobenius companion matrices
3. Fiedler matrices: definition and properties
4. Backward errors of poly. root-finding from Fiedler matrices
5. **Balancing Fiedler matrices**
6. Numerical experiments
7. Backward errors of poly. root-finding from Fiedler PENCILS
8. Conclusions

Key points on balancing

- **Balancing any Fiedler matrix of $p(z)$ before applying QR yields** (very often) **perfect polynomial backward stability**:

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

- However, it is always possible to find $p(z)$ for which balancing does not improve backward stability.
- The theoretical treatment of "balancing" Fiedler matrices from the point of view of polynomial backward errors is trivial from our results, but
- the expressions we get are not useful to predict the backward errors.

Key points on balancing

- **Balancing any Fiedler matrix of $p(z)$ before applying QR yields** (very often) **perfect polynomial backward stability**:

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

- However, it is always possible to find $p(z)$ for which balancing does not improve backward stability.
- The theoretical treatment of "balancing" Fiedler matrices from the point of view of polynomial backward errors is trivial from our results, but
- the expressions we get are not useful to predict the backward errors.

Key points on balancing

- **Balancing any Fiedler matrix of $p(z)$ before applying QR yields** (very often) **perfect polynomial backward stability**:

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

- However, it is always possible to find $p(z)$ for which balancing does not improve backward stability.
- The theoretical treatment of “balancing” Fiedler matrices from the point of view of polynomial backward errors is trivial from our results, but
- the expressions we get are not useful to predict the backward errors.

- **Balancing any Fiedler matrix of $p(z)$ before applying QR yields** (very often) **perfect polynomial backward stability**:

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

- However, it is always possible to find $p(z)$ for which balancing does not improve backward stability.
- The theoretical treatment of "balancing" Fiedler matrices from the point of view of polynomial backward errors is trivial from our results, but
- the expressions we get are not useful to predict the backward errors.

How to deal with balancing?

- Balancing a Fiedler matrix M_σ of $p(z)$ consists in

$$M_\sigma \longrightarrow D M_\sigma D^{-1}, \quad \text{with } D = \text{diag}(2^{t_1}, \dots, 2^{t_n})$$

such that $\|\text{row}_i(D M_\sigma D^{-1})\|_\infty \approx \|\text{col}_i(D M_\sigma D^{-1})\|_\infty$ for all i.

- **Exact** computation with cost $O(n^2)$.
- QR on $D M_\sigma D^{-1}$ computes roots of $p(z)$ which are the exact eigenvalues of

$$D M_\sigma D^{-1} + \widetilde{E}, \quad \text{with} \quad \|\widetilde{E}\|_2 = O(u)\,\|D M_\sigma D^{-1}\|_2$$

- or, the exact roots of

$$\begin{aligned}\widetilde{p}(z) &= \det(zI - (D\, M_\sigma\, D^{-1} + \widetilde{E})) \\ &= \det(zI - (M_\sigma + D^{-1} \widetilde{E} D))\end{aligned}$$

- **We have already solved this problem!!**

How to deal with balancing?

- Balancing a Fiedler matrix M_σ of $p(z)$ consists in

$$M_\sigma \longrightarrow D M_\sigma D^{-1}, \quad \text{with } D = \text{diag}(2^{t_1}, \dots, 2^{t_n})$$

such that $\|\text{row}_i(DM_\sigma D^{-1})\|_\infty \approx \|\text{col}_i(DM_\sigma D^{-1})\|_\infty$ for all i.

- **Exact** computation with cost $O(n^2)$.
- QR on $DM_\sigma D^{-1}$ computes roots of $p(z)$ which are the exact eigenvalues of

$$DM_\sigma D^{-1} + \widetilde{E}, \quad \text{with} \quad \|\widetilde{E}\|_2 = O(u)\,\|DM_\sigma D^{-1}\|_2$$

- or, the exact roots of

$$\begin{aligned}\widetilde{p}(z) &= \det(zI - (D\,M_\sigma\,D^{-1} + \widetilde{E})) \\ &= \det(zI - (M_\sigma + D^{-1}\widetilde{E}D))\end{aligned}$$

- **We have already solved this problem!!**

How to deal with balancing?

- Balancing a Fiedler matrix M_σ of $p(z)$ consists in

$$M_\sigma \longrightarrow D M_\sigma D^{-1}, \quad \text{with } D = \text{diag}(2^{t_1}, \ldots, 2^{t_n})$$

such that $\|\text{row}_i(D M_\sigma D^{-1})\|_\infty \approx \|\text{col}_i(D M_\sigma D^{-1})\|_\infty$ for all i.

- **Exact** computation with cost $O(n^2)$.
- QR on $D M_\sigma D^{-1}$ computes roots of $p(z)$ which are the exact eigenvalues of

$$D M_\sigma D^{-1} + \widetilde{E}, \quad \text{with} \quad \|\widetilde{E}\|_2 = O(u)\, \|D M_\sigma D^{-1}\|_2$$

- or, the exact roots of

$$\begin{aligned} \widetilde{p}(z) &= \det(zI - (D\, M_\sigma\, D^{-1} + \widetilde{E})) \\ &= \det(zI - (M_\sigma + D^{-1} \widetilde{E} D)) \end{aligned}$$

- **We have already solved this problem!!**

- Balancing a Fiedler matrix M_σ of $p(z)$ consists in

$$M_\sigma \longrightarrow D M_\sigma D^{-1}, \quad \text{with } D = \text{diag}(2^{t_1}, \ldots, 2^{t_n})$$

such that $\|\text{row}_i(D M_\sigma D^{-1})\|_\infty \approx \|\text{col}_i(D M_\sigma D^{-1})\|_\infty$ for all i.

- **Exact** computation with cost $O(n^2)$.
- QR on $D M_\sigma D^{-1}$ computes roots of $p(z)$ which are the exact eigenvalues of

$$D M_\sigma D^{-1} + \widetilde{E}, \quad \text{with} \quad \|\widetilde{E}\|_2 = O(u)\, \|D M_\sigma D^{-1}\|_2$$

- or, the exact roots of

$$\begin{aligned} \widetilde{p}(z) &= \det(zI - (D\, M_\sigma\, D^{-1} + \widetilde{E})) \\ &= \det(zI - (M_\sigma + D^{-1} \widetilde{E} D)) \end{aligned}$$

- We have already solved this problem!!

How to deal with balancing?

- Balancing a Fiedler matrix M_σ of $p(z)$ consists in

$$M_\sigma \longrightarrow D M_\sigma D^{-1}, \quad \text{with } D = \text{diag}(2^{t_1}, \dots, 2^{t_n})$$

  such that $\|\text{row}_i(D M_\sigma D^{-1})\|_\infty \approx \|\text{col}_i(D M_\sigma D^{-1})\|_\infty$ for all i.

- **Exact** computation with cost $O(n^2)$.
- QR on $D M_\sigma D^{-1}$ computes roots of $p(z)$ which are the exact eigenvalues of

$$D M_\sigma D^{-1} + \widetilde{E}, \quad \text{with} \quad \|\widetilde{E}\|_2 = O(u)\, \|D M_\sigma D^{-1}\|_2$$

- or, the exact roots of

$$\begin{aligned} \widetilde{p}(z) &= \det(zI - (D\, M_\sigma\, D^{-1} + \widetilde{E})) \\ &= \det(zI - (M_\sigma + D^{-1} \widetilde{E} D)) \end{aligned}$$

- **We have already solved this problem!!**

Theorem

Let M_σ be a Fiedler matrix of $p(z)$, D its diagonal balancing matrix, $\widetilde{E} \in \mathbb{C}^{n\times n}$, and

$$p(z) := \det(zI - M_\sigma) = z^n + a_{n-1}\, z^{n-1} + \cdots + a_1\, z + a_0,$$
$$\widetilde{p}(z) := \det(zI - (M_\sigma + D^{-1}\widetilde{E}D)) = z^n + \widetilde{a}_{n-1}\, z^{n-1} + \cdots + \widetilde{a}_1\, z + \widetilde{a}_0.$$

Then, to first order in $\widetilde{E}$:

$$\widetilde{a}_k - a_k = -\sum_{i,j=1}^{n} p_{ij}^{(\sigma,k)}(a_0, a_1, \ldots, a_{n-1})\, \frac{d_j}{d_i}\, \widetilde{E}_{ij} \quad \text{for } k = 0, 1, \ldots, n-1,$$

where $p_{ij}^{(\sigma,k)}(a_0, a_1, \ldots, a_{n-1})$ are the previous multivariable polynomials.

Moreover, if $\|\widetilde{E}\|_2 = O(u)\, \|DM_\sigma D^{-1}\|_2$, then

$$\|\widetilde{p}(z) - p(z)\|_\infty = O(u)\, \max_{i,j,k}\left(\left| p_{ij}^{(\sigma,k)}(a_0, \ldots, a_{n-1})\frac{d_j}{d_i}\right|\right)\, \|DM_\sigma D^{-1}\|_2.$$

The effect of balancing on polynomial backward error

Theorem

Let M_σ be a Fiedler matrix of $p(z)$, D its diagonal balancing matrix, $\widetilde{E} \in \mathbb{C}^{n\times n}$, and

$$p(z) := \det(zI - M_\sigma) = z^n + a_{n-1}\, z^{n-1} + \cdots + a_1\, z + a_0,$$
$$\widetilde{p}(z) := \det(zI - (M_\sigma + D^{-1}\widetilde{E}D)) = z^n + \widetilde{a}_{n-1}\, z^{n-1} + \cdots + \widetilde{a}_1\, z + \widetilde{a}_0.$$

Then, to first order in $\widetilde{E}$:

$$\widetilde{a}_k - a_k = -\sum_{i,j=1}^{n} p_{ij}^{(\sigma,k)}(a_0, a_1, \ldots, a_{n-1})\, \frac{d_j}{d_i}\, \widetilde{E}_{ij} \quad \text{for } k = 0, 1, \ldots, n-1,$$

where $p_{ij}^{(\sigma,k)}(a_0, a_1, \ldots, a_{n-1})$ are the previous multivariable polynomials.

Moreover, if $\|\widetilde{E}\|_2 = O(u)\,\|DM_\sigma D^{-1}\|_2$, then

$$\|\widetilde{p}(z) - p(z)\|_\infty = O(u)\, \max_{i,j,k} \left(\left| p_{ij}^{(\sigma,k)}(a_0, \ldots, a_{n-1}) \frac{d_j}{d_i} \right| \right) \|DM_\sigma D^{-1}\|_2.$$

...but we cannot go further

$$\|\widetilde{p}(z) - p(z)\|_\infty = O(u) \max_{i,j,k} \left(\left| p_{ij}^{(\sigma,k)}(a_0, \ldots, a_{n-1}) \frac{d_j}{d_i} \right| \right) \; \|DM_\sigma D^{-1}\|_2$$

- because D is a very complicated function of $a_0, \ldots, a_{n-1}$, so
- **we cannot estimate** neither

$$\max_{i,j,k} \left(\left| p_{ij}^{(\sigma,k)}(a_0, \ldots, a_{n-1}) \frac{d_j}{d_i} \right| \right)$$

- nor

$$\|DM_\sigma D^{-1}\|_2$$

  a priori,
- while without balancing

$$\max_{i,j,k} \left(\left| p_{ij}^{(\sigma,k)}(a_0, \ldots, a_{n-1}) \right| \right) \leq n \|p(z)\|_\infty^2, \qquad \|M_\sigma\|_2 \approx \|p(z)\|_\infty .$$

...but we cannot go further

$$\|\widetilde{p}(z) - p(z)\|_\infty = O(u) \max_{i,j,k} \left(\left| p_{ij}^{(\sigma,k)}(a_0, \dots, a_{n-1}) \frac{d_j}{d_i} \right| \right) \; \|D M_\sigma D^{-1}\|_2$$

- because D is a very complicated function of $a_0, \dots, a_{n-1}$, so
- **we cannot estimate** neither

$$\max_{i,j,k} \left(\left| p_{ij}^{(\sigma,k)}(a_0, \dots, a_{n-1}) \frac{d_j}{d_i} \right| \right)$$

- nor

$$\|D M_\sigma D^{-1}\|_2$$

  a priori,
- while without balancing

$$\max_{i,j,k} \left(\left| p_{ij}^{(\sigma,k)}(a_0, \dots, a_{n-1}) \right| \right) \le n \|p(z)\|_\infty^2, \qquad \|M_\sigma\|_2 \approx \|p(z)\|_\infty .$$

...but we cannot go further

$$\|\widetilde{p}(z) - p(z)\|_\infty = O(u) \max_{i,j,k} \left(\left| p_{ij}^{(\sigma,k)}(a_0, \ldots, a_{n-1}) \frac{d_j}{d_i} \right| \right) \; \|D M_\sigma D^{-1}\|_2$$

- because D is a very complicated function of $a_0, \ldots, a_{n-1}$, so
- **we cannot estimate** neither

$$\max_{i,j,k} \left(\left| p_{ij}^{(\sigma,k)}(a_0, \ldots, a_{n-1}) \frac{d_j}{d_i} \right| \right)$$

- nor

$$\|D M_\sigma D^{-1}\|_2$$

a priori,

- while without balancing

$$\max_{i,j,k} \left(\left| p_{ij}^{(\sigma,k)}(a_0, \ldots, a_{n-1}) \right| \right) \leq n \|p(z)\|_\infty^2, \qquad \|M_\sigma\|_2 \approx \|p(z)\|_\infty .$$

$$\|\widetilde{p}(z) - p(z)\|_\infty = O(u) \max_{i,j,k} \left(\left| p_{ij}^{(\sigma,k)}(a_0,\dots,a_{n-1}) \frac{d_j}{d_i} \right| \right) \|D M_\sigma D^{-1}\|_2$$

- because D is a very complicated function of $a_0,\dots,a_{n-1}$, so
- **we cannot estimate** neither

$$\max_{i,j,k} \left(\left| p_{ij}^{(\sigma,k)}(a_0,\dots,a_{n-1}) \frac{d_j}{d_i} \right| \right)$$

- nor

$$\|D M_\sigma D^{-1}\|_2$$

a priori,

- while without balancing

$$\max_{i,j,k} \left(\left| p_{ij}^{(\sigma,k)}(a_0,\dots,a_{n-1}) \right| \right) \le n \|p(z)\|_\infty^2, \qquad \|M_\sigma\|_2 \approx \|p(z)\|_\infty .$$

...but we cannot go further

$$\|\widetilde{p}(z) - p(z)\|_\infty = O(u) \max_{i,j,k} \left(\left| p_{ij}^{(\sigma,k)}(a_0, \ldots, a_{n-1}) \frac{d_j}{d_i} \right| \right) \|D M_\sigma D^{-1}\|_2$$

- because D is a very complicated function of $a_0, \ldots, a_{n-1}$, so
- **we cannot estimate** neither

$$\max_{i,j,k} \left(\left| p_{ij}^{(\sigma,k)}(a_0, \ldots, a_{n-1}) \frac{d_j}{d_i} \right| \right)$$

- nor

$$\|D M_\sigma D^{-1}\|_2$$

  a priori,
- while without balancing

$$\max_{i,j,k} \left(\left| p_{ij}^{(\sigma,k)}(a_0, \ldots, a_{n-1}) \right| \right) \leq n \|p(z)\|_\infty^2, \qquad \|M_\sigma\|_2 \approx \|p(z)\|_\infty .$$

Outline

Goals and design of numerical experiments

- The goals of the numerical experiments are
  1. to show that our bounds correctly predict the dependence on the norm of $p(z)$ of the polynomial backward errors when the roots are computed as the eigenvalues of a Fiedler matrix with QR, and
  2. to study the effect of balancing the Fiedler companion matrices.
- We proceed as follows:
  1. We generate 500 random monic polys of degree 20 for each fixed value $\|p\|_\infty$.
  2. We compute exactly (in quadruple precision) the polynomial backward error corresponding to the roots computed by QR.
  3. We do this for four different Fiedler matrices
     - M_{σ_1} = second classical Frobenius,
     - M_{σ_2} = a pentadiagonal,
     - M_{σ_3} = the second F-matrix,
     - M_{σ_4} = "another one".

- The goals of the numerical experiments are
  1. to show that our bounds correctly predict the dependence on the norm of $p(z)$ of the polynomial backward errors when the roots are computed as the eigenvalues of a Fiedler matrix with QR, and
  2. to study the effect of balancing the Fiedler companion matrices.
- We proceed as follows:
  1. We generate 500 random monic polys of degree 20 for each fixed value $\|p\|_\infty$.
  2. We compute exactly (in quadruple precision) the polynomial backward error corresponding to the roots computed by QR.
  3. We do this for four different Fiedler matrices
     - M_{σ_1} = second classical Frobenius,
     - M_{σ_2} = a pentadiagonal,
     - M_{σ_3} = the second F-matrix,
     - M_{σ_4} = "another one".

Goals and design of numerical experiments

- The goals of the numerical experiments are
  1. to show that our bounds correctly predict the dependence on the norm of $p(z)$ of the polynomial backward errors when the roots are computed as the eigenvalues of a Fiedler matrix with QR, and
  2. to study the effect of balancing the Fiedler companion matrices.
- We proceed as follows:
  1. We generate 500 random monic polys of degree 20 for each fixed value $\|p\|_\infty$.
  2. We compute exactly (in quadruple precision) the polynomial backward error corresponding to the roots computed by QR.
  3. We do this for four different Fiedler matrices
     - M_{σ_1} = second classical Frobenius,
     - M_{σ_2} = a pentadiagonal,
     - M_{σ_3} = the second F-matrix,
     - M_{σ_4} = "another one".

Goals and design of numerical experiments

- The goals of the numerical experiments are
  1. to show that our bounds correctly predict the dependence on the norm of $p(z)$ of the polynomial backward errors when the roots are computed as the eigenvalues of a Fiedler matrix with QR, and
  2. to study the effect of balancing the Fiedler companion matrices.
- We proceed as follows:
  1. We generate 500 random monic polys of degree 20 for each fixed value $\|p\|_\infty$.
  2. We compute exactly (in quadruple precision) the polynomial backward error corresponding to the roots computed by QR.
  3. We do this for four different Fiedler matrices
     - M_{σ_1} = second classical Frobenius,
     - M_{σ_2} = a pentadiagonal,
     - M_{σ_3} = the second F-matrix,
     - M_{σ_4} = “another one”.

Goals and design of numerical experiments

- The goals of the numerical experiments are
  1. to show that our bounds correctly predict the dependence on the norm of $p(z)$ of the polynomial backward errors when the roots are computed as the eigenvalues of a Fiedler matrix with QR, and
  2. to study the effect of balancing the Fiedler companion matrices.
- We proceed as follows:
  1. We generate 500 random monic polys of degree 20 for each fixed value $\|p\|_\infty$.
  2. We compute exactly (in quadruple precision) the polynomial backward error corresponding to the roots computed by QR.
  3. We do this for four different Fiedler matrices
     - M_{σ_1} = second classical Frobenius,
     - M_{σ_2} = a pentadiagonal,
     - M_{σ_3} = the second F-matrix,
     - M_{σ_4} = "another one".

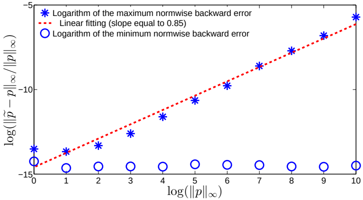
Goals and design of numerical experiments

- The goals of the numerical experiments are
  1. to show that our bounds correctly predict the dependence on the norm of $p(z)$ of the polynomial backward errors when the roots are computed as the eigenvalues of a Fiedler matrix with QR, and
  2. to study the effect of balancing the Fiedler companion matrices.
- We proceed as follows:
  1. We generate 500 random monic polys of degree 20 for each fixed value $\|p\|_\infty$.
  2. We compute exactly (in quadruple precision) the polynomial backward error corresponding to the roots computed by QR.
  3. We do this for four different Fiedler matrices
     - M_{σ_1} = second classical Frobenius,
     - M_{σ_2} = a pentadiagonal,
     - M_{σ_3} = the second F-matrix,
     - M_{σ_4} = "another one".

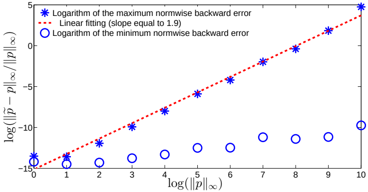
Goals and design of numerical experiments

- The goals of the numerical experiments are
  1. to show that our bounds correctly predict the dependence on the norm of $p(z)$ of the polynomial backward errors when the roots are computed as the eigenvalues of a Fiedler matrix with QR, and
  2. to study the effect of balancing the Fiedler companion matrices.
- We proceed as follows:
  1. We generate 500 random monic polys of degree 20 for each fixed value $\|p\|_\infty$.
  2. We compute exactly (in quadruple precision) the polynomial backward error corresponding to the roots computed by QR.
  3. We do this for four different Fiedler matrices
     - M_{σ_1} = second classical Frobenius,
     - M_{σ_2} = a pentadiagonal,
     - M_{σ_3} = the second F-matrix,
     - M_{σ_4} = “another one”.

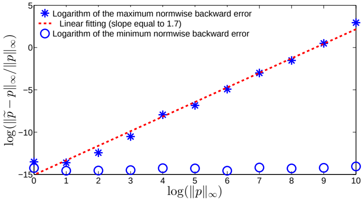
Numerical experiments (without balancing)



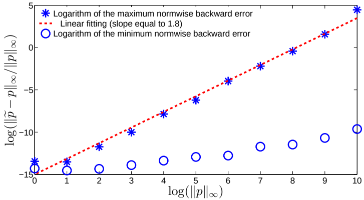
(a) M_{σ_1}



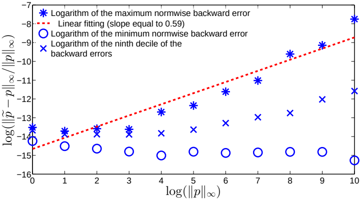
(b) M_{σ_2}



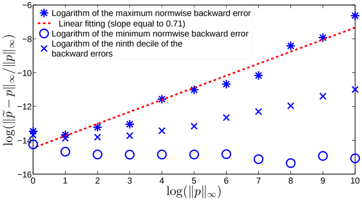
(c) M_{σ_3}



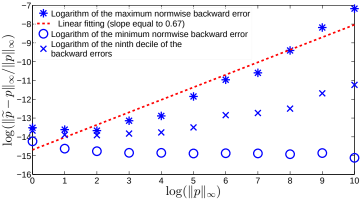
(d) M_{σ_4}



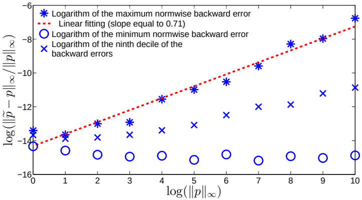
(e) M_{σ_1}



(f) M_{σ_2}



(g) M_{σ_3}



(h) M_{σ_4}

Outline

Theorem (Corollary of Theorem Jacobi)

Let $A, B, E, G \in \mathbb{C}^{n\times n}$, *and*

$$q(z) := \det(zB - A) = b_n\, z^n + b_{n-1}\, z^{n-1} + \cdots + b_1\, z + \, b_0,$$
$$\widetilde{q}(z) := \det(z(B+G) - (A+E)) = \widetilde{b}_n z^n + \widetilde{b}_{n-1}\, z^{n-1} + \cdots + \widetilde{b}_1\, z + \widetilde{b}_0.$$

Then

1. $$\widetilde{q}(z) - q(z) = \operatorname{trace}(\, \operatorname{adj}(zB - A)\,(zG - E)\,) + O(\|[E\, G]\|^2),$$

2. *and, if* $\operatorname{adj}(zB - A) = \sum_{k=0}^{n-1} z^k P_k$, *where* $P_k \in \mathbb{C}^{n\times n}$,

$$\widetilde{b}_k - b_k := \operatorname{trace}(\, P_{k-1}\, G - P_k\, E) + O(\|[E\, G]\|^2), \quad \textit{for } k = 0, 1, \ldots, n,$$

with $P_n = P_{-1} := 0$.

Theorem (Corollary of Theorem Jacobi)

Let $A, B, E, G \in \mathbb{C}^{n\times n}$, *and*

$$q(z) := \det(zB - A) = b_n\, z^n + b_{n-1}\, z^{n-1} + \cdots + b_1\, z + \, b_0,$$
$$\widetilde{q}(z) := \det(z(B+G) - (A+E)) = \widetilde{b}_n z^n + \widetilde{b}_{n-1}\, z^{n-1} + \cdots + \widetilde{b}_1\, z + \widetilde{b}_0.$$

Then

1. $$\widetilde{q}(z) - q(z) = \operatorname{trace}(\,\operatorname{adj}(zB - A)\,(zG - E)\,) + O(\|[E\, G]\|^2),$$

2. *and, if* $\operatorname{adj}(zB - A) = \sum_{k=0}^{n-1} z^k P_k$, *where* $P_k \in \mathbb{C}^{n\times n}$,

$$\widetilde{b}_k - b_k := \operatorname{trace}(\,P_{k-1}\, G - P_k\, E) + O(\|[E\, G]\|^2), \quad \textit{for } k = 0, 1, \ldots, n,$$

with $P_n = P_{-1} := 0.$

Theorem (Corollary of Theorem Jacobi)

Let $A, B, E, G \in \mathbb{C}^{n\times n}$, *and*

$$q(z) := \det(zB - A) = b_n\, z^n + b_{n-1}\, z^{n-1} + \cdots + b_1\, z + \, b_0,$$
$$\widetilde{q}(z) := \det(z(B+G) - (A+E)) = \widetilde{b}_n z^n + \widetilde{b}_{n-1}\, z^{n-1} + \cdots + \widetilde{b}_1\, z + \widetilde{b}_0.$$

Then

1. $$\widetilde{q}(z) - q(z) = \operatorname{trace}(\,\operatorname{adj}(zB - A)\,(zG - E)\,) + O(\|[E\,G]\|^2),$$
2. *and, if* $\operatorname{adj}(zB - A) = \sum_{k=0}^{n-1} z^k P_k$, *where* $P_k \in \mathbb{C}^{n\times n}$,

   $$\widetilde{b}_k - b_k := \operatorname{trace}(\,P_{k-1}\,G - P_k\,E) + O(\|[E\,G]\|^2), \quad \textit{for}\;\; k = 0, 1, \ldots, n,$$

   with $P_n = P_{-1} := 0$.

The case of Fiedler pencils

- So, the key to bound $|\widetilde{b}_k - b_k|$ is to get expressions (or bounds on their norms) for the matrices P_k in $\mathrm{adj}(zB - A) = \sum_{k=0}^{n-1} z^k P_k$.
- In the particular case of Fiedler companion pencils $F_\sigma(q)$ of $q(z)$ (including the classical Frobenius pencils), this problem can be reduced to the already solved case of Fielder matrices M_σ as follows.
- Define from $q(z)$ the monic polynomial $p(z) := q(z)/b_n$. Then, it can be proved

$$F_\sigma(q) = S_\sigma(zI - M_\sigma(p))T_\sigma, \quad \text{with}$$

$$S_\sigma := \mathrm{diag}(b_n, s_2, \ldots, s_n), \quad s_i = 1 \; or \; b_n\,,$$

$$T_\sigma := \mathrm{diag}(1\,, t_2, \ldots, t_n), \quad t_i = 1 \; or \; b_n^{-1}\,,$$

  and $S_\sigma T_\sigma = \mathrm{diag}(b_n, 1, \ldots, 1)$.
- From here,

$$\mathrm{adj} F_\sigma(q) = \mathrm{adj}(T_\sigma)\,\mathrm{adj}(zI - M_\sigma(p))\,\mathrm{adj}(S_\sigma)\,.$$

The case of Fiedler pencils

- So, the key to bound $|\widetilde{b}_k - b_k|$ is to get expressions (or bounds on their norms) for the matrices P_k in $\mathrm{adj}(zB - A) = \sum_{k=0}^{n-1} z^k P_k$.
- In the particular case of Fiedler companion pencils $F_\sigma(q)$ of $q(z)$ (including the classical Frobenius pencils), this problem can be reduced to the already solved case of Fielder matrices M_σ as follows.
- Define from $q(z)$ the monic polynomial $p(z) := q(z)/b_n$. Then, it can be proved

$$F_\sigma(q) = S_\sigma(zI - M_\sigma(p))T_\sigma, \quad \text{with}$$

$$S_\sigma := \mathrm{diag}(b_n, s_2, \ldots, s_n), \quad s_i = 1 \; or \; b_n \,,$$

$$T_\sigma := \mathrm{diag}(1\,, t_2, \ldots, t_n), \quad t_i = 1 \; or \; b_n^{-1} \,,$$

  and $S_\sigma T_\sigma = \mathrm{diag}(b_n, 1, \ldots, 1)$.
- From here,

$$\mathrm{adj} F_\sigma(q) = \mathrm{adj}(T_\sigma)\,\mathrm{adj}(zI - M_\sigma(p))\,\mathrm{adj}(S_\sigma)\,.$$

The case of Fiedler pencils

- So, the key to bound $|\widetilde{b}_k - b_k|$ is to get expressions (or bounds on their norms) for the matrices P_k in $\mathrm{adj}(zB - A) = \sum_{k=0}^{n-1} z^k P_k$.
- In the particular case of Fiedler companion pencils $F_\sigma(q)$ of $q(z)$ (including the classical Frobenius pencils), this problem can be reduced to the already solved case of Fielder matrices M_σ as follows.
- Define from $q(z)$ the monic polynomial $p(z) := q(z)/b_n$. Then, it can be proved

$$F_\sigma(q) = S_\sigma(zI - M_\sigma(p))T_\sigma, \quad \text{with}$$

$$S_\sigma := \mathrm{diag}(b_n, s_2, \ldots, s_n), \quad s_i = 1 \; or \; b_n \,,$$

$$T_\sigma := \mathrm{diag}(1\,, t_2, \ldots, t_n), \quad t_i = 1 \; or \; b_n^{-1} \,,$$

  and $S_\sigma T_\sigma = \mathrm{diag}(b_n, 1, \ldots, 1)$.
- From here,

$$\mathrm{adj} F_\sigma(q) = \mathrm{adj}(T_\sigma)\,\mathrm{adj}(zI - M_\sigma(p))\,\mathrm{adj}(S_\sigma)\,.$$

- So, the key to bound $|\widetilde{b}_k - b_k|$ is to get expressions (or bounds on their norms) for the matrices P_k in $\text{adj}(zB - A) = \sum_{k=0}^{n-1} z^k P_k$.
- In the particular case of Fiedler companion pencils $F_\sigma(q)$ of $q(z)$ (including the classical Frobenius pencils), this problem can be reduced to the already solved case of Fielder matrices M_σ as follows.
- Define from $q(z)$ the monic polynomial $p(z) := q(z)/b_n$. Then, it can be proved

$$F_\sigma(q) = S_\sigma(zI - M_\sigma(p))T_\sigma, \quad \text{with}$$

$$S_\sigma := \text{diag}(b_n, s_2, \ldots, s_n), \quad s_i = 1 \; or \; b_n \,,$$

$$T_\sigma := \text{diag}(1\,, t_2, \ldots, t_n), \quad t_i = 1 \; or \; b_n^{-1} \,,$$

  and $S_\sigma T_\sigma = \text{diag}(b_n, 1, \ldots, 1)$.
- From here,

$$\text{adj}F_\sigma(q) = \text{adj}(T_\sigma)\,\text{adj}(zI - M_\sigma(p))\,\text{adj}(S_\sigma)\,.$$

Final backward-error (perturbation) result for Fiedler pencils

Corollary

Let $F_\sigma(z) = zM_n - M_\sigma$ be a Fiedler pencil of $q(z)$, $E, G \in \mathbb{C}^{n\times n}$, and

$$q(z) := \det(zM_n - M_\sigma) = b_n\, z^n + b_{n-1}\, z^{n-1} + \cdots + b_1\, z + b_0,$$

$$\widetilde{q}(z) := \det(z(M_n + G) - (M_\sigma + E)) = \widetilde{b}_n\, z^n + \widetilde{b}_{n-1}\, z^{n-1} + \cdots + \widetilde{b}_1\, z + \widetilde{b}_0.$$

If $\|G\|_2 = O(u)\,\|M_n\|_2$ and $\|E\|_2 = O(u)\,\|M_\sigma\|_2$, then,

- *for $F_\sigma(z)$ a Frobenius companion pencil,*

$$\|\widetilde{q}(z) - q(z)\|_\infty = O(u)\ \max\{1, \|q(z)\|_\infty^{\mathbf{2}}\}.$$

- *for $F_\sigma(z)$ NOT Frobenius companion pencil,*

$$\|\widetilde{q}(z) - q(z)\|_\infty = O(u)\ \max\{1, \|q(z)\|_\infty^{\mathbf{3}}\}.$$

Remark: Backward stability for normalized polynomials $\|q(z)\|_\infty = 1$ and we can normalize!!!!!

Corollary

Let $F_\sigma(z) = zM_n - M_\sigma$ *be a Fiedler pencil of* $q(z)$, $E, G \in \mathbb{C}^{n\times n}$, *and*

$$q(z) := \det(zM_n - M_\sigma) = b_n\, z^n + b_{n-1}\, z^{n-1} + \cdots + b_1\, z + b_0,$$
$$\widetilde{q}(z) := \det(z(M_n + G) - (M_\sigma + E)) = \widetilde{b}_n\, z^n + \widetilde{b}_{n-1}\, z^{n-1} + \cdots + \widetilde{b}_1\, z + \widetilde{b}_0.$$

If $\|G\|_2 = O(u)\,\|M_n\|_2$ *and* $\|E\|_2 = O(u)\,\|M_\sigma\|_2$, *then,*

- *for* $F_\sigma(z)$ *a Frobenius companion pencil,*

$$\|\widetilde{q}(z) - q(z)\|_\infty = O(u)\,\max\{1, \|q(z)\|_\infty^{\mathbf{2}}\}.$$

- *for* $F_\sigma(z)$ *NOT Frobenius companion pencil,*

$$\|\widetilde{q}(z) - q(z)\|_\infty = O(u)\,\max\{1, \|q(z)\|_\infty^{\mathbf{3}}\}.$$

Remark: Backward stability for normalized polynomials $\|q(z)\|_\infty = 1$

and we can normalize!!!!!

Final backward-error (perturbation) result for Fiedler pencils

Corollary

Let $F_\sigma(z) = zM_n - M_\sigma$ *be a Fiedler pencil of* $q(z)$, $E, G \in \mathbb{C}^{n\times n}$, *and*

$$q(z) := \det(zM_n - M_\sigma) = b_n\, z^n + b_{n-1}\, z^{n-1} + \cdots + b_1\, z + b_0,$$

$$\widetilde{q}(z) := \det(z(M_n + G) - (M_\sigma + E)) = \widetilde{b}_n\, z^n + \widetilde{b}_{n-1}\, z^{n-1} + \cdots + \widetilde{b}_1\, z + \widetilde{b}_0.$$

If $\|G\|_2 = O(u)\,\|M_n\|_2$ *and* $\|E\|_2 = O(u)\,\|M_\sigma\|_2$, *then,*

- *for* $F_\sigma(z)$ *a Frobenius companion pencil,*

$$\|\widetilde{q}(z) - q(z)\|_\infty = O(u)\ \max\{1, \|q(z)\|_\infty^{\mathbf{2}}\}.$$

- *for* $F_\sigma(z)$ *NOT Frobenius companion pencil,*

$$\|\widetilde{q}(z) - q(z)\|_\infty = O(u)\ \max\{1, \|q(z)\|_\infty^{\mathbf{3}}\}.$$

Remark: Backward stability for normalized polynomials $\|q(z)\|_\infty = 1$

and we can normalize!!!!!

Outline

Conclusions

- Assume that we apply to Fiedler and classical Frobenius companion **matrices** of a monic polynomial $p(z)$ the "same eigenvalue algorithm" (or algorithms with similar matrix backward stability properties) for computing its roots.
- Proved: these approaches do NOT lead to guaranteed polynomial backward stability, but from the point of view of polynomial backw-errors:
- Proved: Unbalanced Fiedler matrices are as good as classical Frobenius companion matrices if $\|p(z)\|_\infty$ is moderate.
- Proved: Unbalanced Fiedler matrices are worse than classical Frobenius companion matrices if $\|p(z)\|_\infty \gg 1$, but both are bad.
- From numerical experiments: Balanced Fiedler matrices are as good as classical Frobenius companion matrices always, but none of them are always good.
- Proved: with Fiedler and classical Frobenius companion **pencils+QZ** **perfect polynomial backward stability is guaranteed**, but it is computationally more expensive (at present) and the effect of conditioning needs to be investigated.

Conclusions

- Assume that we apply to Fiedler and classical Frobenius companion **matrices** of a monic polynomial $p(z)$ the "same eigenvalue algorithm" (or algorithms with similar matrix backward stability properties) for computing its roots.
- Proved: these approaches do NOT lead to guaranteed polynomial backward stability, but from the point of view of polynomial backw-errors:
- Proved: Unbalanced Fiedler matrices are as good as classical Frobenius companion matrices if $\|p(z)\|_\infty$ is moderate.
- Proved: Unbalanced Fiedler matrices are worse than classical Frobenius companion matrices if $\|p(z)\|_\infty \gg 1$, but both are bad.
- From numerical experiments: Balanced Fiedler matrices are as good as classical Frobenius companion matrices always, but none of them are always good.
- Proved: with Fiedler and classical Frobenius companion **pencils+QZ perfect polynomial backward stability is guaranteed**, but it is computationally more expensive (at present) and the effect of conditioning needs to be investigated.

Conclusions

- Assume that we apply to Fiedler and classical Frobenius companion **matrices** of a monic polynomial $p(z)$ the "same eigenvalue algorithm" (or algorithms with similar matrix backward stability properties) for computing its roots.
- Proved: these approaches do NOT lead to guaranteed polynomial backward stability, but from the point of view of polynomial backw-errors:
- Proved: Unbalanced Fiedler matrices are as good as classical Frobenius companion matrices if $\|p(z)\|_\infty$ is moderate.
- Proved: Unbalanced Fiedler matrices are worse than classical Frobenius companion matrices if $\|p(z)\|_\infty \gg 1$, but both are bad.
- From numerical experiments: Balanced Fiedler matrices are as good as classical Frobenius companion matrices always, but none of them are always good.
- Proved: with Fiedler and classical Frobenius companion **pencils+QZ** **perfect polynomial backward stability is guaranteed**, but it is computationally more expensive (at present) and the effect of conditioning needs to be investigated.

Conclusions

- Assume that we apply to Fiedler and classical Frobenius companion **matrices** of a monic polynomial $p(z)$ the "same eigenvalue algorithm" (or algorithms with similar matrix backward stability properties) for computing its roots.
- Proved: these approaches do NOT lead to guaranteed polynomial backward stability, but from the point of view of polynomial backw-errors:
- Proved: Unbalanced Fiedler matrices are as good as classical Frobenius companion matrices if $\|p(z)\|_\infty$ is moderate.
- Proved: Unbalanced Fiedler matrices are worse than classical Frobenius companion matrices if $\|p(z)\|_\infty \gg 1$, but both are bad.
- From numerical experiments: Balanced Fiedler matrices are as good as classical Frobenius companion matrices always, but none of them are always good.
- Proved: with Fiedler and classical Frobenius companion **pencils+QZ** **perfect polynomial backward stability is guaranteed**, but it is computationally more expensive (at present) and the effect of conditioning needs to be investigated.

Conclusions

- Assume that we apply to Fiedler and classical Frobenius companion **matrices** of a monic polynomial $p(z)$ the "same eigenvalue algorithm" (or algorithms with similar matrix backward stability properties) for computing its roots.
- Proved: these approaches do NOT lead to guaranteed polynomial backward stability, but from the point of view of polynomial backw-errors:
- Proved: Unbalanced Fiedler matrices are as good as classical Frobenius companion matrices if $\|p(z)\|_\infty$ is moderate.
- Proved: Unbalanced Fiedler matrices are worse than classical Frobenius companion matrices if $\|p(z)\|_\infty \gg 1$, but both are bad.
- From numerical experiments: Balanced Fiedler matrices are as good as classical Frobenius companion matrices always, but none of them are always good.
- Proved: with Fiedler and classical Frobenius companion **pencils+QZ** **perfect polynomial backward stability is guaranteed**, but it is computationally more expensive (at present) and the effect of conditioning needs to be investigated.

Conclusions

- Assume that we apply to Fiedler and classical Frobenius companion **matrices** of a monic polynomial $p(z)$ the "same eigenvalue algorithm" (or algorithms with similar matrix backward stability properties) for computing its roots.
- Proved: these approaches do NOT lead to guaranteed polynomial backward stability, but from the point of view of polynomial backw-errors:
- Proved: Unbalanced Fiedler matrices are as good as classical Frobenius companion matrices if $\|p(z)\|_\infty$ is moderate.
- Proved: Unbalanced Fiedler matrices are worse than classical Frobenius companion matrices if $\|p(z)\|_\infty \gg 1$, but both are bad.
- From numerical experiments: Balanced Fiedler matrices are as good as classical Frobenius companion matrices always, but none of them are always good.
- Proved: with Fiedler and classical Frobenius companion **pencils+QZ** **perfect polynomial backward stability is guaranteed**, but it is computationally more expensive (at present) and the effect of conditioning needs to be investigated.