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
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.
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

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

Rigorous meaning:

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

\[ \|q(z) - \tilde{q}(z)\|_\infty = O(u) \|q(z)\|_\infty, \]

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 \( n \).
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

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

Rigorous meaning:

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

\[ \|q(z) - \tilde{q}(z)\|_\infty = O(u) \|q(z)\|_\infty, \]

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 \( \tilde{q}(z) = \tilde{b}_n z^n + \tilde{b}_{n-1} z^{n-1} + \cdots + \tilde{b}_1 z + \tilde{b}_0 \) and

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

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 \( \rightarrow \) too strict!!

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

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

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 \( \tilde{q}(z) = \tilde{b}_n z^n + \tilde{b}_{n-1} z^{n-1} + \cdots + \tilde{b}_1 z + \tilde{b}_0 \) and

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

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 → too strict!!

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

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

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 \( \tilde{q}(z) = \tilde{b}_n z^n + \tilde{b}_{n-1} z^{n-1} + \cdots + \tilde{b}_1 z + \tilde{b}_0 \) and

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

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 \( \rightarrow \) too strict!!

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

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

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

\[ \tilde{q}(z) = \tilde{b}_n z^n + \tilde{b}_{n-1} z^{n-1} + \cdots + \tilde{b}_1 z + \tilde{b}_0 \]

and

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

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 → too strict!!

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

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

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

  \[ q(z) = b_n z^n + b_{n-1} z^{n-1} + \cdots + b_1 z + b_0 \quad \rightarrow \quad p(z) := q(z)/b_n. \]

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

- **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

  \[ q(z) = b_n z^n + b_{n-1} z^{n-1} + \cdots + b_1 z + b_0 \rightarrow p(z) := q(z) / b_n . \]
  \[ p(z) = z^n + a_{n-1} z^{n-1} + \cdots + a_1 z + a_0 . \]

- **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

  \[ q(z) = b_n z^n + b_{n-1} z^{n-1} + \cdots + b_1 z + b_0 \rightarrow p(z) := q(z)/b_n. \]

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

- **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.

F. M. Dopico (U. Carlos III, Madrid)
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

  \[ q(z) = b_n z^n + b_{n-1} z^{n-1} + \cdots + b_1 z + b_0 \rightarrow p(z) := q(z)/b_n. \]

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

- **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.
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 \texttt{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

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

- \(O(n^3)\) 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 \texttt{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 \texttt{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

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

- \(O(n^3)\) 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 \texttt{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 $C + E$, with $\|E\|_2 = O(u)\|C\|_2$, where $u \approx 10^{-16}$ is the unit roundoff.
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 \( C + E \), with \( \|E\|_2 = O(u)\|C\|_2 = O(u)\|p\|_\infty \),

or **the exact roots of**

\[
\tilde{p}(z) = \det(zI - (C + E)).
\]


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

which means that **perfect matrix backward stability DOES NOT imply** perfect **polynomial backward stability** \( \implies \) **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 \( p(z) = z^n + a_{n-1}z^{n-1} + \cdots + a_1z + a_0 \),

\[
c_n \|C\|_2 \leq \|p\|_\infty \leq d_n \|C\|_2,
\]

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 \( C + E \), with \( \|E\|_2 = O(u)\|C\|_2 = O(u)\|p\|_\infty \),

or the exact roots of

\[
\tilde{p}(z) = \det(zI - (C + E)).
\]


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

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 \( p(z) = z^n + a_{n-1}z^{n-1} + \cdots + a_1z + a_0 \),

\[
c_n \|C\|_2 \leq \|p\|_\infty \leq d_n \|C\|_2,
\]

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 \( C + E \), with \( \|E\|_2 = O(u)\|C\|_2 = O(u)\|p\|_\infty \),

or **the exact roots of**

\[
\tilde{p}(z) = \det(zI - (C + E)).
\]


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

which means that **perfect matrix backward stability** **DOES NOT** imply **perfect polynomial backward stability** \( \implies \) 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 \( p(z) = z^n + a_{n-1}z^{n-1} + \cdots + a_1z + a_0 \),
  \[
  c_n \|C\|_2 \leq \|p\|_\infty \leq d_n \|C\|_2,
  \]
  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 \( C + E \), with \( \|E\|_2 = O(u)\|C\|_2 = O(u)\|p\|_\infty \),
  or the exact roots of \( \tilde{p}(z) = \det(zI - (C + E)) \).

  \[
  \tilde{p}(z) = p(z) + e(z), \quad \text{with} \quad \|e(z)\|_\infty = O(u)\|p(z)\|_\infty^2,
  \]
  which means that perfect matrix backward stability DOES NOT imply perfect polynomial backward stability \( \Rightarrow \) 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 \( C + E \), with \( \|E\|_2 = O(u)\|C\|_2 = O(u)\|p\|_\infty \),
or the exact roots of
  \[
  \tilde{p}(z) = \det(zI - (C + E)).
  \]
  \[
  \tilde{p}(z) = p(z) + e(z), \quad \text{with} \quad \|e(z)\|_\infty = O(u)\|p(z)\|_\infty^2,
  \]
  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 \( p(z) = z^n + a_{n-1}z^{n-1} + \cdots + a_1z + a_0 \),

  \[
  c_n \|C\|_2 \leq \|p\|_\infty \leq d_n \|C\|_2,
  \]

  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 \( C + E \), with \( \|E\|_2 = O(u)\|C\|_2 = O(u)\|p\|_\infty \),

  or the exact roots of

  \[
  \tilde{p}(z) = \det(zI - (C + E)).
  \]


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

  which means that perfect matrix backward stability DOES NOT imply perfect polynomial backward stability \( \implies \) 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

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

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

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

F. M. Dopico (U. Carlos III, Madrid)
Backward stability-Fiedler matrices
November 16, 2015 9 / 42
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 $\|\tilde{p}(z) - p(z)\|_\infty = \|\det(zI - (C + E)) - \det(zI - C)\|_\infty$.

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

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

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

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

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

F. M. Dopico (U. Carlos III, Madrid)
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 $M_\sigma$ is a Fiedler matrix, we consider more general perturbations of $M_\sigma$
  \[ \| E \|_2 = O(u) \alpha(p) \| M_\sigma \|_2, \]
  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 \( M_\sigma \) is a Fiedler matrix, we consider more general perturbations of \( M_\sigma \)

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

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 \( M_\sigma \) is a Fiedler matrix, we consider more general perturbations of \( M_\sigma \)

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

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 $M_\sigma$ is a Fiedler matrix, we consider more general perturbations of $M_\sigma$

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

    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 $M_\sigma$ is a Fiedler matrix, we consider more general perturbations of $M_\sigma$

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

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 $M_\sigma$ is a Fiedler matrix, we consider more general perturbations of $M_\sigma$

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

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

\[
\| \tilde{p}(z) - p(z) \|_\infty = \| \det(zI - (M_\sigma + E)) - \det(zI - M_\sigma) \|_\infty
\]

\[
= O(u) \alpha(p) \| p(z) \|_\infty^3,
\]

if \( M_\sigma \) 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

\[
\| \tilde{p}(z) - p(z) \|_\infty = \| \det(zI - (M_\sigma + E)) - \det(zI - M_\sigma) \|_\infty
\]

\[
= O(u) \alpha(p) \| p(z) \|_\infty^3,
\]

if \( M_\sigma \) 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

\[
\| E \|_2 = O(u) \alpha(p) \| M_\sigma \|_2 \quad (= O(u) \alpha(p) \| p \|_\infty),
\]

then

\[
\| \tilde{p}(z) - p(z) \|_\infty = \| \det(zI - (M_\sigma + E)) - \det(zI - M_\sigma) \|_\infty
\]

\[
= O(u) \alpha(p) \| p(z) \|_\infty^3,
\]

if \( M_\sigma \) 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

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

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

if $M_\sigma$ 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”...”

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

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

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

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

or any other Fiedler companion pencil

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

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

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

or any other Fiedler companion pencil

\[
F_\sigma(z) = z \begin{bmatrix} b_n & 1 & \cdots & 1 \\ 1 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 1 \end{bmatrix} - M_\sigma.
\]

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

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

or any other Fiedler companion pencil

\[ F_{\sigma}(z) = z \begin{bmatrix} b_n & 1 & \cdots & 1 \\ & 1 & \cdots & \ddots \\ & & \ddots & \ddots \\ & & & \ddots \end{bmatrix} - M_{\sigma}. \]

Both satisfy: \( q(z) = \det(C(z)) = \det(F_{\sigma}(z)) \).
Alternative algorithms.

- **Step 1.** Normalize the polynomial

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

- **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

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

- **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**

\[ q(z) = b_n z^n + b_{n-1} z^{n-1} + \cdots + b_1 z + b_0 \quad \rightarrow \quad s(z) := q(z)/\|q(z)\|_\infty. \]

**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

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

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 (II)

Alternative algorithms.

- **Step 1.** Normalize the polynomial

\[ q(z) = b_n z^n + b_{n-1} z^{n-1} + \cdots + b_1 z + b_0 \quad \rightarrow \quad s(z) := q(z)/\|q(z)\|_{\infty}. \]

- **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...
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

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

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

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

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

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

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

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

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

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

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

\[
\|q(z) - \tilde{q}(z)\|_\infty = O(u)\|q(z)\|_\infty,
\]

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

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

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

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

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

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

that is, **perfect polynomial backward stability!!!**
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
Theorem (Jacobi)

Let \( A, E \in \mathbb{C}^{n \times n} \). Then

\[
\tilde{\rho}(z) - p(z) := \det(zI - (A + E)) - \det(zI - A) = -\text{trace}(\text{adj}(zI - A) E) + O(\|E\|^2),
\]

where \( \text{adj}(zI - A) \) 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

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

and

\[
A_{n-1} = I, \quad A_k = A A_{k+1} + a_{k+1} I, \quad \text{for } k = n - 2, n - 3, \ldots, 0.
\]
Jacobi formula and consequences (I)

**Theorem (Jacobi)**

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

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

where $\text{adj}(zI - A)$ 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 $p(z) := \det(zI - A) = z^n + a_{n-1}z^{n-1} + \cdots + a_1z + a_0$. Then

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

and

$$
A_{n-1} = I, \quad A_k = AA_{k+1} + a_{k+1}I, \quad \text{for} \; k = n - 2, n - 3, \ldots, 0.
$$
Let $A, E \in \mathbb{C}^{n \times n}$,

\[ p(z) := \det(zI - A) = z^n + a_{n-1}z^{n-1} + \cdots + a_1z + a_0, \]
\[ \tilde{p}(z) := \det(zI - (A + E)) = z^n + \tilde{a}_{n-1}z^{n-1} + \cdots + \tilde{a}_1z + \tilde{a}_0, \]

and

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

Then

\[ \tilde{a}_k - a_k := -\text{trace}(A_k E) + O(\|E\|^2), \quad \text{for} \quad k = 0, 1, \ldots, n - 1. \]

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

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

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

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

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

and

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

Then

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

Explicit formulas for $\text{trace}(A_k E)$ 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.
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
The best known companion matrices of a monic polynomial

\[ p(z) = z^n + a_{n-1}z^{n-1} + \cdots + a_1z + a_0, \]

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

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

which have the property that

\[
\det(zI - C_1) = \det(zI - C_2) = p(z).
\]
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

$$p(z) := \det(zI - C_1) = z^n + a_{n-1}z^{n-1} + \cdots + a_1z + a_0,$$

$$\tilde{p}(z) := \det(zI - (C_1 + E)) = z^n + \tilde{a}_{n-1}z^{n-1} + \cdots + \tilde{a}_1z + \tilde{a}_0.$$

Then, to first order in $E$:

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

\[
p(z) := \det(zI - C_1) = z^n + a_{n-1}z^{n-1} + \cdots + a_1z + a_0,
\]

\[
\tilde{p}(z) := \det(zI - (C_1 + E)) = z^n + \tilde{a}_{n-1}z^{n-1} + \cdots + \tilde{a}_1z + \tilde{a}_0.
\]

Then, to first order in $E$:

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

$$p(z) := \det(zI - C_1) = z^n + a_{n-1} z^{n-1} + \cdots + a_1 z + a_0,$$

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

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

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

- 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

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

$$p(z) := \det(zI - C_1) = z^n + a_{n-1} z^{n-1} + \cdots + a_1 z + a_0,$$

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

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

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

- 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

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

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

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

\[
\|\tilde{p}(z) - p(z)\|_\infty = O(u) \alpha(p) \|p(z)\|_\infty^2.
\]

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

\[
\|\tilde{p}(z) - p(z)\|_\infty = O(u) \|p(z)\|_\infty.
\]
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

$$p(z) := \det(zI - C_1) = z^n + a_{n-1} z^{n-1} + \cdots + a_1 z + a_0,$$

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

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

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

- 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

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

\[
p(z) := \det(zI - C_1) = z^n + a_{n-1}z^{n-1} + \cdots + a_1 z + a_0,
\]

\[
\tilde{p}(z) := \det(zI - (C_1 + E)) = z^n + \tilde{a}_{n-1}z^{n-1} + \cdots + \tilde{a}_1 z + \tilde{a}_0.
\]

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

\[
\|\tilde{p}(z) - p(z)\|_\infty = O(u) \alpha(p) \|p(z)\|_\infty^2.
\]

- 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

\[
\|\tilde{p}(z) - p(z)\|_\infty = O(u) \|p(z)\|_\infty.
\]
Given \( p(z) = z^n + a_{n-1}z^{n-1} + \cdots + a_1z + a_0 \), we define the following matrices

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

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

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 \( \sigma \) 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 $p(z) = z^n + a_{n-1}z^{n-1} + \cdots + a_1z + a_0$, we define the following matrices

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

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

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 $\sigma$ 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

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

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

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 $\sigma$ 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:  \( C_1 = M_5 M_4 M_3 M_2 M_1 M_0 \)

\[
\begin{pmatrix}
-a_5 & -a_4 & -a_3 & -a_2 & -a_1 & -a_0 \\
1 & 1 & 1 & 1 & 1 & 1
\end{pmatrix}
\]

Second Frobenius companion matrix:  \( C_2 = M_0 M_1 M_2 M_3 M_4 M_5 \)

\[
\begin{pmatrix}
-a_5 & 1 \\
-a_4 & 1 \\
-a_3 & 1 \\
-a_2 & 1 \\
-a_1 & 1 \\
-a_0 & 1
\end{pmatrix}
\]
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:**
\[ C_2 = M_0 M_1 M_2 M_3 M_4 M_5 \]
\[
\begin{bmatrix}
-a_5 & 1 \\
-a_4 & 1 \\
-a_3 & 1 \\
-a_2 & 1 \\
-a_1 & 1 \\
-a_0 & 1
\end{bmatrix}
\]
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:**
\[ C_1 = M_5 M_4 M_3 M_2 M_1 M_0 \]
\[
\begin{bmatrix}
-a_5 & -a_4 & -a_3 & -a_2 & -a_1 & -a_0 \\
1 & & & & & \\
1 & & & & & \\
1 & & & & & \\
1 & & & & & \\
1 & & & & & \\
1 & & & & & \\
\end{bmatrix}
\]

**Second Frobenius companion matrix:**
\[ C_2 = M_0 M_1 M_2 M_3 M_4 M_5 \]
\[
\begin{bmatrix}
-a_5 & 1 \\
-a_4 & 1 \\
-a_3 & 1 \\
-a_2 & 1 \\
-a_1 & 1 \\
a_0 & 1 \\
\end{bmatrix}
\]
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: \( M_\sigma = M_0 M_1 M_5 M_4 M_3 M_2 \)

\[
\begin{bmatrix}
-a_5 & -a_4 & -a_3 & -a_2 & 1 \\
1 & & & & & \\
& 1 & & & & \\
& & 1 & & & \\
& & & 1 & & \\
& & & & -a_1 & \\
& & & & -a_0 & 1 \\
\end{bmatrix}
\]
Examples of Fiedler matrices

**First Frobenius companion matrix:**  \[ C_1 = M_5 M_4 M_3 M_2 M_1 M_0 \]

\[
\begin{bmatrix}
-a_5 & -a_4 & -a_3 & -a_2 & -a_1 & -a_0 \\
1 & 1 & 1 & 1 & 1 & 1
\end{bmatrix}
\]

**Another Fiedler matrix:**  \[ M_{\sigma} = M_0 M_1 M_5 M_4 M_3 M_2 \]

\[
\begin{bmatrix}
-a_5 & -a_4 & -a_3 & -a_2 & 1 \\
1 & 1 & 1 & -a_1 & 1 \\
-a_0 & -a_1 & -a_0 & \end{bmatrix}
\]

**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: \[ C_1 = M_5 M_4 M_3 M_2 M_1 M_0 \]

\[
\begin{bmatrix}
-a_5 & -a_4 & -a_3 & -a_2 & -a_1 & -a_0 \\
1 & 1 & 1 & 1 & 1 & 1 \\
\end{bmatrix}
\]

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

\[
\begin{bmatrix}
-a_5 & -a_4 & -a_3 & -a_2 & 1 \\
1 & 1 & 1 & 1 & -a_1 \\
-a_0 & 1 \\
\end{bmatrix}
\]

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: \textbf{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

\textbf{Frobenius companion matrices} are the Fiedler matrices with largest bandwidth and \textbf{pentadiagonal Fiedler matrices} are the ones with smallest bandwidth.
Recall that the Fiedler matrix $M_\sigma$ associated with a permutation $\sigma$ 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 $p(\zeta)$ of degree $n$. 
Recall that the Fiedler matrix $M_{\sigma}$ associated with a permutation $\sigma$ 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

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
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

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

Then, to first order in $E$:

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

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

$$p(z) := \text{det}(zI - M_{\sigma}) = z^n + a_{n-1}z^{n-1} + \cdots + a_1z + a_0,$$

$$\tilde{p}(z) := \text{det}(zI - (M_{\sigma} + E)) = z^n + \tilde{a}_{n-1}z^{n-1} + \cdots + \tilde{a}_1z + \tilde{a}_0.$$

Then, to first order in $E$:

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

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...
...the horror!!

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

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

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

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

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

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

$$l = \min\{k+1-c_\sigma(n-j:n-i-1),i-1\}$$

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

if $j > i$ and $k + 2 + j - i - n \leq c_\sigma(n-j:n-i-1) \leq k + 1$;

$$l = \min\{k+1-i_\sigma(n-i:n-j-1),j-1\}$$

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

if $j < i$ and $k + 2 + i - j - n \leq i_\sigma(n-i:n-j-1) \leq k + 1$;

0, otherwise.
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

$$p(z) := \det(zI - M_\sigma) = z^n + a_{n-1}z^{n-1} + \cdots + a_1z + a_0,$$

$$\tilde{p}(z) := \det(zI - (M_\sigma + E)) = z^n + \tilde{a}_{n-1}z^{n-1} + \cdots + \tilde{a}_1z + \tilde{a}_0.$$

Then, to first order in $E$:

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

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.
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

$$p(z) := \det(zI - M_\sigma) = z^n + a_{n-1}z^{n-1} + \cdots + a_1z + a_0,$$

$$\tilde{p}(z) := \det(zI - (M_\sigma + E)) = z^n + \tilde{a}_{n-1}z^{n-1} + \cdots + \tilde{a}_1z + \tilde{a}_0.$$

Then, to first order in $E$:

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

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

$$p(z) := \det(zI - M_\sigma) = z^n + a_{n-1} z^{n-1} + \cdots + a_1 z + a_0,$$

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

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

- For $M_\sigma$ Frobenius companion matrix,

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

- For $M_\sigma$ NOT Frobenius companion matrix,

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

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

$$p(z) := \det(zI - M_\sigma) = z^n + a_{n-1}z^{n-1} + \cdots + a_1z + a_0,$$
$$\tilde{p}(z) := \det(zI - (M_\sigma + E)) = z^n + \tilde{a}_{n-1}z^{n-1} + \cdots + \tilde{a}_1z + \tilde{a}_0.$$

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

- For $M_\sigma$ Frobenius companion matrix,
  $$\|\tilde{p}(z) - p(z)\|_\infty = O(u \alpha(p) \|p(z)\|_\infty^2).$$

- For $M_\sigma$ NOT Frobenius companion matrix,
  $$\|\tilde{p}(z) - p(z)\|_\infty = O(u \alpha(p) \|p(z)\|_\infty^3).$$

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

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

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

- For $M_\sigma$ Frobenius companion matrix,
  \[\|\tilde{p}(z) - p(z)\|_\infty = O(u)\alpha(p)\|p(z)\|_\infty^2.\]

- For $M_\sigma$ NOT Frobenius companion matrix,
  \[\|\tilde{p}(z) - p(z)\|_\infty = O(u)\alpha(p)\|p(z)\|_\infty^3.\]

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 immediate to choose $\beta$ such that $\max_i |a_i \beta^{n-i}| = 1$.

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,

$$\|\tilde{t}(z) - t(z)\|_\infty = O(u) \Rightarrow \|\tilde{p}(z) - p(z)\|_\infty = O(u) \max_i |\beta|^{i-n} = O(u) \left(\frac{1}{|\beta|}\right)^n.$$
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 immediate to choose \( \beta \) such that \( \max_i |a_i \beta^{n-i}| = 1 \).

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,

\[
\|\tilde{t}(z) - t(z)\|_\infty = O(u) \Rightarrow \|\tilde{p}(z) - p(z)\|_\infty = O(u) \max_i |\beta|^{i-n} = O(u) \left( \frac{1}{|\beta|} \right)^n.
\]
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 immediate to choose \( \beta \) such that \( \max_i |a_i \beta^{n-i}| = 1 \).

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,

\[
\|\tilde{t}(z) - t(z)\|_\infty = O(u) \Rightarrow \|\tilde{p}(z) - p(z)\|_\infty = O(u) \max_i |\beta|^{i-n} = O(u) \left( \frac{1}{|\beta|} \right)^n.
\]
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 immediate to choose \( \beta \) such that \( \max_i |a_i \beta^{n-i}| = 1 \).

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,

\[
    \| \tilde{t}(z) - t(z) \|_\infty = O(u) \Rightarrow \| \tilde{p}(z) - p(z) \|_\infty = O(u) \max_i |\beta|^{i-n} = O(u) \left( \frac{1}{|\beta|} \right)^n.
\]
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 immediate to choose \( \beta \) such that \( \max_i |a_i \beta^{n-i}| = 1 \).

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,

\[
\|\tilde{t}(z) - t(z)\|_\infty = O(u) \Rightarrow \|\tilde{p}(z) - p(z)\|_\infty = O(u) \max_i |\beta|^{i-n} = O(u) \left(\frac{1}{|\beta|}\right)^n.
\]
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**:
  \[
  \|\tilde{p}(z) - p(z)\|_{\infty} = O(u) \|p(z)\|_{\infty}.
  \]

- 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:

\[
\| \tilde{p}(z) - p(z) \|_\infty = O(u) \| p(z) \|_\infty.
\]

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:

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

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:

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

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 $M_\sigma$ of $p(z)$ consists in

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

  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

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

- or, the exact roots of

  $$\tilde{p}(z) = \det(zI - (DM_\sigma D^{-1} + \tilde{E}))$$

  $$= \det(zI - (M_\sigma + D^{-1}\tilde{E}D))$$

- We have already solved this problem!!
Balancing a Fiedler matrix $M_\sigma$ of $p(z)$ consists in

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

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

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

or, the exact roots of

$$\tilde{p}(z) = \det(zI - (DM_\sigma D^{-1} + \tilde{E}))$$

$$= \det(zI - (M_\sigma + D^{-1}\tilde{E}D))$$

We have already solved this problem!!
How to deal with balancing?

- Balancing a Fiedler matrix $M_\sigma$ of $p(z)$ consists in

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

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

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

- or, the exact roots of

$$\tilde{p}(z) = \det(zI - (DM_\sigma D^{-1} + \tilde{E}))$$

$$= \det(zI - (M_\sigma + D^{-1}\tilde{E}D))$$

- We have already solved this problem!!
Balancing a Fiedler matrix $M_\sigma$ of $p(z)$ consists in

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

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

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

or, the exact roots of

$$\tilde{p}(z) = \det(z I - (D M_\sigma D^{-1} + \tilde{E}))$$

$$= \det(z I - (M_\sigma + D^{-1} \tilde{E} D))$$

We have already solved this problem!!
How to deal with balancing?

- Balancing a Fiedler matrix $M_\sigma$ of $p(z)$ consists in
  
  $$M_\sigma \rightarrow D M_\sigma D^{-1}, \quad \text{with } D = \text{diag}(2^{t_1}, \ldots, 2^{t_n})$$

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

- or, the exact roots of
  
  $$\tilde{p}(z) = \det(z I - (D M_\sigma D^{-1} + \tilde{E})) = \det(z I - (M_\sigma + D^{-1} \tilde{E} D))$$

- We have already solved this problem!!
The effect of balancing on polynomial backward error

**Theorem**

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

\[
p(z) := \det(zI - M_\sigma) = z^n + a_{n-1} z^{n-1} + \cdots + a_1 z + a_0,
\]

\[
\tilde{p}(z) := \det(zI - (M_\sigma + D^{-1}\tilde{E}D)) = z^n + \tilde{a}_{n-1} z^{n-1} + \cdots + \tilde{a}_1 z + \tilde{a}_0.
\]

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

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

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

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

\[
\|\tilde{p}(z) - p(z)\|_\infty = O(u) \max_{i,j,k} \left( \left| p_{ij}^{(\sigma,k)}(a_0, \ldots, a_{n-1}) \frac{d_j}{d_i} \right| \right) \|DM_\sigma D^{-1}\|_2.
\]
The effect of balancing on polynomial backward error

**Theorem**

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

$$p(z) := \det(zI - M_\sigma) = z^n + a_{n-1} z^{n-1} + \cdots + a_1 z + a_0,$$

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

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

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

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

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

$$\|\widetilde{p}(z) - p(z)\|_\infty = O(u) \max_{i,j,k} \left( \left| p_{ij}^{(\sigma,k)}(a_0, \ldots, a_{n-1}) \frac{d_j}{d_i} \right| \right) \|D M_\sigma D^{-1}\|_2.$$
...but we cannot go further

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

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

\[ \max_{i,j,k} \left( \left\| \frac{p_{i,j}^{(\sigma,k)}(a_0, \ldots, a_{n-1})}{d_i} \right\|_\infty \right) \]

- nor

\[ \| DM_\sigma D^{-1} \|_2 \]

a priori,

- while without balancing

\[ \max_{i,j,k} \left( \left\| \frac{p_{i,j}^{(\sigma,k)}(a_0, \ldots, a_{n-1})}{d_i} \right\|_\infty \right) \leq n \| p(z) \|_\infty^2, \quad \| M_\sigma \|_2 \approx \| p(z) \|_\infty. \]
...but we cannot go further

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

- 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

\[ \max_{i,j,k} \left( \left\| p_{ij}^{(\sigma, k)} (a_0, \ldots, a_{n-1}) \right\| \right) \leq n \| p(z) \|_\infty^2, \quad \| M_\sigma \|_2 \approx \| p(z) \|_\infty. \]
...but we cannot go further

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

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

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

- nor

\[ \| DM_\sigma D^{-1} \|_2 \]

- a priori,
- while without balancing

\[ \max_{i,j,k} \left( \left\| p_{i,j}^{(\sigma,k)} (a_0, \ldots, a_{n-1}) \right\| \right) \leq n \| p(z) \|_\infty^2, \quad \| M_\sigma \|_2 \approx \| p(z) \|_\infty. \]
...but we cannot go further

\[ \| \tilde{p}(z) - p(z) \|_\infty = O(u) \max_{i,j,k} \left( \left| \frac{\partial^{(\sigma,k)} p_{ij}(a_0, \ldots, a_{n-1})}{\partial a_i} \right| \right) \| DM_\sigma D^{-1} \|_2 \]

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

\[ \max_{i,j,k} \left( \left| \frac{\partial^{(\sigma,k)} p_{ij}(a_0, \ldots, a_{n-1})}{\partial a_i} \right| \right) \]

- nor

\[ \| DM_\sigma D^{-1} \|_2 \]

a priori,

- while without balancing

\[ \max_{i,j,k} \left( \left| \frac{\partial^{(\sigma,k)} p_{ij}(a_0, \ldots, a_{n-1})}{\partial a_i} \right| \right) \leq n \| p(z) \|^{2}_\infty, \quad \| M_\sigma \|_2 \approx \| p(z) \|_\infty. \]
...but we cannot go further

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

- 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

\[ \max_{i,j,k} \left( \left\| p_{ij}^{(\sigma,k)}(a_0, \ldots, a_{n-1}) \right\| \right) \leq n \| p(z) \|_\infty^2, \quad \| M_\sigma \|_2 \approx \| p(z) \|_\infty. \]
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
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
     - $M_{\sigma_1} =$ second classical Frobenius,
     - $M_{\sigma_2} =$ a pentadiagonal,
     - $M_{\sigma_3} =$ the second F-matrix,
     - $M_{\sigma_4} =$ “another one”.

F. M. Dopico (U. Carlos III, Madrid)
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
   - $M_{\sigma_1}$ = second classical Frobenius,
   - $M_{\sigma_2}$ = a pentadiagonal,
   - $M_{\sigma_3}$ = the second F-matrix,
   - $M_{\sigma_4}$ = “another one”.

F. M. Dopico (U. Carlos III, Madrid)
Backward stability-Fiedler matrices
November 16, 2015
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 $M_{\sigma_1} =$ second classical Frobenius, $M_{\sigma_2} =$ a pentadiagonal, $M_{\sigma_3} =$ the second F-matrix, $M_{\sigma_4} =$ “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
   - $M_{\sigma_1} =$ second classical Frobenius,
   - $M_{\sigma_2} =$ a pentadiagonal,
   - $M_{\sigma_3} =$ the second F-matrix,
   - $M_{\sigma_4} =$ “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
   - $M_{\sigma_1} =$ second classical Frobenius,
   - $M_{\sigma_2} =$ a pentadiagonal,
   - $M_{\sigma_3} =$ the second F-matrix,
   - $M_{\sigma_4} =$ “another one”.

Backward stability-Fiedler matrices  
F. M. Dopico (U. Carlos III, Madrid)
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
   - \( M_{\sigma_1} \) = second classical Frobenius,
   - \( M_{\sigma_2} \) = a pentadiagonal,
   - \( M_{\sigma_3} \) = the second F-matrix,
   - \( M_{\sigma_4} \) = “another one”.

F. M. Dopico (U. Carlos III, Madrid)
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
   - \( M_{\sigma_1} \) = second classical Frobenius,
   - \( M_{\sigma_2} \) = a pentadiagonal,
   - \( M_{\sigma_3} \) = the second F-matrix,
   - \( M_{\sigma_4} \) = “another one”.

F. M. Dopico (U. Carlos III, Madrid)
Numerical experiments (without balancing)

(a) $M_{\sigma_1}$

(b) $M_{\sigma_2}$

(c) $M_{\sigma_3}$

(d) $M_{\sigma_4}$
Numerical experiments (with balancing): surprise!!
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
Let $A, B, E, G \in \mathbb{C}^{n \times n}$, and

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

$$\tilde{q}(z) := \det(z(B + G) - (A + E)) = \tilde{b}_n z^n + \tilde{b}_{n-1} z^{n-1} + \cdots + \tilde{b}_1 z + \tilde{b}_0.$$

Then

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

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

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

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

Theorem (Corollary of Theorem Jacobi)

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

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

Then

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

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

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

with \( P_n = P_{-1} := 0 \).
Theorem (Corollary of Theorem Jacobi)

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

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

$$\tilde{q}(z) := \det(z(B + G) - (A + E)) = \tilde{b}_n z^n + \tilde{b}_{n-1} z^{n-1} + \cdots + \tilde{b}_1 z + \tilde{b}_0.$$

Then

1. $$\tilde{q}(z) - q(z) = \text{trace}(\text{adj}(zB - A)(zG - E)) + O(\|EG\|^2),$$

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

$$\tilde{b}_k - b_k := \text{trace}(P_{k-1} G - P_k E) + O(\|EG\|^2), \quad \text{for} \quad k = 0, 1, \ldots, n,$$

with $P_n = P_{-1} := 0.$
The case of Fiedler pencils

- So, the key to bound $|\tilde{b}_k - b_k|$ 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 Fiedler matrices $M_\sigma$ as follows.

- Define from $q(z)$ the monic polynomial $p(z) := q(z)/b_n$. Then, it can be proved

$$F_\sigma(q) = S_\sigma(zI - M_\sigma(p))T_\sigma,$$

with

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

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

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

- From here,

$$\text{adj} F_\sigma(q) = \text{adj}(T_\sigma) \text{adj}(zI - M_\sigma(p)) \text{adj}(S_\sigma).$$
### The case of Fiedler pencils

- So, the key to bound $|\tilde{b}_k - b_k|$ 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 $M_\sigma$ as follows.

- Define from $q(z)$ the monic polynomial $p(z) := q(z)/b_n$. Then, it can be proved

\[
F_\sigma(q) = S_\sigma(zI - M_\sigma(p))T_\sigma,
\]

with

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

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

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

- From here,

\[
\text{adj} F_\sigma(q) = \text{adj}(T_\sigma) \text{adj}(zI - M_\sigma(p)) \text{adj}(S_\sigma).
\]
So, the key to bound $|b_k - \tilde{b}_k|$ 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 $M_\sigma$ as follows.

Define from $q(z)$ the monic polynomial $p(z) := q(z)/b_n$. Then, it can be proved

$$F_\sigma(q) = S_\sigma(zI - M_\sigma(p))T_\sigma,$$

with

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

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

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

From here,

$$\text{adj} F_\sigma(q) = \text{adj}(T_\sigma) \text{adj}(zI - M_\sigma(p)) \text{adj}(S_\sigma).$$
So, the key to bound $|\tilde{b}_k - b_k|$ 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 $M_{\sigma}$ as follows.

Define from $q(z)$ the monic polynomial $p(z) := q(z)/b_n$. Then, it can be proved

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

with

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

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

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

From here,

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

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

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

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

- for $F_\sigma(z)$ NOT Frobenius companion pencil,
  $$\|\tilde{q}(z) - q(z)\|_\infty = O(u) \max\{1, \|q(z)\|_\infty^3\}.$$

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

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

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

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,

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

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

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

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

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

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,

$$\|\tilde{q}(z) - q(z)\|_{\infty} = O(u) \max\{1, \|q(z)\|_2\}.$$ 

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

$$\|\tilde{q}(z) - q(z)\|_{\infty} = O(u) \max\{1, \|q(z)\|_3\}.$$ 

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

F. M. Dopico (U. Carlos III, Madrid)
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.