Backward Stability of Polynomial and Rational Eigenvalue Problems Solved via Linearizations

Froilán M. Dopico

Departamento de Matemáticas
Universidad Carlos III de Madrid, Spain

2018 CMMSE
18th International Conference Computational and Mathematical Methods in Science and Engineering
Rota, Cádiz - Spain. July 9-14, 2018

Collaborators

- University of California at Santa Barbara (USA): María I. Bueno.
- Universidad Carlos III de Madrid (Spain): Fernando De Terán, Javier González-Pizarro, María C. Quintana.
- Université Catholique de Louvain (Belgium): Piers Lawrence, Paul Van Dooren.
- University of Montana (USA): Javier Pérez.
- Universidad del País Vasco/Euskal Herriko Unibertsitatea: Agurtzane Amparan, Silvia Marcaida, Ion Zaballa.
- Western Michigan University (USA): Steve Mackey.

The basic eigenvalue problem (BEP). Given $A \in \mathbb{C}^{n\times n}$, compute scalars λ (eigenvalues) and nonzero vectors $v \in \mathbb{C}^n$ (eigenvectors) such that

$$Av = \lambda v \iff \boxed{(\lambda I_n - A)\, v = 0}$$

- It arises in many applications. For instance, if one looks for solutions of the form $y(t) = e^{\lambda t} v$ in the system of first order ODEs

$$\frac{dy(t)}{dt} = Ay(t) \Longrightarrow \lambda v = Av$$

- There are stable algorithms for its numerical solution.
- QR algorithm (Francis-Kublanovskaya 1961) for small to medium size dense matrices.
- Arnoldi method (1951) and (many) other variants of Krylov methods for large-scale problems and sparse matrices.
- Easy to use software. For instance MATLAB's commands `eig(A)` or `eigs(A)`.

Different classes of regular matrix eigenvalue problems (I)

The basic eigenvalue problem (BEP). Given $A \in \mathbb{C}^{n\times n}$, compute scalars λ (eigenvalues) and nonzero vectors $v \in \mathbb{C}^n$ (eigenvectors) such that

$$Av = \lambda v \Longleftrightarrow \boxed{(\lambda I_n - A)\, v = 0}$$

- It arises in many applications. For instance, if one looks for solutions of the form $y(t) = e^{\lambda t} v$ in the system of first order ODEs

$$\frac{dy(t)}{dt} = Ay(t) \Longrightarrow \lambda v = Av$$

- There are stable algorithms for its numerical solution.
- QR algorithm (Francis-Kublanovskaya 1961) for small to medium size dense matrices.
- Arnoldi method (1951) and (many) other variants of Krylov methods for large-scale problems and sparse matrices.
- Easy to use software. For instance MATLAB's commands `eig(A)` or `eigs(A)`.

The basic eigenvalue problem (BEP). Given $A \in \mathbb{C}^{n\times n}$, compute scalars λ (eigenvalues) and nonzero vectors $v \in \mathbb{C}^n$ (eigenvectors) such that

$$Av = \lambda v \Longleftrightarrow \boxed{(\lambda I_n - A)\, v = 0}$$

- It arises in many applications. For instance, if one looks for solutions of the form $y(t) = e^{\lambda t} v$ in the system of first order ODEs

$$\frac{dy(t)}{dt} = Ay(t) \Longrightarrow \lambda v = Av$$

- There are stable algorithms for its numerical solution.
- QR algorithm (Francis-Kublanovskaya 1961) for small to medium size dense matrices.
- Arnoldi method (1951) and (many) other variants of Krylov methods for large-scale problems and sparse matrices.
- Easy to use software. For instance MATLAB's commands `eig(A)` or `eigs(A)`.

The basic eigenvalue problem (BEP). Given $A \in \mathbb{C}^{n\times n}$, compute scalars λ (eigenvalues) and nonzero vectors $v \in \mathbb{C}^n$ (eigenvectors) such that

$$Av = \lambda v \iff \boxed{(\lambda I_n - A)\, v = 0}$$

- It arises in many applications. For instance, if one looks for solutions of the form $y(t) = e^{\lambda t} v$ in the system of first order ODEs

$$\frac{dy(t)}{dt} = Ay(t) \Longrightarrow \lambda v = Av$$

- There are stable algorithms for its numerical solution.
- QR algorithm (Francis-Kublanovskaya 1961) for small to medium size dense matrices.
- Arnoldi method (1951) and (many) other variants of Krylov methods for large-scale problems and sparse matrices.
- Easy to use software. For instance MATLAB's commands `eig(A)` or `eigs(A)`.

The basic eigenvalue problem (BEP). Given $A \in \mathbb{C}^{n\times n}$, compute scalars λ (eigenvalues) and nonzero vectors $v \in \mathbb{C}^n$ (eigenvectors) such that

$$Av = \lambda v \iff \boxed{(\lambda I_n - A)\, v = 0}$$

- It arises in many applications. For instance, if one looks for solutions of the form $y(t) = e^{\lambda t} v$ in the system of first order ODEs

$$\frac{dy(t)}{dt} = Ay(t) \Longrightarrow \lambda v = Av$$

- There are stable algorithms for its numerical solution.
- QR algorithm (Francis-Kublanovskaya 1961) for small to medium size dense matrices.
- Arnoldi method (1951) and (many) other variants of Krylov methods for large-scale problems and sparse matrices.
- Easy to use software. For instance MATLAB's commands `eig(A)` or `eigs(A)`.

The basic eigenvalue problem (BEP). Given $A \in \mathbb{C}^{n\times n}$, compute scalars λ (eigenvalues) and nonzero vectors $v \in \mathbb{C}^n$ (eigenvectors) such that

$$Av = \lambda v \Longleftrightarrow \boxed{(\lambda I_n - A)\, v = 0}$$

- It arises in many applications. For instance, if one looks for solutions of the form $y(t) = e^{\lambda t} v$ in the system of first order ODEs

$$\frac{dy(t)}{dt} = Ay(t) \Longrightarrow \lambda v = Av$$

- There are stable algorithms for its numerical solution.
- QR algorithm (Francis-Kublanovskaya 1961) for small to medium size dense matrices.
- Arnoldi method (1951) and (many) other variants of Krylov methods for large-scale problems and sparse matrices.
- Easy to use software. For instance MATLAB's commands `eig(A)` or `eigs(A)`.

The GENERALIZED eigenvalue problem (GEP). Given $A, B \in \mathbb{C}^{n\times n}$, compute scalars λ (eigenvalues) and nonzero vectors $v \in \mathbb{C}^n$ (eigenvectors) such that

$$Av = \lambda Bv \Longleftrightarrow \boxed{(\lambda B - A)\, v = 0},$$

under the regularity assumption $\det(zB - A)$ is not zero for all $z \in \mathbb{C}$.

- It arises in many applications. For instance, if one looks for solutions $y(t) = e^{\lambda t} v$ in the system of first order ALGEBRAIC-ODEs

$$B\frac{dy(t)}{dt} = Ay(t) \Longrightarrow \lambda Bv = Av$$

- There are stable algorithms for its numerical solution.
- QZ algorithm (Moler-Stewart 1973) for small to medium size dense matrices.
- Different (rational) Krylov methods (Ruhe, 1984-1998) for large-scale problems and sparse matrices.
- Easy to use software. For instance MATLAB's commands `eig(A,B)` or `eigs(A,B)`.

The GENERALIZED eigenvalue problem (GEP). Given $A, B \in \mathbb{C}^{n\times n}$, compute scalars λ (eigenvalues) and nonzero vectors $v \in \mathbb{C}^n$ (eigenvectors) such that

$$Av = \lambda Bv \iff \boxed{(\lambda B - A)\, v = 0},$$

under the regularity assumption $\det(zB - A)$ is not zero for all $z \in \mathbb{C}$.

- It arises in many applications. For instance, if one looks for solutions $y(t) = e^{\lambda t} v$ in the system of first order ALGEBRAIC-ODEs

$$B\frac{dy(t)}{dt} = Ay(t) \Longrightarrow \lambda Bv = Av$$

- There are stable algorithms for its numerical solution.
- QZ algorithm (Moler-Stewart 1973) for small to medium size dense matrices.
- Different (rational) Krylov methods (Ruhe, 1984-1998) for large-scale problems and sparse matrices.
- Easy to use software. For instance MATLAB's commands `eig(A,B)` or `eigs(A,B)`.

The GENERALIZED eigenvalue problem (GEP). Given $A, B \in \mathbb{C}^{n\times n}$, compute scalars λ (eigenvalues) and nonzero vectors $v \in \mathbb{C}^n$ (eigenvectors) such that

$$Av = \lambda Bv \Longleftrightarrow \boxed{(\lambda B - A)\, v = 0},$$

under the regularity assumption $\det(zB - A)$ is not zero for all $z \in \mathbb{C}$.

- It arises in many applications. For instance, if one looks for solutions $y(t) = e^{\lambda t} v$ in the system of first order ALGEBRAIC-ODEs

$$B\frac{dy(t)}{dt} = Ay(t) \Longrightarrow \lambda Bv = Av$$

- There are stable algorithms for its numerical solution.
- QZ algorithm (Moler-Stewart 1973) for small to medium size dense matrices.
- Different (rational) Krylov methods (Ruhe, 1984-1998) for large-scale problems and sparse matrices.
- Easy to use software. For instance MATLAB's commands `eig(A,B)` or `eigs(A,B)`.

The GENERALIZED eigenvalue problem (GEP). Given $A, B \in \mathbb{C}^{n\times n}$, compute scalars λ (eigenvalues) and nonzero vectors $v \in \mathbb{C}^n$ (eigenvectors) such that

$$Av = \lambda Bv \iff \boxed{(\lambda B - A)\, v = 0},$$

under the regularity assumption $\det(zB - A)$ is not zero for all $z \in \mathbb{C}$.

- It arises in many applications. For instance, if one looks for solutions $y(t) = e^{\lambda t} v$ in the system of first order ALGEBRAIC-ODEs

$$B\frac{dy(t)}{dt} = Ay(t) \Longrightarrow \lambda Bv = Av$$

- There are stable algorithms for its numerical solution.
- QZ algorithm (Moler-Stewart 1973) for small to medium size dense matrices.
- Different (rational) Krylov methods (Ruhe, 1984-1998) for large-scale problems and sparse matrices.
- Easy to use software. For instance MATLAB's commands `eig(A,B)` or `eigs(A,B)`.

The GENERALIZED eigenvalue problem (GEP). Given $A, B \in \mathbb{C}^{n\times n}$, compute scalars λ (eigenvalues) and nonzero vectors $v \in \mathbb{C}^n$ (eigenvectors) such that

$$Av = \lambda Bv \Longleftrightarrow \boxed{(\lambda B - A)\, v = 0},$$

under the regularity assumption $\det(zB - A)$ is not zero for all $z \in \mathbb{C}$.

- It arises in many applications. For instance, if one looks for solutions $y(t) = e^{\lambda t} v$ in the system of first order ALGEBRAIC-ODEs

$$B\frac{dy(t)}{dt} = Ay(t) \Longrightarrow \lambda Bv = Av$$

- There are stable algorithms for its numerical solution.
- QZ algorithm (Moler-Stewart 1973) for small to medium size dense matrices.
- Different (rational) Krylov methods (Ruhe, 1984-1998) for large-scale problems and sparse matrices.
- Easy to use software. For instance MATLAB's commands `eig(A,B)` or `eigs(A,B)`.

The GENERALIZED eigenvalue problem (GEP). Given $A, B \in \mathbb{C}^{n\times n}$, compute scalars λ (eigenvalues) and nonzero vectors $v \in \mathbb{C}^n$ (eigenvectors) such that

$$Av = \lambda Bv \iff \boxed{(\lambda B - A)\, v = 0},$$

under the regularity assumption $\det(zB - A)$ is not zero for all $z \in \mathbb{C}$.

- It arises in many applications. For instance, if one looks for solutions $y(t) = e^{\lambda t} v$ in the system of first order ALGEBRAIC-ODEs

$$B\frac{dy(t)}{dt} = Ay(t) \Longrightarrow \lambda Bv = Av$$

- There are stable algorithms for its numerical solution.
- QZ algorithm (Moler-Stewart 1973) for small to medium size dense matrices.
- Different (rational) Krylov methods (Ruhe, 1984-1998) for large-scale problems and sparse matrices.
- Easy to use software. For instance MATLAB's commands `eig(A,B)` or `eigs(A,B)`.

The POLYNOMIAL eigenvalue problem (PEP). Given $P_0, \ldots, P_d \in \mathbb{C}^{n\times n}$, compute scalars λ (eigenvalues) and nonzero vectors $v \in \mathbb{C}^n$ (eigenvectors) such that

$$\boxed{(P_d\lambda^d + \cdots + P_1\lambda + P_0)v = 0},$$

under the regularity assumption $\det(P_d z^d + \cdots + P_1 z + P_0) \not\equiv 0$.

- It arises in many applications. For instance, if one looks for solutions $y(t) = e^{\lambda t}v$ in the system of dth order ALGEBRAIC-ODEs

$$P_d \frac{d^d y(t)}{dt^d} + \cdots + P_1 \frac{dy(t)}{dt} + P_0 y(t) = 0 \Longrightarrow (P_d\lambda^d + \cdots + P_1\lambda + P_0)v = 0$$

- There are stable (? debatable) algorithms for its numerical solution.
- Easy to use software for small to medium size dense matrices: MATLAB's commands `polyeig(P0,P1,...,Pd)` (Van Dooren, 1979).
- Different **specific-structured Krylov methods** for large-scale problems and sparse matrix coefficients (Su-Bai-Lu, 2008, 2016), (Van Beeumen-Meerbergen-Michiels, 2015).
- HPC implementations of these methods in **SLEPc** (Roman, UPV).

The POLYNOMIAL eigenvalue problem (PEP). Given $P_0, \dots, P_d \in \mathbb{C}^{n \times n}$, compute scalars λ (eigenvalues) and nonzero vectors $v \in \mathbb{C}^n$ (eigenvectors) such that

$$\boxed{(P_d \lambda^d + \cdots + P_1 \lambda + P_0) v = 0},$$

under the regularity assumption $\det(P_d z^d + \cdots + P_1 z + P_0) \not\equiv 0$.

- It arises in many applications. For instance, if one looks for solutions $y(t) = e^{\lambda t} v$ in the system of dth order ALGEBRAIC-ODEs

$$P_d \frac{d^d y(t)}{dt^d} + \cdots + P_1 \frac{dy(t)}{dt} + P_0 y(t) = 0 \Longrightarrow (P_d \lambda^d + \cdots + P_1 \lambda + P_0) v = 0$$

- There are stable (? debatable) algorithms for its numerical solution.
- Easy to use software for small to medium size dense matrices: MATLAB's commands `polyeig(P0,P1,...,Pd)` (Van Dooren, 1979).
- Different **specific-structured Krylov methods** for large-scale problems and sparse matrix coefficients (Su-Bai-Lu, 2008, 2016), (Van Beeumen-Meerbergen-Michiels, 2015).
- HPC implementations of these methods in **SLEPc** (Roman, UPV).

The POLYNOMIAL eigenvalue problem (PEP). Given $P_0, \ldots, P_d \in \mathbb{C}^{n\times n}$, compute scalars λ (eigenvalues) and nonzero vectors $v \in \mathbb{C}^n$ (eigenvectors) such that

$$\boxed{(P_d\lambda^d + \cdots + P_1\lambda + P_0)v = 0},$$

under the regularity assumption $\det(P_d z^d + \cdots + P_1 z + P_0) \not\equiv 0$.

- It arises in many applications. For instance, if one looks for solutions $y(t) = e^{\lambda t}v$ in the system of dth order ALGEBRAIC-ODEs

$$P_d \frac{d^d y(t)}{dt^d} + \cdots + P_1 \frac{dy(t)}{dt} + P_0 y(t) = 0 \Longrightarrow (P_d\lambda^d + \cdots + P_1\lambda + P_0)v = 0$$

- There are stable (? debatable) algorithms for its numerical solution.
- Easy to use software for small to medium size dense matrices: MATLAB's commands `polyeig(P0,P1,...,Pd)` (Van Dooren, 1979).
- Different **specific-structured Krylov methods** for large-scale problems and sparse matrix coefficients (Su-Bai-Lu, 2008, 2016), (Van Beeumen-Meerbergen-Michiels, 2015).
- HPC implementations of these methods in **SLEPc** (Roman, UPV).

The POLYNOMIAL eigenvalue problem (PEP). Given $P_0, \ldots, P_d \in \mathbb{C}^{n\times n}$, compute scalars λ (eigenvalues) and nonzero vectors $v \in \mathbb{C}^n$ (eigenvectors) such that

$$\boxed{(P_d\lambda^d + \cdots + P_1\lambda + P_0)v = 0},$$

under the regularity assumption $\det(P_d z^d + \cdots + P_1 z + P_0) \not\equiv 0$.

- It arises in many applications. For instance, if one looks for solutions $y(t) = e^{\lambda t}v$ in the system of dth order ALGEBRAIC-ODEs

$$P_d \frac{d^d y(t)}{dt^d} + \cdots + P_1 \frac{dy(t)}{dt} + P_0 y(t) = 0 \Longrightarrow (P_d\lambda^d + \cdots + P_1\lambda + P_0)v = 0$$

- There are stable (? debatable) algorithms for its numerical solution.
- Easy to use software for small to medium size dense matrices: MATLAB's commands `polyeig(P0,P1,...,Pd)` (Van Dooren, 1979).
- Different **specific-structured Krylov methods** for large-scale problems and sparse matrix coefficients (Su-Bai-Lu, 2008, 2016), (Van Beeumen-Meerbergen-Michiels, 2015).
- HPC implementations of these methods in **SLEPc** (Roman, UPV).

The POLYNOMIAL eigenvalue problem (PEP). Given $P_0, \ldots, P_d \in \mathbb{C}^{n\times n}$, compute scalars λ (eigenvalues) and nonzero vectors $v \in \mathbb{C}^n$ (eigenvectors) such that

$$(P_d\lambda^d + \cdots + P_1\lambda + P_0)v = 0,$$

under the regularity assumption $\det(P_d z^d + \cdots + P_1 z + P_0) \not\equiv 0$.

- It arises in many applications. For instance, if one looks for solutions $y(t) = e^{\lambda t}v$ in the system of dth order ALGEBRAIC-ODEs

$$P_d \frac{d^d y(t)}{dt^d} + \cdots + P_1 \frac{dy(t)}{dt} + P_0 y(t) = 0 \Longrightarrow (P_d\lambda^d + \cdots + P_1\lambda + P_0)v = 0$$

- There are stable (? debatable) algorithms for its numerical solution.
- Easy to use software for small to medium size dense matrices: MATLAB's commands `polyeig(P0,P1,...,Pd)` (Van Dooren, 1979).
- Different **specific-structured Krylov methods** for large-scale problems and sparse matrix coefficients (Su-Bai-Lu, 2008, 2016), (Van Beeumen-Meerbergen-Michiels, 2015).
- HPC implementations of these methods in **SLEPc** (Roman, UPV).

The POLYNOMIAL eigenvalue problem (PEP). Given $P_0, \dots, P_d \in \mathbb{C}^{n\times n}$, compute scalars λ (eigenvalues) and nonzero vectors $v \in \mathbb{C}^n$ (eigenvectors) such that

$$\boxed{(P_d\lambda^d + \cdots + P_1\lambda + P_0)v = 0},$$

under the regularity assumption $\det(P_d z^d + \cdots + P_1 z + P_0) \not\equiv 0$.

- It arises in many applications. For instance, if one looks for solutions $y(t) = e^{\lambda t}v$ in the system of dth order ALGEBRAIC-ODEs

$$P_d \frac{d^d y(t)}{dt^d} + \cdots + P_1 \frac{dy(t)}{dt} + P_0 y(t) = 0 \Longrightarrow (P_d\lambda^d + \cdots + P_1\lambda + P_0)v = 0$$

- There are stable (? debatable) algorithms for its numerical solution.
- Easy to use software for small to medium size dense matrices: MATLAB's commands `polyeig(P0,P1,...,Pd)` (Van Dooren, 1979).
- Different **specific-structured Krylov methods** for large-scale problems and sparse matrix coefficients (Su-Bai-Lu, 2008, 2016), (Van Beeumen-Meerbergen-Michiels, 2015).
- HPC implementations of these methods in **SLEPc** (Roman, UPV).

The RATIONAL eigenvalue problem (REP). Given a rational matrix $G(z) \in \mathbb{C}(z)^{n\times n}$, i.e.,

$$\begin{array}{rccc} G: & \mathbb{C} & \to & \mathbb{C}^{n\times n} \\ & z & \mapsto & G(z), \end{array}$$

such that $G(z)_{ij}$ is a scalar rational function of $z \in \mathbb{C}$, for $1 \leq i, j \leq n$, compute scalars λ (eigenvalues) and nonzero vectors $v \in \mathbb{C}^n$ (eigenvectors) such that λ is not a pole of any $G(z)_{ij}$ and

$$\boxed{G(\lambda)v = 0},$$

under the regularity assumption $\det(G(z)) \not\equiv 0$.

- It arises in applications either directly (multivariable system theory and control theory) or as an approximation.
- There are algorithms for its numerical solution (stability analysis open).
- For small to medium size dense matrices (Su-Bai, 2011).
- For large-scale problems and sparse matrix coefficients (Van Beeumen-Meerbergen-Michiels, 2015), (D & González-Pizarro, 2018).
- HPC implementations of some methods in **SLEPc** (Roman, UPV).

The RATIONAL eigenvalue problem (REP). Given a rational matrix $G(z) \in \mathbb{C}(z)^{n\times n}$, i.e.,

$$\begin{array}{rccc} G: & \mathbb{C} & \to & \mathbb{C}^{n\times n} \\ & z & \mapsto & G(z), \end{array}$$

such that $G(z)_{ij}$ is a scalar rational function of $z \in \mathbb{C}$, for $1 \leq i, j \leq n$, compute scalars λ (eigenvalues) and nonzero vectors $v \in \mathbb{C}^n$ (eigenvectors) such that λ is not a pole of any $G(z)_{ij}$ and

$$\boxed{G(\lambda)v = 0},$$

under the regularity assumption $\det(G(z)) \not\equiv 0$.

- It arises in applications either directly (multivariable system theory and control theory) or as an approximation.
- There are algorithms for its numerical solution (stability analysis open).
- For small to medium size dense matrices (Su-Bai, 2011).
- For large-scale problems and sparse matrix coefficients (Van Beeumen-Meerbergen-Michiels, 2015), (D & González-Pizarro, 2018).
- HPC implementations of some methods in **SLEPc** (Roman, UPV).

The RATIONAL eigenvalue problem (REP). Given a rational matrix $G(z) \in \mathbb{C}(z)^{n\times n}$, i.e.,

$$\begin{array}{cccc} G: & \mathbb{C} & \to & \mathbb{C}^{n\times n} \\ & z & \mapsto & G(z), \end{array}$$

such that $G(z)_{ij}$ is a scalar rational function of $z \in \mathbb{C}$, for $1 \le i, j \le n$, compute scalars λ (eigenvalues) and nonzero vectors $v \in \mathbb{C}^n$ (eigenvectors) such that λ is not a pole of any $G(z)_{ij}$ and

$$\boxed{G(\lambda)v = 0},$$

under the regularity assumption $\det(G(z)) \not\equiv 0$.

- It arises in applications either directly (multivariable system theory and control theory) or as an approximation.
- There are algorithms for its numerical solution (stability analysis open).
- For small to medium size dense matrices (Su-Bai, 2011).
- For large-scale problems and sparse matrix coefficients (Van Beeumen-Meerbergen-Michiels, 2015), (D & González-Pizarro, 2018).
- HPC implementations of some methods in **SLEPc** (Roman, UPV).

The RATIONAL eigenvalue problem (REP). Given a rational matrix $G(z) \in \mathbb{C}(z)^{n\times n}$, i.e.,

$$\begin{array}{rccc} G: & \mathbb{C} & \to & \mathbb{C}^{n\times n} \\ & z & \mapsto & G(z), \end{array}$$

such that $G(z)_{ij}$ is a scalar rational function of $z \in \mathbb{C}$, for $1 \leq i, j \leq n$, compute scalars λ (eigenvalues) and nonzero vectors $v \in \mathbb{C}^n$ (eigenvectors) such that λ is not a pole of any $G(z)_{ij}$ and

$$\boxed{G(\lambda)v = 0},$$

under the regularity assumption $\det(G(z)) \not\equiv 0$.

- It arises in applications either directly (multivariable system theory and control theory) or as an approximation.
- There are algorithms for its numerical solution (stability analysis open).
- For small to medium size dense matrices (Su-Bai, 2011).
- For large-scale problems and sparse matrix coefficients (Van Beeumen-Meerbergen-Michiels, 2015), (D & González-Pizarro, 2018).
- HPC implementations of some methods in **SLEPc** (Roman, UPV).

The RATIONAL eigenvalue problem (REP). Given a rational matrix $G(z) \in \mathbb{C}(z)^{n\times n}$, i.e.,

$$\begin{array}{rccc} G: & \mathbb{C} & \to & \mathbb{C}^{n\times n} \\ & z & \mapsto & G(z), \end{array}$$

such that $G(z)_{ij}$ is a scalar rational function of $z \in \mathbb{C}$, for $1 \leq i, j \leq n$, compute scalars λ (eigenvalues) and nonzero vectors $v \in \mathbb{C}^n$ (eigenvectors) such that λ is not a pole of any $G(z)_{ij}$ and

$$\boxed{G(\lambda)v = 0},$$

under the regularity assumption $\det(G(z)) \not\equiv 0$.

- It arises in applications either directly (multivariable system theory and control theory) or as an approximation.
- There are algorithms for its numerical solution (stability analysis open).
- For small to medium size dense matrices (Su-Bai, 2011).
- For large-scale problems and sparse matrix coefficients (Van Beeumen-Meerbergen-Michiels, 2015), (D & González-Pizarro, 2018).
- HPC implementations of some methods in SLEPc (Roman, UPV).

The RATIONAL eigenvalue problem (REP). Given a rational matrix $G(z) \in \mathbb{C}(z)^{n \times n}$, i.e.,

$$\begin{array}{rccc} G: & \mathbb{C} & \to & \mathbb{C}^{n\times n} \\ & z & \mapsto & G(z), \end{array}$$

such that $G(z)_{ij}$ is a scalar rational function of $z \in \mathbb{C}$, for $1 \leq i, j \leq n$, compute scalars λ (eigenvalues) and nonzero vectors $v \in \mathbb{C}^n$ (eigenvectors) such that λ is not a pole of any $G(z)_{ij}$ and

$$\boxed{G(\lambda)v = 0},$$

under the regularity assumption $\det(G(z)) \not\equiv 0$.

- It arises in applications either directly (multivariable system theory and control theory) or as an approximation.
- There are algorithms for its numerical solution (stability analysis open).
- For small to medium size dense matrices (Su-Bai, 2011).
- For large-scale problems and sparse matrix coefficients (Van Beeumen-Meerbergen-Michiels, 2015), (D & González-Pizarro, 2018).
- HPC implementations of some methods in **SLEPc** (Roman, UPV).

The NONLINEAR eigenvalue problem (NEP). Given a non-empty open set $\Omega \subseteq \mathbb{C}$ and a matrix-valued function

$$\begin{array}{cccc} F: & \Omega & \to & \mathbb{C}^{n\times n} \\ & z & \mapsto & F(z), \end{array}$$

compute scalars $\lambda \in \Omega$ (eigenvalues) and nonzero vectors $v \in \mathbb{C}^n$ (eigenvectors) such that

$$\boxed{F(\lambda)v = 0},$$

under the regularity assumption $\det(F(z)) \not\equiv 0$.

- It arises in applications. For instance, if one looks for solutions $y(t) = e^{\lambda t}v$ in the system of first order DELAYED differential equations

$$\frac{dy(t)}{dt} + Ay(t) + By(t-1) = 0 \Longrightarrow (\lambda I_n + A + Be^{-\lambda})v = 0$$

- Usually $F(z)$ is assumed to be holomorphic in Ω.

The NONLINEAR eigenvalue problem (NEP). Given a non-empty open set $\Omega \subseteq \mathbb{C}$ and a matrix-valued function

$$\begin{array}{cccc} F: & \Omega & \to & \mathbb{C}^{n\times n} \\ & z & \mapsto & F(z), \end{array}$$

compute scalars $\lambda \in \Omega$ (eigenvalues) and nonzero vectors $v \in \mathbb{C}^n$ (eigenvectors) such that

$$\boxed{F(\lambda)v = 0},$$

under the regularity assumption $\det(F(z)) \not\equiv 0$.

- It arises in applications. For instance, if one looks for solutions $y(t) = e^{\lambda t}v$ in the system of first order DELAYED differential equations

$$\frac{dy(t)}{dt} + Ay(t) + By(t-1) = 0 \Longrightarrow (\lambda I_n + A + Be^{-\lambda})v = 0$$

- Usually $F(z)$ is assumed to be holomorphic in Ω.

The NONLINEAR eigenvalue problem (NEP). Given a non-empty open set $\Omega \subseteq \mathbb{C}$ and a matrix-valued function

$$\begin{array}{rccc} F: & \Omega & \rightarrow & \mathbb{C}^{n\times n} \\ & z & \mapsto & F(z), \end{array}$$

compute scalars $\lambda \in \Omega$ (eigenvalues) and nonzero vectors $v \in \mathbb{C}^n$ (eigenvectors) such that

$$\boxed{F(\lambda)v = 0},$$

under the regularity assumption $\det(F(z)) \not\equiv 0$.

- It arises in applications. For instance, if one looks for solutions $y(t) = e^{\lambda t}v$ in the system of first order DELAYED differential equations

$$\frac{dy(t)}{dt} + Ay(t) + By(t-1) = 0 \Longrightarrow (\lambda I_n + A + Be^{-\lambda})v = 0$$

- Usually $F(z)$ is assumed to be holomorphic in Ω.

Different classes of regular matrix eigenvalue problems (V) (continued)

$$\begin{array}{rccc} F: & \Omega & \rightarrow & \mathbb{C}^{n\times n} \\ & z & \mapsto & F(z) \end{array} \qquad \boxed{F(\lambda)v = 0}$$

- There are different algorithms for the numerical solution of NEP.
- One of the most important family of algorithms is based on the following two step strategy
  1. Approximate $F(z)$ by a rational matrix $G(z)$ with poles outside Ω.
  2. Solve the REP associated to $G(z)$.
- There is software available for NEPs developed by the authors of some key papers that follow the previous strategy:
  1. NLEIGS (Güttel, Van Beeumen, Meerbergen, Michiels, 2014) (not easy to use).
  2. Automatic Rational Approximation and Linearization of NEP (Lietaert, Pérez, Vandereycken, Meerbergen, 2018) (the authors claim that is easy to use and good).

Different classes of regular matrix eigenvalue problems (V) (continued)

$$\begin{array}{rccc} F: & \Omega & \to & \mathbb{C}^{n\times n} \\ & z & \mapsto & F(z) \end{array} \qquad \boxed{F(\lambda)v = 0}$$

- There are different algorithms for the numerical solution of NEP.
- One of the most important family of algorithms is based on the following two step strategy
    1. Approximate $F(z)$ by a rational matrix $G(z)$ with poles outside Ω.
    2. Solve the REP associated to $G(z)$.
- There is software available for NEPs developed by the authors of some key papers that follow the previous strategy:
    1. NLEIGS (Güttel, Van Beeumen, Meerbergen, Michiels, 2014) (not easy to use).
    2. Automatic Rational Approximation and Linearization of NEP (Lietaert, Pérez, Vandereycken, Meerbergen, 2018) (the authors claim that is easy to use and good).

$$\begin{array}{rccc} F: & \Omega & \rightarrow & \mathbb{C}^{n\times n} \\ & z & \mapsto & F(z) \end{array} \qquad \boxed{F(\lambda)v = 0}$$

- There are different algorithms for the numerical solution of NEP.
- One of the most important family of algorithms is based on the following two step strategy
  1. Approximate $F(z)$ by a rational matrix $G(z)$ with poles outside Ω.
  2. Solve the REP associated to $G(z)$.
- There is software available for NEPs developed by the authors of some key papers that follow the previous strategy:
  1. NLEIGS (Güttel, Van Beeumen, Meerbergen, Michiels, 2014) (not easy to use).
  2. Automatic Rational Approximation and Linearization of NEP (Lietaert, Pérez, Vandereycken, Meerbergen, 2018) (the authors claim that is easy to use and good).

1st KEY IDEA on matrix eigenvalue problems

1. **BEP**: $\boxed{(\lambda I_n - A)\, v = 0}$
2. **GEP**: $\boxed{(\lambda B - A)\, v = 0}$
3. **PEP**: $\boxed{(P_d \lambda^d + \cdots + P_1 \lambda + P_0) v = 0}$
4. **REP**: $\boxed{G(\lambda) v = 0}$
5. **NEP**: $\boxed{F(\lambda) v = 0}$

- **1st KEY IDEA: ALL THESE PROBLEMS CAN BE SOLVED BY TRANSFORMING THE PROBLEM INTO A GEP $\rightarrow$ LINEARIZATION.**
- For **PEPs** and **REPs**, this transformation is **exact**.
- For **NEPs**, this transformation requires to **approximate** the NEP by a **REP**, but all current methods for NEPs require some approximation.
- The use of **linearizations** is (probably) the **MOST RELIABLE** approach to solve numerically these problems.

1st KEY IDEA on matrix eigenvalue problems

1. **BEP**: $(\lambda I_n - A)\, v = 0$
2. **GEP**: $(\lambda B - A)\, v = 0$
3. **PEP**: $(P_d \lambda^d + \cdots + P_1 \lambda + P_0) v = 0$
4. **REP**: $G(\lambda) v = 0$
5. **NEP**: $F(\lambda) v = 0$

- **1st KEY IDEA: ALL THESE PROBLEMS CAN BE SOLVED BY TRANSFORMING THE PROBLEM INTO A GEP $\rightarrow$ LINEARIZATION.**
- For **PEPs** and **REPs**, this transformation is **exact**.
- For **NEPs**, this transformation requires to **approximate** the NEP by a **REP**, but all current methods for NEPs require some approximation.
- The use of **linearizations** is (probably) the **MOST RELIABLE** approach to solve numerically these problems.

1st KEY IDEA on matrix eigenvalue problems

1. **BEP**: $\boxed{(\lambda I_n - A)\, v = 0}$
2. **GEP**: $\boxed{(\lambda B - A)\, v = 0}$
3. **PEP**: $\boxed{(P_d \lambda^d + \cdots + P_1 \lambda + P_0) v = 0}$
4. **REP**: $\boxed{G(\lambda) v = 0}$
5. **NEP**: $\boxed{F(\lambda) v = 0}$

- **1st KEY IDEA: ALL THESE PROBLEMS CAN BE SOLVED BY TRANSFORMING THE PROBLEM INTO A GEP $\rightarrow$ LINEARIZATION.**
- For **PEPs** and **REPs**, this transformation is **exact**.
- For **NEPs**, this transformation requires to **approximate** the NEP by a **REP**, but all current methods for NEPs require some approximation.
- The use of **linearizations** is (probably) the **MOST RELIABLE** approach to solve numerically these problems.

1st KEY IDEA on matrix eigenvalue problems

1. **BEP**: $\boxed{(\lambda I_n - A)\,v = 0}$
2. **GEP**: $\boxed{(\lambda B - A)\,v = 0}$
3. **PEP**: $\boxed{(P_d\lambda^d + \cdots + P_1\lambda + P_0)v = 0}$
4. **REP**: $\boxed{G(\lambda)v = 0}$
5. **NEP**: $\boxed{F(\lambda)v = 0}$

- **1st KEY IDEA: ALL THESE PROBLEMS CAN BE SOLVED BY TRANSFORMING THE PROBLEM INTO A GEP $\rightarrow$ LINEARIZATION.**
- For **PEPs** and **REPs**, this transformation is **exact**.
- For **NEPs**, this transformation requires to **approximate** the NEP by a **REP**, but all current methods for NEPs require some approximation.
- The use of **linearizations** is (probably) the **MOST RELIABLE** approach to solve numerically these problems.

1st KEY IDEA on matrix eigenvalue problems

1. **BEP**: $(\lambda I_n - A)\, v = 0$
2. **GEP**: $(\lambda B - A)\, v = 0$
3. **PEP**: $(P_d \lambda^d + \cdots + P_1 \lambda + P_0) v = 0$
4. **REP**: $G(\lambda) v = 0$
5. **NEP**: $F(\lambda) v = 0$

- **1st KEY IDEA: ALL THESE PROBLEMS CAN BE SOLVED BY TRANSFORMING THE PROBLEM INTO A GEP $\to$ LINEARIZATION.**
- For **PEPs** and **REPs**, this transformation is **exact**.
- For **NEPs**, this transformation requires to **approximate** the NEP by a **REP**, but all current methods for NEPs require some approximation.
- The use of **linearizations** is (probably) the **MOST RELIABLE** approach to solve numerically these problems.

2nd KEY IDEA on matrix eigenvalue problems

1. **BEP**: $(\lambda I_n - A)\, v = 0$
2. **GEP**: $(\lambda B - A)\, v = 0$
3. **PEP**: $(P_d \lambda^d + \cdots + P_1 \lambda + P_0) v = 0$
4. **REP**: $G(\lambda) v = 0$
5. **NEP**: $F(\lambda) v = 0$

- **2nd KEY IDEA:** usual methods for transforming a **PEP** or a **REP** into a **GEP**, **produce highly structured GEPs**.
- **These structures must be used**, in particular in large-scale problems, **for developing efficient algorithms for PEPs or REPs**.

2nd KEY IDEA on matrix eigenvalue problems

1. **BEP**: $(\lambda I_n - A)\, v = 0$
2. **GEP**: $(\lambda B - A)\, v = 0$
3. **PEP**: $(P_d \lambda^d + \cdots + P_1 \lambda + P_0) v = 0$
4. **REP**: $G(\lambda) v = 0$
5. **NEP**: $F(\lambda) v = 0$

- **2nd KEY IDEA:** usual methods for transforming a **PEP** or a **REP** into a **GEP**, **produce highly structured GEPs**.
- **These structures must be used**, in particular in large-scale problems, **for developing efficient algorithms for PEPs or REPs**.

2nd KEY IDEA on matrix eigenvalue problems

1. **BEP**: $(\lambda I_n - A)\, v = 0$
2. **GEP**: $(\lambda B - A)\, v = 0$
3. **PEP**: $(P_d \lambda^d + \cdots + P_1 \lambda + P_0) v = 0$
4. **REP**: $G(\lambda) v = 0$
5. **NEP**: $F(\lambda) v = 0$

- **2nd KEY IDEA:** usual methods for transforming a **PEP** or a **REP** into a **GEP**, **produce highly structured GEPs**.
- **These structures must be used**, in particular in large-scale problems, **for developing efficient algorithms for PEPs or REPs**.

Outline

- Every matrix $F(z)$ defining an $n \times n$ PEP, REP or NEP can be written in **“split form”** with at most n^2 terms, i.e.,

$$F(z) = f_1(z)\, C_1 + f_2(z)\, C_2 + \cdots + f_\ell(z)\, C_\ell,$$

  where $f_i : \mathbb{C} \to \mathbb{C}$, $C_i \in \mathbb{C}^{n\times n}$, and $\ell \leq n^2$.

- This result is, of course, a triviality,

$$\begin{bmatrix} e^z & z^2+1 \\ \frac{1}{z+1} & \sin(z) \end{bmatrix} = e^z \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} + (z^2+1) \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} + \frac{1}{z+1} \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} + \sin(z) \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$$

- The **3rd KEY IDEA is that in most applications $\ell \ll n^2$**,
- this is not important in theoretical developments, but yes in the development of algorithms and in the practical approximation of NEPs by REPs or PEPs.
- Our scenario is **large matrices C_i and very few scalar functions $f_i(z)$**.

- Every matrix $F(z)$ defining an $n \times n$ PEP, REP or NEP can be written in **"split form"** with at most n^2 terms, i.e.,

$$F(z) = f_1(z)\, C_1 + f_2(z)\, C_2 + \cdots + f_\ell(z)\, C_\ell,$$

where $f_i : \mathbb{C} \to \mathbb{C}$, $C_i \in \mathbb{C}^{n \times n}$, and $\ell \leq n^2$.

- This result is, of course, a triviality,

$$\begin{bmatrix} e^z & z^2+1 \\ \frac{1}{z+1} & \sin(z) \end{bmatrix} = e^z \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} + (z^2+1) \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} + \frac{1}{z+1} \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} + \sin(z) \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$$

- The **3rd KEY IDEA is that in most applications** $\ell \ll n^2$,
- this is not important in theoretical developments, but yes in the development of algorithms and in the practical approximation of NEPs by REPs or PEPs.
- Our scenario is **large matrices** C_i **and very few scalar functions** $f_i(z)$.

3rd KEY IDEA: applications usually lead to very shorts "split forms"

- Every matrix $F(z)$ defining an $n \times n$ PEP, REP or NEP can be written in **"split form"** with at most n^2 terms, i.e.,

$$F(z) = f_1(z)\, C_1 + f_2(z)\, C_2 + \cdots + f_\ell(z)\, C_\ell,$$

where $f_i : \mathbb{C} \to \mathbb{C}$, $C_i \in \mathbb{C}^{n \times n}$, and $\ell \leq n^2$.

- This result is, of course, a triviality,

$$\begin{bmatrix} e^z & z^2+1 \\ \frac{1}{z+1} & \sin(z) \end{bmatrix} = e^z \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} + (z^2+1) \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} + \frac{1}{z+1} \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} + \sin(z) \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$$

- The **3rd KEY IDEA is that in most applications** $\ell \ll n^2$,
- this is not important in theoretical developments, but yes in the development of algorithms and in the practical approximation of NEPs by REPs or PEPs.
- Our scenario is **large matrices C_i and very few scalar functions $f_i(z)$**.

- Every matrix $F(z)$ defining an $n \times n$ PEP, REP or NEP can be written in **"split form"** with at most n^2 terms, i.e.,

$$F(z) = f_1(z)\, C_1 + f_2(z)\, C_2 + \cdots + f_\ell(z)\, C_\ell,$$

where $f_i : \mathbb{C} \to \mathbb{C}$, $C_i \in \mathbb{C}^{n\times n}$, and $\ell \leq n^2$.

- This result is, of course, a triviality,

$$\begin{bmatrix} e^z & z^2+1 \\ \frac{1}{z+1} & \sin(z) \end{bmatrix} = e^z \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} + (z^2+1) \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} + \frac{1}{z+1} \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} + \sin(z) \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$$

- The **3rd KEY IDEA is that in most applications** $\ell \ll n^2$,
- this is not important in theoretical developments, but yes in the development of algorithms and in the practical approximation of NEPs by REPs or PEPs.
- Our scenario is **large matrices** C_i **and very few scalar functions** $f_i(z)$.

3rd KEY IDEA: applications usually lead to very shorts "split forms"

- Every matrix $F(z)$ defining an $n \times n$ PEP, REP or NEP can be written in **"split form"** with at most n^2 terms, i.e.,

$$F(z) = f_1(z)\, C_1 + f_2(z)\, C_2 + \cdots + f_\ell(z)\, C_\ell,$$

where $f_i : \mathbb{C} \to \mathbb{C}$, $C_i \in \mathbb{C}^{n \times n}$, and $\ell \leq n^2$.

- This result is, of course, a triviality,

$$\begin{bmatrix} e^z & z^2+1 \\ \frac{1}{z+1} & \sin(z) \end{bmatrix} = e^z \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} + (z^2+1) \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} + \frac{1}{z+1} \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} + \sin(z) \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$$

- The **3rd KEY IDEA is that in most applications $\ell \ll n^2$**,
- this is not important in theoretical developments, but yes in the development of algorithms and in the practical approximation of NEPs by REPs or PEPs.
- Our scenario is **large matrices C_i and very few scalar functions $f_i(z)$**.

How large is the degree of $P(z) = P_d z^d + \cdots + P_1 z + P_0$ in practical PEPs?

- In most direct applications coming from vibrational problems in mechanics $\mathbf{d = 2}$: **the quadratic eigenvalue problem (QEP)**

$$(z^2 M + zC + K)v = 0,$$

while $M, C, K \in \mathbb{C}^{n \times n}$ with $\mathbf{n = 10^3, 10^4, 10^5, 10^6, \ldots}$.

- Betcke, Higham, Mehrmann, Schröder, Tisseur, *"NLEVP: A Collection of Nonlinear Eigenvalue Problems"*, (2013) reports on applications with
  - $\mathbf{d = 4}$: Hamiltonian control problems, homography-based method for calibrating a central cadioptric vision system, spatial stability analysis of the Orr-Sommerfeld equation, and finite element solution of the equation for the modes of a planar waveguide using piecewise linear basis functions.
  - $\mathbf{d = 3}$: modeling of drift instabilities in the plasma edge inside a Tokamak reactor, and the five point relative pose problem in computer vision.

- PEPs used to approximate other NEPs. Then d can be much larger. Kressner and Roman (2014) report on $\mathbf{d = 30, n = 10000}$ and $\mathbf{d = 11, n = 6223}$.

How large is the degree of $P(z) = P_d z^d + \cdots + P_1 z + P_0$ in practical PEPs?

- In most direct applications coming from vibrational problems in mechanics $\mathbf{d = 2}$: **the quadratic eigenvalue problem (QEP)**

$$(z^2 M + zC + K)v = 0,$$

while $M, C, K \in \mathbb{C}^{n \times n}$ with $\mathbf{n = 10^3, 10^4, 10^5, 10^6, \ldots}$.

- Betcke, Higham, Mehrmann, Schröder, Tisseur, *"NLEVP: A Collection of Nonlinear Eigenvalue Problems"*, (2013) reports on applications with
  - $\mathbf{d = 4}$: Hamiltonian control problems, homography-based method for calibrating a central cadioptric vision system, spatial stability analysis of the Orr-Sommerfeld equation, and finite element solution of the equation for the modes of a planar waveguide using piecewise linear basis functions.
  - $\mathbf{d = 3}$: modeling of drift instabilities in the plasma edge inside a Tokamak reactor, and the five point relative pose problem in computer vision.

- PEPs used to approximate other NEPs. Then d can be much larger. Kressner and Roman (2014) report on $\mathbf{d = 30, n = 10000}$ and $\mathbf{d = 11, n = 6223}$.

- In most direct applications coming from vibrational problems in mechanics $\mathbf{d = 2}$: **the quadratic eigenvalue problem (QEP)**

$$(z^2 M + zC + K)v = 0,$$

while $M, C, K \in \mathbb{C}^{n \times n}$ with $\mathbf{n = 10^3, 10^4, 10^5, 10^6, \ldots}$.

- Betcke, Higham, Mehrmann, Schröder, Tisseur, *"NLEVP: A Collection of Nonlinear Eigenvalue Problems"*, (2013) reports on applications with
  - $\mathbf{d = 4}$: Hamiltonian control problems, homography-based method for calibrating a central cadioptric vision system, spatial stability analysis of the Orr-Sommerfeld equation, and finite element solution of the equation for the modes of a planar waveguide using piecewise linear basis functions.
  - $\mathbf{d = 3}$: modeling of drift instabilities in the plasma edge inside a Tokamak reactor, and the five point relative pose problem in computer vision.

- PEPs used to approximate other NEPs. Then d can be much larger. Kressner and Roman (2014) report on $\mathbf{d = 30, n = 10000}$ and $\mathbf{d = 11, n = 6223}$.

How large is the degree of $P(z) = P_d z^d + \cdots + P_1 z + P_0$ in practical PEPs?

- In most direct applications coming from vibrational problems in mechanics $\mathbf{d = 2}$: **the quadratic eigenvalue problem (QEP)**

$$(z^2 M + zC + K)v = 0,$$

  while $M, C, K \in \mathbb{C}^{n \times n}$ with $\mathbf{n = 10^3, 10^4, 10^5, 10^6, \ldots}$.

- Betcke, Higham, Mehrmann, Schröder, Tisseur, *"NLEVP: A Collection of Nonlinear Eigenvalue Problems"*, (2013) reports on applications with
  - $\mathbf{d = 4}$: Hamiltonian control problems, homography-based method for calibrating a central cadioptric vision system, spatial stability analysis of the Orr-Sommerfeld equation, and finite element solution of the equation for the modes of a planar waveguide using piecewise linear basis functions.
  - $\mathbf{d = 3}$: modeling of drift instabilities in the plasma edge inside a Tokamak reactor, and the five point relative pose problem in computer vision.

- PEPs used to approximate other NEPs. Then d can be much larger. Kressner and Roman (2014) report on $\mathbf{d = 30, n = 10000}$ and $\mathbf{d = 11, n = 6223}$.

How large is the degree of $P(z) = P_d z^d + \cdots + P_1 z + P_0$ in practical PEPs?

- In most direct applications coming from vibrational problems in mechanics $\mathbf{d = 2}$: **the quadratic eigenvalue problem (QEP)**

$$(z^2 M + zC + K)v = 0,$$

while $M, C, K \in \mathbb{C}^{n \times n}$ with $\mathbf{n = 10^3, 10^4, 10^5, 10^6, \ldots}$.

- Betcke, Higham, Mehrmann, Schröder, Tisseur, *"NLEVP: A Collection of Nonlinear Eigenvalue Problems"*, (2013) reports on applications with
  - $\mathbf{d = 4}$: Hamiltonian control problems, homography-based method for calibrating a central cadioptric vision system, spatial stability analysis of the Orr-Sommerfeld equation, and finite element solution of the equation for the modes of a planar waveguide using piecewise linear basis functions.
  - $\mathbf{d = 3}$: modeling of drift instabilities in the plasma edge inside a Tokamak reactor, and the five point relative pose problem in computer vision.
- PEPs used to approximate other NEPs. Then d can be much larger. Kressner and Roman (2014) report on $\mathbf{d = 30, n = 10000}$ and $\mathbf{d = 11, n = 6223}$.

A few examples of "direct" applied REPs

Change of notation $z \to \lambda$

- Loaded elastic string (Betcke et al., NLEVP-collection, (2013)):

$$G(\lambda) = A - \lambda B + \frac{\lambda}{\lambda - \sigma} E = (A + E) - \lambda B + \frac{\sigma}{\lambda - \sigma} E,$$

which almost shows the polynomial and the strictly proper parts of $G(\lambda)$. Only 3 functions (terms) in split form, $A, B \in \mathbb{R}^{n \times n}$ symmetric tridiagonal matrices, E only one nonzero entry in (n, n) position. $n \geq 10^3$ large.

- Damped vibration of a viscoelastic structure (Mehrmann & Voss, (2004)):

$$G(\lambda) = \lambda^2 M + K - \sum_{i=1}^{k} \frac{1}{1 + b_i \lambda} \Delta G_i,$$

which shows the polynomial and the strictly proper parts of $G(\lambda)$. Only $k + 2$ functions in split form, M, K positive definite, $n = 10704$ large.

Change of notation $z \to \lambda$

- Loaded elastic string (Betcke et al., NLEVP-collection, (2013)):

$$G(\lambda) = A - \lambda B + \frac{\lambda}{\lambda - \sigma} E = (A + E) - \lambda B + \frac{\sigma}{\lambda - \sigma} E,$$

which almost shows the polynomial and the strictly proper parts of $G(\lambda)$. Only 3 functions (terms) in split form, $A, B \in \mathbb{R}^{n \times n}$ symmetric tridiagonal matrices, E only one nonzero entry in (n, n) position. $n \geq 10^3$ large.

- Damped vibration of a viscoelastic structure (Mehrmann & Voss, (2004)):

$$G(\lambda) = \lambda^2 M + K - \sum_{i=1}^{k} \frac{1}{1 + b_i \lambda} \Delta G_i,$$

which shows the polynomial and the strictly proper parts of $G(\lambda)$. Only $k + 2$ functions in split form, M, K positive definite, $n = 10704$ large.

Change of notation $z \to \lambda$

- Loaded elastic string (Betcke et al., NLEVP-collection, (2013)):

$$G(\lambda) = A - \lambda B + \frac{\lambda}{\lambda - \sigma} E = (A + E) - \lambda B + \frac{\sigma}{\lambda - \sigma} E,$$

  which almost shows the polynomial and the strictly proper parts of $G(\lambda)$. Only 3 functions (terms) in split form, $A, B \in \mathbb{R}^{n \times n}$ symmetric tridiagonal matrices, E only one nonzero entry in (n, n) position. $n \geq 10^3$ large.

- Damped vibration of a viscoelastic structure (Mehrmann & Voss, (2004)):

$$G(\lambda) = \lambda^2 M + K - \sum_{i=1}^{k} \frac{1}{1 + b_i \lambda} \Delta G_i,$$

  which shows the polynomial and the strictly proper parts of $G(\lambda)$. Only $k + 2$ functions in split form, M, K positive definite, $n = 10704$ large.

- NLEIGS-REPs coming from linear rational interpolation of NEPs (Güttel, Van Beeumen, Meerbergen, Michiels (2014)):

$$Q_N(\lambda) = b_0(\lambda)D_0 + b_1(\lambda)D_1 + \cdots + b_N(\lambda)D_N,$$

with $D_j \in \mathbb{C}^{n\times n}$,

$$b_j(\lambda) = \frac{1}{\beta_0}\prod_{k=1}^{j}\frac{\lambda-\sigma_{k-1}}{\beta_k(1-\lambda/\xi_k)},$$

$j = 0, 1, \ldots, N$, rational scalar functions, with the poles ξ_i all distinct from the nodes σ_j. $N \leq 140$, $n = 16281$.

among many others, **the following NEPs**:

- **The radio-frequency gun cavity problem**:

$$\left[(K-\lambda M)+i\sqrt{\lambda-\sigma_1^2}\,W_1+i\sqrt{\lambda-\sigma_2^2}\,W_2\right]v=0,$$

  where M, K, W_1, W_2 are real sparse symmetric 9956×9956 matrices (only 4 scalar functions involved in split form).

- **Bound states in semiconductor devices problems**:

$$\left[(H-\lambda I)+\sum_{j=0}^{80} e^{i\sqrt{\lambda-\alpha_j}}\,S_j\right]v=0,$$

  where $H, S_j \in \mathbb{R}^{16281\times 16281}$, H symmetric and the matrices S_j have low rank (only 83 scalar functions involved in split form).

- ....

among many others, **the following NEPs**:

- **The radio-frequency gun cavity problem**:

$$\left[(K - \lambda M) + i\sqrt{\lambda - \sigma_1^2}\, W_1 + i\sqrt{\lambda - \sigma_2^2}\, W_2\right] v = 0,$$

where M, K, W_1, W_2 are real sparse symmetric 9956×9956 matrices (only 4 scalar functions involved in split form).

- **Bound states in semiconductor devices problems**:

$$\left[(H - \lambda I) + \sum_{j=0}^{80} e^{i\sqrt{\lambda - \alpha_j}}\, S_j\right] v = 0,$$

where $H, S_j \in \mathbb{R}^{16281 \times 16281}$, H symmetric and the matrices S_j have low rank (only 83 scalar functions involved in split form).

- ....

among many others, **the following NEPs**:

- **The radio-frequency gun cavity problem**:

$$\left[(K - \lambda M) + i\sqrt{\lambda - \sigma_1^2}\, W_1 + i\sqrt{\lambda - \sigma_2^2}\, W_2 \right] v = 0,$$

  where M, K, W_1, W_2 are real sparse symmetric 9956×9956 matrices (only 4 scalar functions involved in split form).

- **Bound states in semiconductor devices problems**:

$$\left[(H - \lambda I) + \sum_{j=0}^{80} e^{i\sqrt{\lambda - \alpha_j}}\, S_j \right] v = 0,$$

  where $H, S_j \in \mathbb{R}^{16281 \times 16281}$, H symmetric and the matrices S_j have low rank (only 83 scalar functions involved in split form).

- ....

For the rest of the talk...

- since NEPs can be approximated by REPs, **I will focus on PEPs and REPs**.
- Though PEPs are mathematically a particular case of REPs,
- PEPs and REPs have been always considered separately from the point of view of numerical algorithms,
- because PEPs are very important by themselves in applications (the quadratic PEP in mechanical problems, in particular), and also
- because PEPs are numerically simpler and "better" specific algorithms can be developed for them.

- since NEPs can be approximated by REPs, **I will focus on PEPs and REPs**.
- Though PEPs are mathematically a particular case of REPs,
- PEPs and REPs have been always considered separately from the point of view of numerical algorithms,
- because PEPs are very important by themselves in applications (the quadratic PEP in mechanical problems, in particular), and also
- because PEPs are numerically simpler and “better” specific algorithms can be developed for them.

For the rest of the talk...

- since NEPs can be approximated by REPs, **I will focus on PEPs and REPs**.
- Though PEPs are mathematically a particular case of REPs,
- PEPs and REPs have been always considered separately from the point of view of numerical algorithms,
- because PEPs are very important by themselves in applications (the quadratic PEP in mechanical problems, in particular), and also
- because PEPs are numerically simpler and “better” specific algorithms can be developed for them.

For the rest of the talk...

- since NEPs can be approximated by REPs, **I will focus on PEPs and REPs**.
- Though PEPs are mathematically a particular case of REPs,
- PEPs and REPs have been always considered separately from the point of view of numerical algorithms,
- because PEPs are very important by themselves in applications (the quadratic PEP in mechanical problems, in particular), and also
- because PEPs are numerically simpler and "better" specific algorithms can be developed for them.

- since NEPs can be approximated by REPs, **I will focus on PEPs and REPs**.
- Though PEPs are mathematically a particular case of REPs,
- PEPs and REPs have been always considered separately from the point of view of numerical algorithms,
- because PEPs are very important by themselves in applications (the quadratic PEP in mechanical problems, in particular), and also
- because PEPs are numerically simpler and "better" specific algorithms can be developed for them.

Outline

1. **BEP**: $(\lambda I_n - A)\, v = 0$
2. **GEP**: $(\lambda B - A)\, v = 0$
3. **PEP**: $(P_d \lambda^d + \cdots + P_1 \lambda + P_0) v = 0$
4. **REP**: $G(\lambda) v = 0$

- So far, we have only considered **finite eigenvalues**, but
- regular **GEPs, PEPs, REPs** may have also **infinite eigenvalues**.
- **GEPs, PEPs, REPs** may be **singular** (BEPs are always regular) and to have, in addition to eigenvalues, **minimal indices**.
- **REPs** have **poles**. In modern applications, the poles are usually known (even chosen in approximating NEPs by REPs), but in other applications (Control), poles are not known and must be computed.
- We illustrate informally some of these concepts on matrix polynomials...

GEPs-PEPs-REPs have more spectral "structural" data than BEPs

1. **BEP**: $(\lambda I_n - A)\, v = 0$
2. **GEP**: $(\lambda B - A)\, v = 0$
3. **PEP**: $(P_d \lambda^d + \cdots + P_1 \lambda + P_0) v = 0$
4. **REP**: $G(\lambda) v = 0$

- So far, we have only considered **finite eigenvalues**, but
- regular **GEPs, PEPs, REPs** may have also **infinite eigenvalues**.
- **GEPs, PEPs, REPs** may be **singular** (BEPs are always regular) and to have, in addition to eigenvalues, **minimal indices**.
- **REPs** have **poles**. In modern applications, the poles are usually known (even chosen in approximating NEPs by REPs), but in other applications (Control), poles are not known and must be computed.
- We illustrate informally some of these concepts on matrix polynomials...

1. **BEP**: $(\lambda I_n - A)\, v = 0$
2. **GEP**: $(\lambda B - A)\, v = 0$
3. **PEP**: $(P_d \lambda^d + \cdots + P_1 \lambda + P_0) v = 0$
4. **REP**: $G(\lambda) v = 0$

- So far, we have only considered **finite eigenvalues**, but
- regular **GEPs, PEPs, REPs** may have also **infinite eigenvalues**.
- **GEPs, PEPs, REPs** may be **singular** (BEPs are always regular) and to have, in addition to eigenvalues, **minimal indices**.
- **REPs** have **poles**. In modern applications, the poles are usually known (even chosen in approximating NEPs by REPs), but in other applications (Control), poles are not known and must be computed.
- We illustrate informally some of these concepts on matrix polynomials...

GEPs-PEPs-REPs have more spectral "structural" data than BEPs

1. **BEP**: $(\lambda I_n - A)\, v = 0$
2. **GEP**: $(\lambda B - A)\, v = 0$
3. **PEP**: $(P_d \lambda^d + \cdots + P_1 \lambda + P_0) v = 0$
4. **REP**: $G(\lambda) v = 0$

- So far, we have only considered **finite eigenvalues**, but
- regular **GEPs, PEPs, REPs** may have also **infinite eigenvalues**.
- **GEPs, PEPs, REPs** may be **singular** (BEPs are always regular) and to have, in addition to eigenvalues, **minimal indices**.
- **REPs** have **poles**. In modern applications, the poles are usually known (even chosen in approximating NEPs by REPs), but in other applications (Control), poles are not known and must be computed.
- We illustrate informally some of these concepts on matrix polynomials...

1. **BEP**: $(\lambda I_n - A)\, v = 0$
2. **GEP**: $(\lambda B - A)\, v = 0$
3. **PEP**: $(P_d \lambda^d + \cdots + P_1 \lambda + P_0) v = 0$
4. **REP**: $G(\lambda) v = 0$

- So far, we have only considered **finite eigenvalues**, but
- regular **GEPs, PEPs, REPs** may have also **infinite eigenvalues**.
- **GEPs, PEPs, REPs** may be **singular** (BEPs are always regular) and to have, in addition to eigenvalues, **minimal indices**.
- **REPs** have **poles**. In modern applications, the poles are usually known (even chosen in approximating NEPs by REPs), but in other applications (Control), poles are not known and must be computed.
- We illustrate informally some of these concepts on matrix polynomials...

GEPs-PEPs-REPs have more spectral “structural” data than BEPs

1. **BEP**: $(\lambda I_n - A)\, v = 0$
2. **GEP**: $(\lambda B - A)\, v = 0$
3. **PEP**: $(P_d \lambda^d + \cdots + P_1 \lambda + P_0) v = 0$
4. **REP**: $G(\lambda) v = 0$

- So far, we have only considered **finite eigenvalues**, but
- regular **GEPs, PEPs, REPs** may have also **infinite eigenvalues**.
- **GEPs, PEPs, REPs** may be **singular** (BEPs are always regular) and to have, in addition to eigenvalues, **minimal indices**.
- **REPs** have **poles**. In modern applications, the poles are usually known (even chosen in approximating NEPs by REPs), but in other applications (Control), poles are not known and must be computed.
- We illustrate informally some of these concepts on matrix polynomials...

Infinite eigenvalues in PEPs

- Given the **regular ($\det P(\lambda) \neq 0$) polynomial matrix**

$$\boxed{P(\lambda) = P_d \lambda^d + \cdots + P_1 \lambda + P_0}, \qquad P_i \in \mathbb{C}^{n \times n},$$

  then the eigenvalues of the PEP $\boxed{P(\lambda_0)\, v = 0, \quad 0 \neq v \in \mathbb{C}^n}$ are the roots of the scalar polynomial $\boxed{\det P(\lambda)}$.

- Thus, $P(\lambda)$ has **at most dn finite eigenvalues** since

$$\det P(\lambda) = (\det P_d)\, \lambda^{dn} + \text{lower degree terms in } \lambda\,.$$

- If $\det P_d = 0$, then the number of finite eigenvalues of the PEP is degree $(\det P(\lambda))$ and it is said that

- the **PEP has $dn -$ degree $(\det P(\lambda))$ infinite eigenvalues**.

- Given the **regular ($\det P(\lambda) \not\equiv 0$) polynomial matrix**

$$\boxed{P(\lambda) = P_d \lambda^d + \cdots + P_1 \lambda + P_0}, \qquad P_i \in \mathbb{C}^{n\times n},$$

then the eigenvalues of the PEP $\boxed{P(\lambda_0)\, v = 0, \quad 0 \neq v \in \mathbb{C}^n}$ are the roots of the scalar polynomial $\boxed{\det P(\lambda)}$.

- Thus, $P(\lambda)$ has **at most dn finite eigenvalues** since

$$\det P(\lambda) = (\det P_d)\, \lambda^{dn} + \text{lower degree terms in } \lambda\,.$$

- If $\det P_d = 0$, then the number of finite eigenvalues of the PEP is degree $(\det P(\lambda))$ and it is said that
- the PEP has $dn -$ degree $(\det P(\lambda))$ **infinite eigenvalues**.

- Given the **regular ($\det P(\lambda) \not\equiv 0$) polynomial matrix**

$$\boxed{P(\lambda) = P_d \lambda^d + \cdots + P_1 \lambda + P_0}, \qquad P_i \in \mathbb{C}^{n\times n},$$

then the eigenvalues of the PEP $\boxed{P(\lambda_0)\, v = 0, \quad 0 \neq v \in \mathbb{C}^n}$ are the roots of the scalar polynomial $\boxed{\det P(\lambda)}$.

- Thus, $P(\lambda)$ has **at most dn finite eigenvalues** since

$$\det P(\lambda) = (\det P_d)\, \lambda^{dn} + \text{lower degree terms in } \lambda\,.$$

- If $\det P_d = 0$, then the number of finite eigenvalues of the PEP is degree $(\det P(\lambda))$ and it is said that
- the PEP has $dn -$ degree $(\det P(\lambda))$ **infinite eigenvalues**.

- Given the **regular ($\det P(\lambda) \not\equiv 0$) polynomial matrix**

$$\boxed{P(\lambda) = P_d \lambda^d + \cdots + P_1 \lambda + P_0}, \qquad P_i \in \mathbb{C}^{n \times n},$$

  then the eigenvalues of the PEP $\boxed{P(\lambda_0)\, v = 0, \quad 0 \neq v \in \mathbb{C}^n}$ are the roots of the scalar polynomial $\boxed{\det P(\lambda)}$.

- Thus, $P(\lambda)$ has **at most dn finite eigenvalues** since

$$\det P(\lambda) = (\det P_d)\, \lambda^{dn} + \text{lower degree terms in } \lambda\,.$$

- If $\det P_d = 0$, then the number of finite eigenvalues of the PEP is degree $(\det P(\lambda))$ and it is said that
- the PEP has dn – degree $(\det P(\lambda))$ **infinite eigenvalues**.

- Another way to define infinite eigenvalues of a PEP that can be generalized to non-regular or **singular polynomial matrices** is through **the reversal polynomial**.

- Given $P(\lambda) = P_d\lambda^d + \cdots + P_1\lambda + P_0$, its reversal is

$$\text{rev}P(\lambda) := \lambda^d P(\tfrac{1}{\lambda}) = P_0\lambda^d + \cdots + P_{d-1}\lambda + P_d \; .$$

- Then the **infinite eigenvalues** of $P(\lambda)$ correspond to the **zero eigenvalues** of rev$P(\lambda)$.

- Why the name **infinite eigenvalues**? A possible reason is that if a polynomial with infinite eigenvalues, i.e., with P_d singular, is perturbed a bit, then eigenvalues with very large absolute values often appears.

- Another way to define infinite eigenvalues of a PEP that can be generalized to non-regular or **singular polynomial matrices** is through **the reversal polynomial**.
- Given $\boxed{P(\lambda) = P_d\lambda^d + \cdots + P_1\lambda + P_0}$, its reversal is

$$\boxed{\text{rev}P(\lambda) := \lambda^d P(\tfrac{1}{\lambda}) = P_0\lambda^d + \cdots + P_{d-1}\lambda + P_d}\ .$$

- Then the **infinite eigenvalues** of $P(\lambda)$ correspond to the **zero eigenvalues** of rev$P(\lambda)$.
- Why the name **infinite eigenvalues**? A possible reason is that if a polynomial with infinite eigenvalues, i.e., with P_d singular, is perturbed a bit, then eigenvalues with very large absolute values often appears.

- Another way to define infinite eigenvalues of a PEP that can be generalized to non-regular or **singular polynomial matrices** is through **the reversal polynomial**.
- Given $P(\lambda) = P_d\lambda^d + \cdots + P_1\lambda + P_0$, its reversal is

$$\mathrm{rev}P(\lambda) := \lambda^d P(\tfrac{1}{\lambda}) = P_0\lambda^d + \cdots + P_{d-1}\lambda + P_d \quad .$$

- Then the **infinite eigenvalues** of $P(\lambda)$ correspond to the **zero eigenvalues** of rev$P(\lambda)$.
- Why the name **infinite eigenvalues**? A possible reason is that if a polynomial with infinite eigenvalues, i.e., with P_d singular, is perturbed a bit, then eigenvalues with very large absolute values often appears.

- Another way to define infinite eigenvalues of a PEP that can be generalized to non-regular or **singular polynomial matrices** is through **the reversal polynomial**.
- Given $\boxed{P(\lambda) = P_d\lambda^d + \cdots + P_1\lambda + P_0}$, its reversal is

$$\boxed{\text{rev}P(\lambda) := \lambda^d P(\tfrac{1}{\lambda}) = P_0\lambda^d + \cdots + P_{d-1}\lambda + P_d}\,.$$

- Then the **infinite eigenvalues** of $P(\lambda)$ correspond to the **zero eigenvalues** of rev$P(\lambda)$.
- Why the name **infinite eigenvalues**? A possible reason is that if a polynomial with infinite eigenvalues, i.e., with P_d singular, is perturbed a bit, then eigenvalues with very large absolute values often appears.

Let ϵ be a small parameter and consider the quadratic matrix polynomial

$$P(\lambda) = \begin{bmatrix} (\lambda-1)(\lambda-2) & 0 \\ 0 & \lambda(\epsilon\lambda-1) \end{bmatrix}$$
$$= \lambda^2 \begin{bmatrix} 1 & 0 \\ 0 & \epsilon \end{bmatrix} + \lambda \begin{bmatrix} -3 & 0 \\ 0 & -1 \end{bmatrix} + \begin{bmatrix} 2 & 0 \\ 0 & 0 \end{bmatrix}.$$

- If $\epsilon \neq 0$, then the eigenvalues are $\{1, 2, 0, 1/\epsilon\}$, (very large if $|\epsilon| \ll 1$).
- If $\epsilon = 0$, then the eigenvalues are $\{1, 2, 0, \infty\}$.

Example 1

Let ϵ be a small parameter and consider the quadratic matrix polynomial

$$P(\lambda) = \begin{bmatrix} (\lambda-1)(\lambda-2) & 0 \\ 0 & \lambda(\epsilon\lambda - 1) \end{bmatrix}$$
$$= \lambda^2 \begin{bmatrix} 1 & 0 \\ 0 & \epsilon \end{bmatrix} + \lambda \begin{bmatrix} -3 & 0 \\ 0 & -1 \end{bmatrix} + \begin{bmatrix} 2 & 0 \\ 0 & 0 \end{bmatrix}.$$

- If $\epsilon \neq 0$, then the eigenvalues are $\{1, 2, 0, 1/\epsilon\}$, (very large if $|\epsilon| \ll 1$).
- If $\epsilon = 0$, then the eigenvalues are $\{1, 2, 0, \infty\}$.

Example 1

Let ϵ be a small parameter and consider the quadratic matrix polynomial

$$\begin{aligned} P(\lambda) &= \begin{bmatrix} (\lambda-1)(\lambda-2) & 0 \\ 0 & \lambda(\epsilon\lambda-1) \end{bmatrix} \\ &= \lambda^2 \begin{bmatrix} 1 & 0 \\ 0 & \epsilon \end{bmatrix} + \lambda \begin{bmatrix} -3 & 0 \\ 0 & -1 \end{bmatrix} + \begin{bmatrix} 2 & 0 \\ 0 & 0 \end{bmatrix}. \end{aligned}$$

- If $\epsilon \neq 0$, then the eigenvalues are $\{1, 2, 0, 1/\epsilon\}$, (very large if $|\epsilon| \ll 1$).
- If $\epsilon = 0$, then the eigenvalues are $\{1, 2, 0, \infty\}$.

Example 1

Let ϵ be a small parameter and consider the quadratic matrix polynomial

$$P(\lambda) = \begin{bmatrix} (\lambda-1)(\lambda-2) & 0 \\ 0 & \lambda(\epsilon\lambda - 1) \end{bmatrix} = \lambda^2 \begin{bmatrix} 1 & 0 \\ 0 & \epsilon \end{bmatrix} + \lambda \begin{bmatrix} -3 & 0 \\ 0 & -1 \end{bmatrix} + \begin{bmatrix} 2 & 0 \\ 0 & 0 \end{bmatrix}.$$

- If $\epsilon \neq 0$, then the eigenvalues are $\{1, 2, 0, 1/\epsilon\}$, (very large if $|\epsilon| \ll 1$).
- If $\epsilon = 0$, then the eigenvalues are $\{1, 2, 0, \infty\}$.

- An additional high step of difficulty is that PEPs can be **singular**, which happens when

$$\boxed{P(\lambda) = P_d\lambda^d + \cdots + P_1\lambda + P_0}$$

is either **rectangular or square with** $\det P(\lambda) \equiv 0$, i.e., zero for all λ.

- **Singular PEPs also appear in applications**, in particular in Multivariable System Theory and Control Theory.

- An additional high step of difficulty is that PEPs can be **singular**, which happens when

$$P(\lambda) = P_d \lambda^d + \cdots + P_1 \lambda + P_0$$

is either **rectangular or square with** $\det P(\lambda) \equiv 0$, i.e., zero for all λ.

- **Singular PEPs also appear in applications**, in particular in Multivariable System Theory and Control Theory.

- In addition to eigenvalues, **singular matrix polynomials have** other "interesting numbers" attached to them called **minimal indices**.
- Recall that eigenvalues are related to the existence of nontrivial **null spaces**. For instance, $\mathcal{N}_r(\lambda_0 I_n - A) \neq \{0\}$ in BEPs.
- **Minimal indices** are related to the fact that a singular $m \times n$ matrix polynomial $P(\lambda)$ has non-trivial left and/or right null-spaces over the field $\mathbb{C}(\lambda)$ of rational functions:

$$\begin{aligned} \mathcal{N}_\ell(P) &:= \left\{ y(\lambda)^T \in \mathbb{F}(\lambda)^{1\times m} \, : \, y(\lambda)^T P(\lambda) \equiv 0^T \right\}, \\ \mathcal{N}_r(P) &:= \left\{ x(\lambda) \in \mathbb{F}(\lambda)^{n\times 1} \, : \, P(\lambda) x(\lambda) \equiv 0 \right\}, \end{aligned}$$

- which have bases consisting entirely of vector polynomials.
- Looking for polynomials bases with "minimal degree", in a certain sense, leads to the concepts of **minimal bases and indices**.

- In addition to eigenvalues, **singular matrix polynomials have** other "interesting numbers" attached to them called **minimal indices**.
- Recall that eigenvalues are related to the existence of nontrivial **null spaces**. For instance, $\mathcal{N}_r(\lambda_0 I_n - A) \neq \{0\}$ in BEPs.
- **Minimal indices** are related to the fact that a singular $m \times n$ matrix polynomial $P(\lambda)$ has non-trivial left and/or right null-spaces over the field $\mathbb{C}(\lambda)$ of rational functions:

$$
\begin{aligned}
\mathcal{N}_\ell(P) &:= \left\{ y(\lambda)^T \in \mathbb{F}(\lambda)^{1\times m} \,:\, y(\lambda)^T P(\lambda) \equiv 0^T \right\}, \\
\mathcal{N}_r(P) &:= \left\{ x(\lambda) \in \mathbb{F}(\lambda)^{n\times 1} \,:\, P(\lambda) x(\lambda) \equiv 0 \right\},
\end{aligned}
$$

- which have bases consisting entirely of vector polynomials.
- Looking for polynomials bases with "minimal degree", in a certain sense, leads to the concepts of **minimal bases and indices**.

- In addition to eigenvalues, **singular matrix polynomials have** other "interesting numbers" attached to them called **minimal indices**.
- Recall that eigenvalues are related to the existence of nontrivial **null spaces**. For instance, $\mathcal{N}_r(\lambda_0 I_n - A) \neq \{0\}$ in BEPs.
- **Minimal indices** are related to the fact that a singular $m \times n$ matrix polynomial $P(\lambda)$ has non-trivial left and/or right null-spaces over the field $\mathbb{C}(\lambda)$ of rational functions:

$$
\begin{aligned}
\mathcal{N}_\ell(P) &:= \left\{ y(\lambda)^T \in \mathbb{F}(\lambda)^{1\times m} \; : \; y(\lambda)^T P(\lambda) \equiv 0^T \right\}, \\
\mathcal{N}_r(P) &:= \left\{ x(\lambda) \in \mathbb{F}(\lambda)^{n\times 1} \; : \; P(\lambda) x(\lambda) \equiv 0 \right\},
\end{aligned}
$$

- which have bases consisting entirely of vector polynomials.
- Looking for polynomials bases with "minimal degree", in a certain sense, leads to the concepts of **minimal bases and indices**.

- In addition to eigenvalues, **singular matrix polynomials have** other "interesting numbers" attached to them called **minimal indices**.
- Recall that eigenvalues are related to the existence of nontrivial **null spaces**. For instance, $\mathcal{N}_r(\lambda_0 I_n - A) \neq \{0\}$ in BEPs.
- **Minimal indices** are related to the fact that a singular $m \times n$ matrix polynomial $P(\lambda)$ has non-trivial left and/or right null-spaces over the field $\mathbb{C}(\lambda)$ of rational functions:

$$\begin{aligned} \mathcal{N}_\ell(P) &:= \left\{ y(\lambda)^T \in \mathbb{F}(\lambda)^{1\times m} \, : \, y(\lambda)^T P(\lambda) \equiv 0^T \right\}, \\ \mathcal{N}_r(P) &:= \left\{ x(\lambda) \in \mathbb{F}(\lambda)^{n\times 1} \, : \, P(\lambda) x(\lambda) \equiv 0 \right\}, \end{aligned}$$

- which have bases consisting entirely of vector polynomials.
- Looking for polynomials bases with "minimal degree", in a certain sense, leads to the concepts of **minimal bases and indices**.

- In addition to eigenvalues, **singular matrix polynomials have** other "interesting numbers" attached to them called **minimal indices**.
- Recall that eigenvalues are related to the existence of nontrivial **null spaces**. For instance, $\mathcal{N}_r(\lambda_0 I_n - A) \neq \{0\}$ in BEPs.
- **Minimal indices** are related to the fact that a singular $m \times n$ matrix polynomial $P(\lambda)$ has non-trivial left and/or right null-spaces over the field $\mathbb{C}(\lambda)$ of rational functions:

$$\begin{aligned}
\mathcal{N}_\ell(P) &:= \left\{ y(\lambda)^T \in \mathbb{F}(\lambda)^{1\times m} \,:\, y(\lambda)^T P(\lambda) \equiv 0^T \right\}, \\
\mathcal{N}_r(P) &:= \left\{ x(\lambda) \in \mathbb{F}(\lambda)^{n\times 1} \,:\, P(\lambda) x(\lambda) \equiv 0 \right\},
\end{aligned}$$

- which have bases consisting entirely of vector polynomials.
- Looking for polynomials bases with "minimal degree", in a certain sense, leads to the concepts of **minimal bases and indices**.

Example III: right minimal bases and right minimal indices

$$P(\lambda) = \begin{bmatrix} \lambda & -\lambda^4 & 0 & 0 & 0 \\ 0 & 0 & 1 & -\lambda & 0 \\ 0 & 0 & 0 & 1 & -\lambda \end{bmatrix} \in \mathbb{C}[\lambda]^{3\times 5}$$

$$\mathcal{N}_r(P) = \text{Span}\{ \underbrace{\begin{bmatrix} \lambda^3 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}}_{u_1}, \underbrace{\begin{bmatrix} 0 \\ 0 \\ \lambda^2 \\ \lambda \\ 1 \end{bmatrix}}_{u_2} \} = \text{Span}\{ \underbrace{\begin{bmatrix} \lambda^3 \\ 1 \\ \lambda^3 \\ \lambda^2 \\ \lambda \end{bmatrix}}_{w_1}, \underbrace{\begin{bmatrix} \lambda^5 \\ \lambda^2 \\ \lambda^2 \\ \lambda \\ 1 \end{bmatrix}}_{w_2} \}$$

Sum of degrees of $\{u_1, u_2\} = 3 + 2 = 5$ (right minimal bases of $P(\lambda)$)

Sum of degrees of $\{w_1, w_2\} = 3 + 5 = 8$

Right minimal indices of $P(\lambda)$ = $\{2, 3\}$

Example III: right minimal bases and right minimal indices

$$P(\lambda) = \begin{bmatrix} \lambda & -\lambda^4 & 0 & 0 & 0 \\ 0 & 0 & 1 & -\lambda & 0 \\ 0 & 0 & 0 & 1 & -\lambda \end{bmatrix} \in \mathbb{C}[\lambda]^{3\times 5}$$

$$\mathcal{N}_r(P) = \text{Span}\{ \underbrace{\begin{bmatrix} \lambda^3 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}}_{u_1}, \underbrace{\begin{bmatrix} 0 \\ 0 \\ \lambda^2 \\ \lambda \\ 1 \end{bmatrix}}_{u_2} \} = \text{Span}\{ \underbrace{\begin{bmatrix} \lambda^3 \\ 1 \\ \lambda^3 \\ \lambda^2 \\ \lambda \end{bmatrix}}_{w_1}, \underbrace{\begin{bmatrix} \lambda^5 \\ \lambda^2 \\ \lambda^2 \\ \lambda \\ 1 \end{bmatrix}}_{w_2} \}$$

Sum of degrees of $\{u_1, u_2\} = 3 + 2 = 5$ (right minimal bases of $P(\lambda)$)

Sum of degrees of $\{w_1, w_2\} = 3 + 5 = 8$

Right minimal indices of $P(\lambda)$ = $\{2, 3\}$

Example III: right minimal bases and right minimal indices

$$P(\lambda) = \begin{bmatrix} \lambda & -\lambda^4 & 0 & 0 & 0 \\ 0 & 0 & 1 & -\lambda & 0 \\ 0 & 0 & 0 & 1 & -\lambda \end{bmatrix} \in \mathbb{C}[\lambda]^{3\times 5}$$

$$\mathcal{N}_r(P) = \text{Span}\{ \underbrace{\begin{bmatrix} \lambda^3 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}}_{u_1}, \underbrace{\begin{bmatrix} 0 \\ 0 \\ \lambda^2 \\ \lambda \\ 1 \end{bmatrix}}_{u_2} \} = \text{Span}\{ \underbrace{\begin{bmatrix} \lambda^3 \\ 1 \\ \lambda^3 \\ \lambda^2 \\ \lambda \end{bmatrix}}_{w_1}, \underbrace{\begin{bmatrix} \lambda^5 \\ \lambda^2 \\ \lambda^2 \\ \lambda \\ 1 \end{bmatrix}}_{w_2} \}$$

Sum of degrees of $\{u_1, u_2\} = 3 + 2 = 5$ (right minimal bases of $P(\lambda)$)

Sum of degrees of $\{w_1, w_2\} = 3 + 5 = 8$

Right minimal indices of $P(\lambda)$ = $\{2, 3\}$

Example III: right minimal bases and right minimal indices

$$P(\lambda) = \begin{bmatrix} \lambda & -\lambda^4 & 0 & 0 & 0 \\ 0 & 0 & 1 & -\lambda & 0 \\ 0 & 0 & 0 & 1 & -\lambda \end{bmatrix} \in \mathbb{C}[\lambda]^{3\times 5}$$

$$\mathcal{N}_r(P) = \text{Span}\{ \underbrace{\begin{bmatrix} \lambda^3 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}}_{u_1}, \underbrace{\begin{bmatrix} 0 \\ 0 \\ \lambda^2 \\ \lambda \\ 1 \end{bmatrix}}_{u_2} \} = \text{Span}\{ \underbrace{\begin{bmatrix} \lambda^3 \\ 1 \\ \lambda^3 \\ \lambda^2 \\ \lambda \end{bmatrix}}_{w_1}, \underbrace{\begin{bmatrix} \lambda^5 \\ \lambda^2 \\ \lambda^2 \\ \lambda \\ 1 \end{bmatrix}}_{w_2} \}$$

Sum of degrees of $\{u_1, u_2\} = 3 + 2 = 5$ (right minimal bases of $P(\lambda)$)

Sum of degrees of $\{w_1, w_2\} = 3 + 5 = 8$

Right minimal indices of $P(\lambda) = \{2, 3\}$

Example III: right minimal bases and right minimal indices

$$P(\lambda) = \begin{bmatrix} \lambda & -\lambda^4 & 0 & 0 & 0 \\ 0 & 0 & 1 & -\lambda & 0 \\ 0 & 0 & 0 & 1 & -\lambda \end{bmatrix} \in \mathbb{C}[\lambda]^{3\times 5}$$

$$\mathcal{N}_r(P) = \text{Span}\{ \underbrace{\begin{bmatrix} \lambda^3 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}}_{u_1}, \underbrace{\begin{bmatrix} 0 \\ 0 \\ \lambda^2 \\ \lambda \\ 1 \end{bmatrix}}_{u_2} \} = \text{Span}\{ \underbrace{\begin{bmatrix} \lambda^3 \\ 1 \\ \lambda^3 \\ \lambda^2 \\ \lambda \end{bmatrix}}_{w_1}, \underbrace{\begin{bmatrix} \lambda^5 \\ \lambda^2 \\ \lambda^2 \\ \lambda \\ 1 \end{bmatrix}}_{w_2} \}$$

Sum of degrees of $\{u_1, u_2\} = 3 + 2 = 5$ (right minimal bases of $P(\lambda)$)

Sum of degrees of $\{w_1, w_2\} = 3 + 5 = 8$

Right minimal indices of $P(\lambda)$ = $\{2, 3\}$

As a consequence of the previous discussion, we define:

Definition

The **complete "eigenstructure"** of a polynomial matrix $P(\lambda)$ is comprised of:

- its finite eigenvalues, together with their partial multiplicities,
- its infinite eigenvalue, together with its partial multiplicities,
- its right minimal indices, and
- its left minimal indices.

Remarks

- The partial multiplicities are rigorously defined through the Smith form of $P(\lambda)$ and for matrices they are just the sizes of the Jordan blocks associated to each eigenvalue.

As a consequence of the previous discussion, we define:

Definition

The **complete "eigenstructure"** of a polynomial matrix $P(\lambda)$ is comprised of:

- its finite eigenvalues, together with their partial multiplicities,
- its infinite eigenvalue, together with its partial multiplicities,
- its right minimal indices, and
- its left minimal indices.

Remarks

- The partial multiplicities are rigorously defined through the Smith form of $P(\lambda)$ and for matrices they are just the sizes of the Jordan blocks associated to each eigenvalue.

Analogously, we define:

Definition

The **complete “eigenstructure”** of a rational matrix $G(\lambda)$ is comprised of:

- its finite zeros and poles, together with their partial multiplicities,
- its infinite zeros and poles, together with its partial multiplicities,
- its right minimal indices, and
- its left minimal indices.

Remarks

- The partial multiplicities are rigorously defined through the Smith-McMillan form of $G(\lambda)$.
- The **eigenvalues** of $G(\lambda)$ are those zeros that are not poles.

Analogously, we define:

Definition

The **complete “eigenstructure”** of a rational matrix $G(\lambda)$ is comprised of:

- its finite zeros and poles, together with their partial multiplicities,
- its infinite zeros and poles, together with its partial multiplicities,
- its right minimal indices, and
- its left minimal indices.

Remarks

- The partial multiplicities are rigorously defined through the Smith-McMillan form of $G(\lambda)$.
- The **eigenvalues** of $G(\lambda)$ are those zeros that are not poles.

Outline

As said before, the most reliable methods for solving numerically PEPs are based on the concept of linearization.

Definition

- A **linear polynomial matrix (or matrix pencil)** $L(\lambda)$ is a **linearization** of $P(\lambda) = P_d \lambda^d + \cdots + P_1 \lambda + P_0$ if there exist **unimodular** polynomial matrices $U(\lambda), V(\lambda)$ such that

$$U(\lambda)\, L(\lambda)\, V(\lambda) = \begin{bmatrix} I_s & 0 \\ 0 & P(\lambda) \end{bmatrix}.$$

- $L(\lambda)$ is a **strong linearization** of $P(\lambda)$ if, in addition, $\mathrm{rev}\, L(\lambda)$ is a linearization for $\mathrm{rev}\, P(\lambda)$.

As said before, the most reliable methods for solving numerically PEPs are based on the concept of linearization.

Definition

- A **linear polynomial matrix (or matrix pencil)** $L(\lambda)$ is a **linearization** of $P(\lambda) = P_d\,\lambda^d + \cdots + P_1\lambda + P_0$ if there exist **unimodular** polynomial matrices $U(\lambda), V(\lambda)$ such that

$$U(\lambda)\,L(\lambda)\,V(\lambda) = \begin{bmatrix} I_s & 0 \\ 0 & P(\lambda) \end{bmatrix} .$$

- $L(\lambda)$ is a **strong linearization** of $P(\lambda)$ if, in addition, $\mathrm{rev}\,L(\lambda)$ is a linearization for $\mathrm{rev}\,P(\lambda)$.

Theorem

A matrix pencil $L(\lambda)$ is a linearization *of a polynomial matrix* $P(\lambda)$ **if and only if**

(1) $L(\lambda)$ *and* $P(\lambda)$ **have the same number of right minimal indices**.

(2) $L(\lambda)$ *and* $P(\lambda)$ **have the same number of left minimal indices**.

(3) $L(\lambda)$ *and* $P(\lambda)$ **have the same finite eigenvalues** *with the same partial multiplicities.*

$L(\lambda)$ **is a strong linearization** *of* $P(\lambda)$ **if and only if** *(1), (2), (3) and*

(4) $L(\lambda)$ *and* $P(\lambda)$ **have the same infinite eigenvalues** *with the same partial multiplicities.*

Theorem

A matrix pencil $L(\lambda)$ is a linearization *of a polynomial matrix $P(\lambda)$* **if and only if**

(1) $L(\lambda)$ *and* $P(\lambda)$ **have the same number of right minimal indices**.

(2) $L(\lambda)$ *and* $P(\lambda)$ **have the same number of left minimal indices**.

(3) $L(\lambda)$ *and* $P(\lambda)$ **have the same finite eigenvalues** *with the same partial multiplicities.*

$L(\lambda)$ **is a strong linearization** *of* $P(\lambda)$ **if and only if** *(1), (2), (3) and*

(4) $L(\lambda)$ *and* $P(\lambda)$ **have the same infinite eigenvalues** *with the same partial multiplicities.*

The most famous strong linearization (I)

The classical **Frobenius companion form** of the $m \times n$ matrix polynomial

$$P(\lambda) = P_d \lambda^d + \cdots + P_1 \lambda + P_0$$

is

$$C_1(\lambda) := \begin{bmatrix} \lambda P_d + P_{d-1} & P_{d-2} & \cdots & P_1 & P_0 \\ -I_n & \lambda I_n & & & \\ & \ddots & \ddots & & \\ & & \ddots & \lambda I_n & \\ & & & -I_n & \lambda I_n \end{bmatrix} \in \mathbb{C}[\lambda]^{(m+n(d-1))\times nd}$$

Additional property of $C_1(\lambda)$: Example of strong linearization whose right (resp. left) minimal indices allow us to recover the ones of the polynomial via addition of a constant.

The classical **Frobenius companion form** of the $m \times n$ matrix polynomial

$$P(\lambda) = P_d\lambda^d + \cdots + P_1\lambda + P_0$$

is

$$C_1(\lambda) := \begin{bmatrix} \lambda P_d + P_{d-1} & P_{d-2} & \cdots & P_1 & P_0 \\ -I_n & \lambda I_n & & & \\ & \ddots & \ddots & & \\ & & \ddots & \lambda I_n & \\ & & & -I_n & \lambda I_n \end{bmatrix} \in \mathbb{C}[\lambda]^{(m+n(d-1))\times nd}$$

Additional property of $C_1(\lambda)$: Example of strong linearization whose right (resp. left) minimal indices allow us to recover the ones of the polynomial via addition of a constant.

Theorem (recovery of eigenvectors from $C_1(\lambda)$)

Let $P(\lambda) = P_d\lambda^d + \cdots + P_1\lambda + P_0$ be a **regular** *matrix polynomial,* $\lambda_0 \in \mathbb{C}$ **be a finite eigenvalue of** $P(\lambda)$*, and $C_1(\lambda)$ be the Frobenius companion form of $P(\lambda)$. Then,* **any eigenvector** v **of** $C_1(\lambda)$ **associated to** λ_0 **has the form**

$$v = \begin{bmatrix} \lambda_0^{d-1}\, x \\ \vdots \\ \lambda_0\, x \\ x \end{bmatrix} = \begin{bmatrix} \lambda_0^{d-1} \\ \vdots \\ \lambda_0 \\ 1 \end{bmatrix} \otimes x$$

with x an eigenvector of $P(\lambda)$ associated to λ_0.

- $C_1(\lambda)$ is one (among many others) strong linearization of $P(\lambda)$ that allows us to recover without computational cost the eigenvectors of the polynomial from those of the linearization.

Theorem (recovery of eigenvectors from $C_1(\lambda)$)

Let $P(\lambda) = P_d\lambda^d + \cdots + P_1\lambda + P_0$ *be a* **regular** *matrix polynomial,* $\lambda_0 \in \mathbb{C}$ **be a finite eigenvalue of** $P(\lambda)$*, and* $C_1(\lambda)$ *be the Frobenius companion form of* $P(\lambda)$*. Then,* **any eigenvector** v **of** $C_1(\lambda)$ **associated to** λ_0 **has the form**

$$v = \begin{bmatrix} \lambda_0^{d-1}\, x \\ \vdots \\ \lambda_0\, x \\ x \end{bmatrix} = \begin{bmatrix} \lambda_0^{d-1} \\ \vdots \\ \lambda_0 \\ 1 \end{bmatrix} \otimes x$$

with x an eigenvector of $P(\lambda)$ associated to λ_0.

- $C_1(\lambda)$ is one (among many others) strong linearization of $P(\lambda)$ that allows us to recover without computational cost the eigenvectors of the polynomial from those of the linearization.

- Since 2006 (Mackey, Mackey, Mehl, Mehrmann), many "new" strong linearizations of matrix polynomials have been developed by many authors all around the world
- which also allow us to recover minimal indices and eigenvectors of PEPs without any computational cost.
- One relevant motivation for developing new classes of linearizations is to preserve structures appearing in applications, which is important for saving operations in algorithms and for preserving properties of the eigenvalues in floating point arithmetic.
- For instance, if $P(\lambda) = P_d\lambda^d + \cdots + P_1\lambda + P_0$ is Hermitian, i.e., it has Hermitian coefficients, **the Frobenius companion form is not!!**

$$C_1(\lambda) := \begin{bmatrix} \lambda P_d + P_{d-1} & P_{d-2} & \cdots & P_1 & P_0 \\ -I_n & \lambda I_n & & & \\ & \ddots & \ddots & & \\ & & \ddots & \lambda I_n & \\ & & & -I_n & \lambda I_n \end{bmatrix}$$

- Since 2006 (Mackey, Mackey, Mehl, Mehrmann), many "new" strong linearizations of matrix polynomials have been developed by many authors all around the world
- which also allow us to recover minimal indices and eigenvectors of PEPs without any computational cost.
- One relevant motivation for developing new classes of linearizations is to preserve structures appearing in applications, which is important for saving operations in algorithms and for preserving properties of the eigenvalues in floating point arithmetic.
- For instance, if $P(\lambda) = P_d\lambda^d + \cdots + P_1\lambda + P_0$ is Hermitian, i.e., it has Hermitian coefficients, **the Frobenius companion form is not!!**

$$C_1(\lambda) := \begin{bmatrix} \lambda P_d + P_{d-1} & P_{d-2} & \cdots & P_1 & P_0 \\ -I_n & \lambda I_n & & & \\ & \ddots & \ddots & & \\ & & \ddots & \lambda I_n & \\ & & & -I_n & \lambda I_n \end{bmatrix}$$

Parenthesis: There are many other strong linearizations of PEPs (I)

- Since 2006 (Mackey, Mackey, Mehl, Mehrmann), many "new" strong linearizations of matrix polynomials have been developed by many authors all around the world
- which also allow us to recover minimal indices and eigenvectors of PEPs without any computational cost.
- One relevant motivation for developing new classes of linearizations is to preserve structures appearing in applications, which is important for saving operations in algorithms and for preserving properties of the eigenvalues in floating point arithmetic.
- For instance, if $P(\lambda) = P_d\lambda^d + \cdots + P_1\lambda + P_0$ is Hermitian, i.e., it has Hermitian coefficients, **the Frobenius companion form is not!!**

$$C_1(\lambda) := \begin{bmatrix} \lambda P_d + P_{d-1} & P_{d-2} & \cdots & P_1 & P_0 \\ -I_n & \lambda I_n & & & \\ & \ddots & \ddots & & \\ & & \ddots & \lambda I_n & \\ & & & -I_n & \lambda I_n \end{bmatrix}$$

- Since 2006 (Mackey, Mackey, Mehl, Mehrmann), many "new" strong linearizations of matrix polynomials have been developed by many authors all around the world
- which also allow us to recover minimal indices and eigenvectors of PEPs without any computational cost.
- One relevant motivation for developing new classes of linearizations is to preserve structures appearing in applications, which is important for saving operations in algorithms and for preserving properties of the eigenvalues in floating point arithmetic.
- For instance, if $P(\lambda) = P_d\lambda^d + \cdots + P_1\lambda + P_0$ is Hermitian, i.e., it has Hermitian coefficients, **the Frobenius companion form is not!!**

$$C_1(\lambda) := \begin{bmatrix} \lambda P_d + P_{d-1} & P_{d-2} & \cdots & P_1 & P_0 \\ -I_n & \lambda I_n & & & \\ & \ddots & \ddots & & \\ & & \ddots & \lambda I_n & \\ & & & -I_n & \lambda I_n \end{bmatrix}$$

- but

$$\widetilde{L}(\lambda) = \begin{bmatrix} \lambda P_1 + P_0 & \lambda I_n & & & & & 0 \\ \lambda I_n & 0 & I_n & & & & \\ & I_n & \lambda P_3 + P_2 & \lambda I_n & & & \\ & & \lambda I_n & 0 & I_n & & \\ & & & I_n & \lambda P_5 + P_4 & \lambda I_n & \\ & & & & \lambda I_n & 0 & I_n \\ 0 & & & & & I_n & \lambda P_7 + P_6 \end{bmatrix},$$

is a **Hermitian strong linearization** of the $n \times n$ Hermitian matrix polynomial $P(\lambda) = P_7\lambda^7 + \cdots + P_1\lambda + P_0$ (Antoniou-Vologiannidis 2004; De Terán-D-Mackey 2010; Mackey-Mackey-Mehl-Mehrmann 2010).

- "Good" strong linearizations of a matrix polynomial $P(\lambda)$ are linear matrix polynomials (matrix pencils) that have the same eigenvalues as $P(\lambda)$ and that allow us to recover the eigenvectors when $P(\lambda)$ is regular, and the minimal indices when $P(\lambda)$ is singular.
- **They allow to solve numerically PEPs because there exist excellent algorithms for solving linear PEPs, i.e., GEPs.**
- The fundamental proposed approach

"linearization + linear eigenvalue algorithm on the linearization"

for solving numerically PEPs can be traced back at least to Van Dooren-De Wilde (1983) and Van Dooren's PhD Thesis (1979).

Linearizations transform PEPs into GEPs ($P(\lambda) \longrightarrow \lambda B - A$)

- "Good" strong linearizations of a matrix polynomial $P(\lambda)$ are linear matrix polynomials (matrix pencils) that have the same eigenvalues as $P(\lambda)$ and that allow us to recover the eigenvectors when $P(\lambda)$ is regular, and the minimal indices when $P(\lambda)$ is singular.
- **They allow to solve numerically PEPs because there exist excellent algorithms for solving linear PEPs, i.e., GEPs.**
- The fundamental proposed approach

"linearization + linear eigenvalue algorithm on the linearization"

for solving numerically PEPs can be traced back at least to Van Dooren-De Wilde (1983) and Van Dooren's PhD Thesis (1979).

- "Good" strong linearizations of a matrix polynomial $P(\lambda)$ are linear matrix polynomials (matrix pencils) that have the same eigenvalues as $P(\lambda)$ and that allow us to recover the eigenvectors when $P(\lambda)$ is regular, and the minimal indices when $P(\lambda)$ is singular.
- **They allow to solve numerically PEPs because there exist excellent algorithms for solving linear PEPs, i.e., GEPs.**
- The fundamental proposed approach

  "linearization + linear eigenvalue algorithm on the linearization"

  for solving numerically PEPs can be traced back at least to Van Dooren-De Wilde (1983) and Van Dooren's PhD Thesis (1979).

- **QZ algorithm for regular GEPs** (Moler-Stewart 1973).
  - QZ is implemented in MATLAB command `eig(A,B)` and cost $O(n^3)$.
- **Staircase or GUPTRI algorithm for singular GEPs** (Van Dooren 1979; Demmel-Kågström, 1993).
  - It computes eigenvalues and minimal indices and there are FORTRAN public implementations.
- **QZ and Staircase are both backward stable algorithms**.
- The command `polyeig` of MATLAB computes all the eigenvalues of a “not too large” regular PEP by applying QZ to the first Frobenius companion form of the matrix polynomial defining the PEP.

- **QZ algorithm for regular GEPs** (Moler-Stewart 1973).
  - QZ is implemented in MATLAB command `eig(A,B)` and cost $O(n^3)$.
- **Staircase or GUPTRI algorithm for singular GEPs** (Van Dooren 1979; Demmel-Kågström, 1993).
  - It computes eigenvalues and minimal indices and there are FORTRAN public implementations.
- **QZ and Staircase are both backward stable algorithms**.
- The command `polyeig` of MATLAB computes all the eigenvalues of a "not too large" regular PEP by applying QZ to the first Frobenius companion form of the matrix polynomial defining the PEP.

- **QZ algorithm for regular GEPs** (Moler-Stewart 1973).
  - QZ is implemented in MATLAB command `eig(A,B)` and cost $O(n^3)$.
- **Staircase or GUPTRI algorithm for singular GEPs** (Van Dooren 1979; Demmel-Kågström, 1993).
  - It computes eigenvalues and minimal indices and there are FORTRAN public implementations.
- **QZ and Staircase are both backward stable algorithms**.
- The command `polyeig` of MATLAB computes all the eigenvalues of a “not too large” regular PEP by applying QZ to the first Frobenius companion form of the matrix polynomial defining the PEP.

- **QZ algorithm for regular GEPs** (Moler-Stewart 1973).
  - QZ is implemented in MATLAB command `eig(A,B)` and cost $O(n^3)$.
- **Staircase or GUPTRI algorithm for singular GEPs** (Van Dooren 1979; Demmel-Kågström, 1993).
  - It computes eigenvalues and minimal indices and there are FORTRAN public implementations.
- **QZ and Staircase are both backward stable algorithms**.
- The command `polyeig` of MATLAB computes all the eigenvalues of a "not too large" regular PEP by applying QZ to the first Frobenius companion form of the matrix polynomial defining the PEP.

- **QZ algorithm for regular GEPs** (Moler-Stewart 1973).
  - QZ is implemented in MATLAB command `eig(A,B)` and cost $O(n^3)$.
- **Staircase or GUPTRI algorithm for singular GEPs** (Van Dooren 1979; Demmel-Kågström, 1993).
  - It computes eigenvalues and minimal indices and there are FORTRAN public implementations.
- **QZ and Staircase are both backward stable algorithms**.
- The command `polyeig` of MATLAB computes all the eigenvalues of a "not too large" regular PEP by applying QZ to the first Frobenius companion form of the matrix polynomial defining the PEP.

- **QZ algorithm for regular GEPs** (Moler-Stewart 1973).
  - QZ is implemented in MATLAB command `eig(A,B)` and cost $O(n^3)$.
- **Staircase or GUPTRI algorithm for singular GEPs** (Van Dooren 1979; Demmel-Kågström, 1993).
  - It computes eigenvalues and minimal indices and there are FORTRAN public implementations.
- **QZ and Staircase are both backward stable algorithms**.
- The command `polyeig` of MATLAB computes all the eigenvalues of a “not too large” regular PEP by applying QZ to the first Frobenius companion form of the matrix polynomial defining the PEP.

- **Many are based on linearizations** $\lambda B - A$ of the PEP
- and on **Krylov subspace methods on** $\lambda B - A$ (Arnoldi on $B^{-1}A$, Rational-Krylov with shifts on $(A - \theta_j B)^{-1}B$) for computing a few desired eigenvalues,
- **but the application of these Krylov methods is NOT direct**,
- **since this would be very expensive in terms of memory and orthogonalization costs**, because
- if $P(\lambda) = P_d\lambda^d + \cdots + P_1\lambda + P_0 \in \mathbb{C}[\lambda]^{n\times n}$ then its Frobenius companion form (and any other strong linearization) has size $nd \times nd$

$$C_1(\lambda) := \begin{bmatrix} \lambda P_d + P_{d-1} & P_{d-2} & \cdots & P_1 & P_0 \\ -I_n & \lambda I_n & & & \\ & \ddots & \ddots & & \\ & & \ddots & \lambda I_n & \\ & & & -I_n & \lambda I_n \end{bmatrix} \in \mathbb{C}[\lambda]^{nd\times nd}.$$

So, if n is very large, then nd is very very large.

- **Many are based on linearizations** $\lambda B - A$ of the PEP
- and on **Krylov subspace methods on** $\lambda B - A$ (Arnoldi on $B^{-1}A$, Rational-Krylov with shifts on $(A - \theta_j B)^{-1}B$) for computing a few desired eigenvalues,
- **but the application of these Krylov methods is NOT direct**,
- **since this would be very expensive in terms of memory and orthogonalization costs**, because
- if $P(\lambda) = P_d\lambda^d + \cdots + P_1\lambda + P_0 \in \mathbb{C}[\lambda]^{n\times n}$ then its Frobenius companion form (and any other strong linearization) has size $nd \times nd$

$$C_1(\lambda) := \begin{bmatrix} \lambda P_d + P_{d-1} & P_{d-2} & \cdots & P_1 & P_0 \\ -I_n & \lambda I_n & & & \\ & \ddots & \ddots & & \\ & & \ddots & \lambda I_n & \\ & & & -I_n & \lambda I_n \end{bmatrix} \in \mathbb{C}[\lambda]^{nd\times nd}.$$

So, if n is very large, then nd is very very large.

- **Many are based on linearizations** $\lambda B - A$ of the PEP
- and on **Krylov subspace methods on** $\lambda B - A$ (Arnoldi on $B^{-1}A$, Rational-Krylov with shifts on $(A - \theta_j B)^{-1}B$) for computing a few desired eigenvalues,
- **but the application of these Krylov methods is NOT direct**,
- **since this would be very expensive in terms of memory and orthogonalization costs**, because
- if $P(\lambda) = P_d\lambda^d + \cdots + P_1\lambda + P_0 \in \mathbb{C}[\lambda]^{n\times n}$ then its Frobenius companion form (and any other strong linearization) has size $nd \times nd$

$$C_1(\lambda) := \begin{bmatrix} \lambda P_d + P_{d-1} & P_{d-2} & \cdots & P_1 & P_0 \\ -I_n & \lambda I_n & & & \\ & \ddots & \ddots & & \\ & & \ddots & \lambda I_n & \\ & & & -I_n & \lambda I_n \end{bmatrix} \in \mathbb{C}[\lambda]^{nd\times nd}.$$

So, if n is very large, then nd is very very large.

- **Many are based on linearizations** $\lambda B - A$ of the PEP
- and on **Krylov subspace methods on** $\lambda B - A$ (Arnoldi on $B^{-1}A$, Rational-Krylov with shifts on $(A - \theta_j B)^{-1}B$) for computing a few desired eigenvalues,
- **but the application of these Krylov methods is NOT direct**,
- **since this would be very expensive in terms of memory and orthogonalization costs**, because
- if $P(\lambda) = P_d\lambda^d + \cdots + P_1\lambda + P_0 \in \mathbb{C}[\lambda]^{n\times n}$ then its Frobenius companion form (and any other strong linearization) has size $nd \times nd$

$$C_1(\lambda) := \begin{bmatrix} \lambda P_d + P_{d-1} & P_{d-2} & \cdots & P_1 & P_0 \\ -I_n & \lambda I_n & & & \\ & \ddots & \ddots & & \\ & & \ddots & \lambda I_n & \\ & & & -I_n & \lambda I_n \end{bmatrix} \in \mathbb{C}[\lambda]^{nd\times nd}.$$

So, if n is very large, then nd is very very large.

- **Many are based on linearizations** $\lambda B - A$ of the PEP
- and on **Krylov subspace methods on** $\lambda B - A$ (Arnoldi on $B^{-1}A$, Rational-Krylov with shifts on $(A - \theta_j B)^{-1} B$) for computing a few desired eigenvalues,
- **but the application of these Krylov methods is NOT direct**,
- **since this would be very expensive in terms of memory and orthogonalization costs**, because
- if $P(\lambda) = P_d \lambda^d + \cdots + P_1 \lambda + P_0 \in \mathbb{C}[\lambda]^{n \times n}$ then its Frobenius companion form (and any other strong linearization) has size $nd \times nd$

$$C_1(\lambda) := \begin{bmatrix} \lambda P_d + P_{d-1} & P_{d-2} & \cdots & P_1 & P_0 \\ -I_n & \lambda I_n & & & \\ & \ddots & \ddots & & \\ & & \ddots & \lambda I_n & \\ & & & -I_n & \lambda I_n \end{bmatrix} \in \mathbb{C}[\lambda]^{nd \times nd}.$$

So, if n is very large, then nd is very very large.

- Therefore, Krylov subspace methods for PEPs take advantage, **in a sophisticated way**, of the structure of the linearization and of the bases of their Krylov subspaces
- to obtain **memory** and orthogonalization costs of the same order of those of an $n \times n$ standard matrix problem (almost no influence of d).
- The most stable and efficient methods in this family are
  1. TOAR (Two level Orthogonal ARnoldi) for QEPs (Su-Bai-Lu, 2008 and 2016) based on $C_1(\lambda)$,
  2. CORK (COmpact Rational Krylov) for arbitrary PEPs (Van Beeumen-Meerbergen-Michiels, 2015) very general, it can use many linearizations and bases for expressing the PEP.
- **Available HPC software:** parallel implementations of TOAR for any degree (including symm. versions) in SLEPc (Roman, UPV, 2016).

- Therefore, Krylov subspace methods for PEPs take advantage, **in a sophisticated way**, of the structure of the linearization and of the bases of their Krylov subspaces
- to obtain **memory** and orthogonalization costs of the same order of those of an $n \times n$ standard matrix problem (almost no influence of d).
- The most stable and efficient methods in this family are
  1. TOAR (Two level Orthogonal ARnoldi) for QEPs (Su-Bai-Lu, 2008 and 2016) based on $C_1(\lambda)$,
  2. CORK (COmpact Rational Krylov) for arbitrary PEPs (Van Beeumen-Meerbergen-Michiels, 2015) very general, it can use many linearizations and bases for expressing the PEP.
- **Available HPC software:** parallel implementations of TOAR for any degree (including symm. versions) in SLEPc (Roman, UPV, 2016).

- Therefore, Krylov subspace methods for PEPs take advantage, **in a sophisticated way**, of the structure of the linearization and of the bases of their Krylov subspaces
- to obtain **memory** and orthogonalization costs of the same order of those of an $n \times n$ standard matrix problem (almost no influence of d).
- The most stable and efficient methods in this family are
  1. TOAR (Two level Orthogonal ARnoldi) for QEPs (Su-Bai-Lu, 2008 and 2016) based on $C_1(\lambda)$,
  2. CORK (COmpact Rational Krylov) for arbitrary PEPs (Van Beeumen-Meerbergen-Michiels, 2015) very general, it can use many linearizations and bases for expressing the PEP.
- **Available HPC software:** parallel implementations of TOAR for any degree (including symm. versions) in SLEPc (Roman, UPV, 2016).

- Therefore, Krylov subspace methods for PEPs take advantage, **in a sophisticated way**, of the structure of the linearization and of the bases of their Krylov subspaces
- to obtain **memory** and orthogonalization costs of the same order of those of an $n \times n$ standard matrix problem (almost no influence of d).
- The most stable and efficient methods in this family are
  1. TOAR (Two level Orthogonal ARnoldi) for QEPs (Su-Bai-Lu, 2008 and 2016) based on $C_1(\lambda)$,
  2. CORK (COmpact Rational Krylov) for arbitrary PEPs (Van Beeumen-Meerbergen-Michiels, 2015) very general, it can use many linearizations and bases for expressing the PEP.
- **Available HPC software:** parallel implementations of TOAR for any degree (including symm. versions) in SLEPc (Roman, UPV, 2016).

Most of the linearizations of polynomial matrices available in the literature are inside (or very closely connected to) the following class of pencils.

Definition (D, Lawrence, Pérez, Van Dooren, Numer. Math., 2018)

A matrix pencil

$$\mathcal{L}(\lambda) = \begin{bmatrix} M(\lambda) & K_2(\lambda)^T \\ K_1(\lambda) & 0 \end{bmatrix}$$

is a block minimal bases pencil (BMBP) if $K_1(\lambda)$ and $K_2(\lambda)$ are minimal bases. If, in addition, the row degrees of $K_1(\lambda)$ and $K_2(\lambda)$ are all one, and the row degrees of each of their dual minimal bases $N_1(\lambda)$ and $N_2(\lambda)$ are all equal, then $\mathcal{L}(\lambda)$ is a strong block minimal bases pencil (SBMBP).

Theorem (D, Lawrence, Pérez, Van Dooren, Numer. Math., 2018)

If $\mathcal{L}(\lambda)$ is a BMBP (resp. SBMBP), then it is a linearization (resp. strong linearization) of the matrix polynomial

$$Q(\lambda) = N_2(\lambda) M(\lambda) N_1(\lambda)^T.$$

Most of the linearizations of polynomial matrices available in the literature are inside (or very closely connected to) the following class of pencils.

Definition (D, Lawrence, Pérez, Van Dooren, Numer. Math., 2018)

A matrix pencil

$$\mathcal{L}(\lambda) = \begin{bmatrix} M(\lambda) & K_2(\lambda)^T \\ K_1(\lambda) & 0 \end{bmatrix}$$

is a block minimal bases pencil (BMBP) if $K_1(\lambda)$ and $K_2(\lambda)$ are minimal bases. If, in addition, the row degrees of $K_1(\lambda)$ and $K_2(\lambda)$ are all one, and the row degrees of each of their dual minimal bases $N_1(\lambda)$ and $N_2(\lambda)$ are all equal, then $\mathcal{L}(\lambda)$ is a strong block minimal bases pencil (SBMBP).

Theorem (D, Lawrence, Pérez, Van Dooren, Numer. Math., 2018)

If $\mathcal{L}(\lambda)$ is a BMBP (resp. SBMBP), then it is a linearization (resp. strong linearization) of the matrix polynomial

$$Q(\lambda) = N_2(\lambda) M(\lambda) N_1(\lambda)^T .$$

Most of the linearizations of polynomial matrices available in the literature are inside (or very closely connected to) the following class of pencils.

Definition (D, Lawrence, Pérez, Van Dooren, Numer. Math., 2018)

A matrix pencil

$$\mathcal{L}(\lambda) = \begin{bmatrix} M(\lambda) & K_2(\lambda)^T \\ K_1(\lambda) & 0 \end{bmatrix}$$

is a block minimal bases pencil (BMBP) if $K_1(\lambda)$ and $K_2(\lambda)$ are minimal bases. If, in addition, the row degrees of $K_1(\lambda)$ and $K_2(\lambda)$ are all one, and the row degrees of each of their dual minimal bases $N_1(\lambda)$ and $N_2(\lambda)$ are all equal, then $\mathcal{L}(\lambda)$ is a strong block minimal bases pencil (SBMBP).

Theorem (D, Lawrence, Pérez, Van Dooren, Numer. Math., 2018)

If $\mathcal{L}(\lambda)$ is a BMBP (resp. SBMBP), then it is a linearization (resp. strong linearization) of the matrix polynomial

$$Q(\lambda) = N_2(\lambda) M(\lambda) N_1(\lambda)^T .$$

- In the complex linear space of matrix pencils of size $m \times n$ with $m < n$ endowed with the Euclidean metric, the set of pencils that are minimal bases is open and dense,
- even more is the complement of a proper algebraic set.
- If $m = (n - m)\eta$ with η integer, then the set of pencils that are minimal bases with all their row degrees equal to one and with their dual minimal bases having all the row degrees equal to η is open and dense, even more is the complement of a proper algebraic set.

These definitions are not restrictive (Van Dooren, D, Lin. Alg., 2018)

- In the complex linear space of matrix pencils of size $m \times n$ with $m < n$ endowed with the Euclidean metric, the set of pencils that are minimal bases is open and dense,
- even more is the complement of a proper algebraic set.
- If $m = (n - m)\eta$ with η integer, then the set of pencils that are minimal bases with all their row degrees equal to one and with their dual minimal bases having all the row degrees equal to η is open and dense, even more is the complement of a proper algebraic set.

These definitions are not restrictive (Van Dooren, D, Lin. Alg., 2018)

- In the complex linear space of matrix pencils of size $m \times n$ with $m < n$ endowed with the Euclidean metric, the set of pencils that are minimal bases is open and dense,
- even more is the complement of a proper algebraic set.
- If $m = (n-m)\eta$ with η integer, then the set of pencils that are minimal bases with all their row degrees equal to one and with their dual minimal bases having all the row degrees equal to η is open and dense, even more is the complement of a proper algebraic set.

Two fundamental auxiliary matrix polynomials in the rest of the talk are the pair of dual minimal bases

$$L_k(\lambda) := \begin{bmatrix} -1 & \lambda & & & \\ & -1 & \lambda & & \\ & & \ddots & \ddots & \\ & & & -1 & \lambda \end{bmatrix} \in \mathbb{C}[\lambda]^{k\times(k+1)},$$

$$\Lambda_k(\lambda)^T := \begin{bmatrix} \lambda^k & \lambda^{k-1} & \cdots & \lambda & 1 \end{bmatrix} \in \mathbb{C}[\lambda]^{1\times(k+1)},$$

and their Kronecker products by identities

$$L_k(\lambda)\otimes I_n := \begin{bmatrix} -I_n & \lambda I_n & & & \\ & -I_n & \lambda I_n & & \\ & & \ddots & \ddots & \\ & & & -I_n & \lambda I_n \end{bmatrix} \in \mathbb{C}[\lambda]^{nk\times n(k+1)},$$

$$\Lambda_k(\lambda)^T\otimes I_n := \begin{bmatrix} \lambda^k I_n & \lambda^{k-1} I_n & \cdots & \lambda I_n & I_n \end{bmatrix} \in \mathbb{C}[\lambda]^{n\times n(k+1)},$$

which are also dual minimal bases.

Two fundamental auxiliary matrix polynomials in the rest of the talk are the pair of dual minimal bases

$$L_k(\lambda) := \begin{bmatrix} -1 & \lambda & & & \\ & -1 & \lambda & & \\ & & \ddots & \ddots & \\ & & & -1 & \lambda \end{bmatrix} \in \mathbb{C}[\lambda]^{k\times(k+1)},$$

$$\Lambda_k(\lambda)^T := \begin{bmatrix} \lambda^k & \lambda^{k-1} & \cdots & \lambda & 1 \end{bmatrix} \in \mathbb{C}[\lambda]^{1\times(k+1)},$$

and their Kronecker products by identities

$$L_k(\lambda)\otimes I_n := \begin{bmatrix} -I_n & \lambda I_n & & & \\ & -I_n & \lambda I_n & & \\ & & \ddots & \ddots & \\ & & & -I_n & \lambda I_n \end{bmatrix} \in \mathbb{C}[\lambda]^{nk\times n(k+1)},$$

$$\Lambda_k(\lambda)^T\otimes I_n := \begin{bmatrix} \lambda^k I_n & \lambda^{k-1} I_n & \cdots & \lambda I_n & I_n \end{bmatrix} \in \mathbb{C}[\lambda]^{n\times n(k+1)},$$

which are also dual minimal bases.

Examples of SBMBP: block-Kronecker pencils (II)

The Frobenius companion form of the $m \times n$ matrix polynomial $P(\lambda) = P_d\lambda^d + \cdots + P_1\lambda + P_0$ is

$$C_1(\lambda) := \begin{bmatrix} \lambda P_d + P_{d-1} & P_{d-2} & \cdots & P_1 & P_0 \\ -I_n & \lambda I_n & & & \\ & \ddots & \ddots & & \\ & & \ddots & \lambda I_n & \\ & & & -I_n & \lambda I_n \end{bmatrix},$$

and can be compactly written with the polynomials defined above as

$$C_1(\lambda) := \left[\begin{array}{ccccc} \lambda P_d + P_{d-1} & P_{d-2} & \cdots & P_1 & P_0 \\ \hline \multicolumn{5}{c}{L_{d-1}(\lambda) \otimes I_n} \end{array} \right].$$

The Frobenius companion form of the $m \times n$ matrix polynomial $P(\lambda) = P_d\lambda^d + \cdots + P_1\lambda + P_0$ is

$$C_1(\lambda) := \begin{bmatrix} \lambda P_d + P_{d-1} & P_{d-2} & \cdots & P_1 & P_0 \\ -I_n & \lambda I_n & & & \\ & \ddots & \ddots & & \\ & & \ddots & \lambda I_n & \\ & & & -I_n & \lambda I_n \end{bmatrix},$$

and can be compactly written with the polynomials defined above as

$$C_1(\lambda) := \left[\begin{array}{c} \lambda P_d + P_{d-1} \quad P_{d-2} \quad \cdots \quad P_1 \quad P_0 \\ \hline L_{d-1}(\lambda) \otimes I_n \end{array} \right].$$

Definition (D, Lawrence, Pérez, Van Dooren, Numer. Math., 2018)

Let $M(\lambda)$ be an arbitrary pencil. Then any pencil of the form

$$\mathcal{L}(\lambda) = \begin{array}{c c} \left[\begin{array}{c|c} M(\lambda) & L_\eta(\lambda)^T \otimes I_m \\ \hline L_\varepsilon(\lambda) \otimes I_n & 0 \end{array}\right] & \begin{array}{l} \}\, (\eta+1)m \\ \}\, \varepsilon n \end{array} \\ \begin{array}{cc} \underbrace{\hphantom{L_\varepsilon(\lambda) \otimes I_n}}_{(\varepsilon+1)n} & \underbrace{\hphantom{L_\eta(\lambda)^T \otimes I_m}}_{\eta m} \end{array} & \end{array},$$

is called a block Kronecker pencil (one-block row and column cases included).

Theorem (D, Lawrence, Pérez, Van Dooren, Numer. Math., 2018)

Any block Kronecker pencil $\mathcal{L}(\lambda)$ is a SBMBP and, so, a strong linearization of the matrix polynomial

$$Q(\lambda) := (\Lambda_\eta(\lambda)^T \otimes I_m) M(\lambda) (\Lambda_\varepsilon(\lambda) \otimes I_n) \in \mathbb{C}[\lambda]^{m \times n}.$$

Definition (D, Lawrence, Pérez, Van Dooren, Numer. Math., 2018)

Let $M(\lambda)$ be an arbitrary pencil. Then any pencil of the form

$$\mathcal{L}(\lambda) = \begin{array}{c} \left[\begin{array}{c|c} M(\lambda) & L_\eta(\lambda)^T \otimes I_m \\ \hline L_\varepsilon(\lambda) \otimes I_n & 0 \end{array} \right] \begin{array}{l} \}\, (\eta+1)m \\ \}\, \varepsilon n \end{array} \\ \begin{array}{cc} \underbrace{\hphantom{L_\varepsilon(\lambda) \otimes I_n}}_{(\varepsilon+1)n} & \underbrace{\hphantom{L_\eta(\lambda)^T \otimes I_m}}_{\eta m} \end{array} \end{array},$$

is called a block Kronecker pencil (one-block row and column cases included).

Theorem (D, Lawrence, Pérez, Van Dooren, Numer. Math., 2018)

Any block Kronecker pencil $\mathcal{L}(\lambda)$ is a SBMBP and, so, a strong linearization of the matrix polynomial

$$Q(\lambda) := (\Lambda_\eta(\lambda)^T \otimes I_m) M(\lambda) (\Lambda_\varepsilon(\lambda) \otimes I_n) \in \mathbb{C}[\lambda]^{m \times n}.$$

Examples of block Kronecker pencils (I)

$$P(\lambda) = \lambda^5 P_5 + \lambda^4 P_4 + \lambda^3 P_3 + \lambda^2 P_2 + \lambda P_1 + P_0 \in \mathbb{C}[\lambda]^{m\times n}$$

$$\left[\begin{array}{ccc|cc}
\lambda P_5 + P_4 & 0 & 0 & -I_m & 0 \\
0 & \lambda P_3 + P_2 & 0 & \lambda I_m & -I_m \\
0 & 0 & \lambda P_1 + P_0 & 0 & \lambda I_m \\
\hline
-I_n & \lambda I_n & 0 & 0 & 0 \\
0 & -I_n & \lambda I_n & 0 & 0
\end{array}\right]$$

Examples of block Kronecker pencils (II)

$$P(\lambda) = \lambda^5 P_5 + \lambda^4 P_4 + \lambda^3 P_3 + \lambda^2 P_2 + \lambda P_1 + P_0 \in \mathbb{C}[\lambda]^{m\times n}$$

$$\left[\begin{array}{ccc|cc} \lambda P_5 & \lambda P_4 & \lambda P_3 & -I_m & 0 \\ 0 & 0 & \lambda P_2 & \lambda I_m & -I_m \\ 0 & 0 & \lambda P_1 + P_0 & 0 & \lambda I_m \\ \hline -I_n & \lambda I_n & 0 & 0 & 0 \\ 0 & -I_n & \lambda I_n & 0 & 0 \end{array}\right]$$

Outline

First comments on linearizations of REPs

- **A difference between REPs and PEPs** is that there is no agreement yet on what is a **linearization** of a rational matrix.
- Many authors have developed "linearizations" of rational matrices, but they very rarely prove that properties analogous to those of linearizations of polynomial matrices are satisfied $\rightarrow$ MORE DIFFICULT PROBLEM.
- Pioneering works on linearizations of rational matrices:
  1. Van Dooren and Verghese in late 70s & early 80s construct pencils that have exactly the same eigenstructure as any given rational matrix. The constructions require numerical computations.
  2. Su and Bai, 2011, construct a Frobenius-like linearization from a representation of $G(\lambda)$ as polynomial + state-space realization.
- The definitions in this talk are those in Amparan, D, Marcaida, and Zaballa, *Strong linearizations of rational matrices*, MIMS Eprint (2016).
- Another approach for defining (non-strong) linearizations of rational matrices can be found in Alam-Behera, 2016.

First comments on linearizations of REPs

- **A difference between REPs and PEPs** is that there is no agreement yet on what is a **linearization** of a rational matrix.
- Many authors have developed "linearizations" of rational matrices, but they very rarely prove that properties analogous to those of linearizations of polynomial matrices are satisfied $\rightarrow$ MORE DIFFICULT PROBLEM.
- Pioneering works on linearizations of rational matrices:
  1. Van Dooren and Verghese in late 70s & early 80s construct pencils that have exactly the same eigenstructure as any given rational matrix. The constructions require numerical computations.
  2. Su and Bai, 2011, construct a Frobenius-like linearization from a representation of $G(\lambda)$ as polynomial + state-space realization.
- The definitions in this talk are those in Amparan, D, Marcaida, and Zaballa, *Strong linearizations of rational matrices*, MIMS Eprint (2016).
- Another approach for defining (non-strong) linearizations of rational matrices can be found in Alam-Behera, 2016.

First comments on linearizations of REPs

- **A difference between REPs and PEPs** is that there is no agreement yet on what is a **linearization** of a rational matrix.
- Many authors have developed "linearizations" of rational matrices, but they very rarely prove that properties analogous to those of linearizations of polynomial matrices are satisfied $\rightarrow$ MORE DIFFICULT PROBLEM.
- Pioneering works on linearizations of rational matrices:
  1. Van Dooren and Verghese in late 70s & early 80s construct pencils that have exactly the same eigenstructure as any given rational matrix. The constructions require numerical computations.
  2. Su and Bai, 2011, construct a Frobenius-like linearization from a representation of $G(\lambda)$ as polynomial + state-space realization.
- The definitions in this talk are those in Amparan, D, Marcaida, and Zaballa, *Strong linearizations of rational matrices*, MIMS Eprint (2016).
- Another approach for defining (non-strong) linearizations of rational matrices can be found in Alam-Behera, 2016.

- **A difference between REPs and PEPs** is that there is no agreement yet on what is a **linearization** of a rational matrix.
- Many authors have developed "linearizations" of rational matrices, but they very rarely prove that properties analogous to those of linearizations of polynomial matrices are satisfied $\rightarrow$ MORE DIFFICULT PROBLEM.
- Pioneering works on linearizations of rational matrices:
  1. Van Dooren and Verghese in late 70s & early 80s construct pencils that have exactly the same eigenstructure as any given rational matrix. The constructions require numerical computations.
  2. Su and Bai, 2011, construct a Frobenius-like linearization from a representation of $G(\lambda)$ as polynomial + state-space realization.
- The definitions in this talk are those in Amparan, D, Marcaida, and Zaballa, *Strong linearizations of rational matrices*, MIMS Eprint (2016).
- Another approach for defining (non-strong) linearizations of rational matrices can be found in Alam-Behera, 2016.

- **A difference between REPs and PEPs** is that there is no agreement yet on what is a **linearization** of a rational matrix.
- Many authors have developed "linearizations" of rational matrices, but they very rarely prove that properties analogous to those of linearizations of polynomial matrices are satisfied $\rightarrow$ MORE DIFFICULT PROBLEM.
- Pioneering works on linearizations of rational matrices:
  1. Van Dooren and Verghese in late 70s & early 80s construct pencils that have exactly the same eigenstructure as any given rational matrix. The constructions require numerical computations.
  2. Su and Bai, 2011, construct a Frobenius-like linearization from a representation of $G(\lambda)$ as polynomial + state-space realization.
- The definitions in this talk are those in Amparan, D, Marcaida, and Zaballa, *Strong linearizations of rational matrices*, MIMS Eprint (2016).
- Another approach for defining (non-strong) linearizations of rational matrices can be found in Alam-Behera, 2016.

- **A difference between REPs and PEPs** is that there is no agreement yet on what is a **linearization** of a rational matrix.
- Many authors have developed "linearizations" of rational matrices, but they very rarely prove that properties analogous to those of linearizations of polynomial matrices are satisfied $\rightarrow$ MORE DIFFICULT PROBLEM.
- Pioneering works on linearizations of rational matrices:
    1. Van Dooren and Verghese in late 70s & early 80s construct pencils that have exactly the same eigenstructure as any given rational matrix. The constructions require numerical computations.
    2. Su and Bai, 2011, construct a Frobenius-like linearization from a representation of $G(\lambda)$ as polynomial + state-space realization.
- The definitions in this talk are those in Amparan, D, Marcaida, and Zaballa, *Strong linearizations of rational matrices*, MIMS Eprint (2016).
- Another approach for defining (non-strong) linearizations of rational matrices can be found in Alam-Behera, 2016.

- **Any rational matrix $G(\lambda)$ can be uniquely expressed as**

$$G(\lambda)=D(\lambda)+G_{sp}(\lambda),$$

  where

  1. $D(\lambda)$ is a polynomial matrix (polynomial part of $G(\lambda)$), and
  2. the rational matrix $G_{sp}(\lambda)$ is **strictly proper** (strictly proper part of $G(\lambda)$), i.e., $\lim_{\lambda\to\infty} G_{sp}(\lambda)=0$.

- Let $d=\deg(D)$ if $D(\lambda)\neq 0$ and $d=0$ otherwise. We define the **reversal of $G(\lambda)$** as

$$\text{rev}\, G(\lambda)=\lambda^d G\left(\frac{1}{\lambda}\right).$$

- **Any rational matrix $G(\lambda)$ can be uniquely expressed as**

$$G(\lambda) = D(\lambda) + G_{sp}(\lambda),$$

  where

  1. $D(\lambda)$ is a polynomial matrix (polynomial part of $G(\lambda)$), and
  2. the rational matrix $G_{sp}(\lambda)$ is **strictly proper** (strictly proper part of $G(\lambda)$), i.e., $\lim_{\lambda\to\infty} G_{sp}(\lambda) = 0$.

- Let $d = \deg(D)$ if $D(\lambda) \neq 0$ and $d = 0$ otherwise. We define the **reversal of $G(\lambda)$** as

$$\operatorname{rev} G(\lambda) = \lambda^d \, G\left(\frac{1}{\lambda}\right).$$

Definition (Rosenbrock, 1970)

Let $G(\lambda) \in \mathbb{C}(\lambda)^{p\times m}$ be a rational matrix. The polynomial matrix

$$P(\lambda) = \begin{bmatrix} A(\lambda) & B(\lambda) \\ -C(\lambda) & D(\lambda) \end{bmatrix} \in \mathbb{C}[\lambda]^{(n+p)\times(n+m)}$$

is a **polynomial system matrix** of $G(\lambda)$ if

$$G(\lambda) = D(\lambda) + C(\lambda)A(\lambda)^{-1}B(\lambda).$$

If, in addition, $\begin{bmatrix} A(\lambda) \\ -C(\lambda) \end{bmatrix}$ and $\begin{bmatrix} A(\lambda) & B(\lambda) \end{bmatrix}$ do not have finite eigenvalues, then $P(\lambda)$ is a **minimal polynomial system matrix** of $G(\lambda)$.

Theorem (Rosenbrock, 1970)

Each rational matrix has infinitely many minimal polynomial system matrices and, in particular, has minimal polynomial system matrices in **space-state form**, *i.e.,*

$$A(\lambda) = \lambda I_n - A, \quad B(\lambda) = B, \quad C(\lambda) = C\,.$$

Minimal polynomial system matrices of rational matrices

Definition (Rosenbrock, 1970)

Let $G(\lambda) \in \mathbb{C}(\lambda)^{p\times m}$ be a rational matrix. The polynomial matrix

$$P(\lambda) = \begin{bmatrix} A(\lambda) & B(\lambda) \\ -C(\lambda) & D(\lambda) \end{bmatrix} \in \mathbb{C}[\lambda]^{(n+p)\times(n+m)}$$

is a **polynomial system matrix** of $G(\lambda)$ if

$$G(\lambda) = D(\lambda) + C(\lambda) A(\lambda)^{-1} B(\lambda).$$

If, in addition, $\begin{bmatrix} A(\lambda) \\ -C(\lambda) \end{bmatrix}$ and $\begin{bmatrix} A(\lambda) & B(\lambda) \end{bmatrix}$ do not have finite eigenvalues, then $P(\lambda)$ is a **minimal polynomial system matrix** of $G(\lambda)$.

Theorem (Rosenbrock, 1970)

Each rational matrix has infinitely many minimal polynomial system matrices and, in particular, has minimal polynomial system matrices in **space-state form**, *i.e.,*

$$A(\lambda) = \lambda I_n - A, \quad B(\lambda) = B, \quad C(\lambda) = C\,.$$

Theorem (Rosenbrock, 1970)

If

$$P(\lambda) = \begin{bmatrix} A(\lambda) & B(\lambda) \\ -C(\lambda) & D(\lambda) \end{bmatrix} \in \mathbb{C}[\lambda]^{(n+p)\times(n+m)}$$

is a **minimal** **polynomial system matrix** *of* $G(\lambda) = D(\lambda) + C(\lambda)A(\lambda)^{-1}B(\lambda)$, *then:*

1. *The finite eigenvalue structure of $P(\lambda)$ (including all types of multiplicities) coincides exactly with the finite zero structure of $G(\lambda)$.*
2. *The finite eigenvalue structure of $A(\lambda)$ (including all types of multiplicities) coincides exactly with the finite pole structure of $G(\lambda)$.*

Remark:

- Nothing can be guaranteed on the structure at infinity.

Minimal polynomial system matrices contain the whole finite structure

Theorem (Rosenbrock, 1970)

If

$$P(\lambda) = \begin{bmatrix} A(\lambda) & B(\lambda) \\ -C(\lambda) & D(\lambda) \end{bmatrix} \in \mathbb{C}[\lambda]^{(n+p)\times(n+m)}$$

is a **minimal polynomial system matrix** *of* $G(\lambda) = D(\lambda) + C(\lambda)A(\lambda)^{-1}B(\lambda)$, *then:*

1. *The finite eigenvalue structure of $P(\lambda)$ (including all types of multiplicities) coincides exactly with the finite zero structure of $G(\lambda)$.*
2. *The finite eigenvalue structure of $A(\lambda)$ (including all types of multiplicities) coincides exactly with the finite pole structure of $G(\lambda)$.*

Remark:

- Nothing can be guaranteed on the structure at infinity.

Definition (Amparan, D, Marcaida, Zaballa, 2016)

A **linearization of $G(\lambda) \in \mathbb{C}(\lambda)^{p\times m}$ is a matrix pencil**

$$L(\lambda) = \begin{bmatrix} A_1\lambda + A_0 & B_1\lambda + B_0 \\ -(C_1\lambda + C_0) & D_1\lambda + D_0 \end{bmatrix} \in \mathbb{C}[\lambda]^{(n+(p+s))\times(n+(m+s))}$$

such that:

(a) $L(\lambda)$ is a **minimal polynomial system matrix** of

$$\widehat{G}(\lambda) = (D_1\lambda + D_0) + (C_1\lambda + C_0)(A_1\lambda + A_0)^{-1}(B_1\lambda + B_0),$$

and

(b) there exist unimodular matrices $U_1(\lambda)$, $U_2(\lambda)$ such that

$$U_1(\lambda)\,\mathrm{diag}(G(\lambda), I_s)\,U_2(\lambda) = \widehat{G}(\lambda).$$

Definition (D, Marcaida, Quintana, Van Dooren, in progress, 2018)

A **linearization at infinity of** $G(\lambda) \in \mathbb{C}(\lambda)^{p\times m}$ **is a matrix pencil**

$$L(\lambda) = \begin{bmatrix} A_1\lambda + A_0 & B_1\lambda + B_0 \\ -(C_1\lambda + C_0) & D_1\lambda + D_0 \end{bmatrix} \in \mathbb{C}[\lambda]^{(n+(p+s))\times(n+(m+s))}$$

such that:

(a) if $n > 0$, then A_1 is invertible, and

(b) if $\widehat{G}(\lambda) = (D_1\lambda + D_0) + (C_1\lambda + C_0)(A_1\lambda + A_0)^{-1}(B_1\lambda + B_0)$, then there exist rational matrices invertible at $\lambda = 0$, $R_1(\lambda)$, $R_2(\lambda)$ (that is, $R_i(\lambda)$ does not have poles at $\lambda = 0$ and $\det R_i(0) \neq 0$) such that

$$R_1(\lambda)\ \mathrm{diag}(\mathrm{rev}\, G(\lambda), I_s)\, R_2(\lambda) = \mathrm{rev}\, \widehat{G}(\lambda).$$

Definition (Amparan, D, Marcaida, Zaballa, 2016)

A **strong linearization of $G(\lambda) \in \mathbb{C}(\lambda)^{p\times m}$ is a matrix pencil** $L(\lambda)$ such that

1. $L(\lambda)$ is a linearization of $G(\lambda)$, and
2. $L(\lambda)$ is a linearization at infinity of $G(\lambda)$.

Remark

If $G(\lambda)$ is a polynomial matrix, then linearizations and strong linearizations of $G(\lambda)$ according to the definitions above are linearizations and strong linearizations of $G(\lambda)$ according to the polynomial definition.

Definition (Amparan, D, Marcaida, Zaballa, 2016)

A **strong linearization of** $G(\lambda) \in \mathbb{C}(\lambda)^{p\times m}$ **is a matrix pencil** $L(\lambda)$ such that

1. $L(\lambda)$ is a linearization of $G(\lambda)$, and
2. $L(\lambda)$ is a linearization at infinity of $G(\lambda)$.

Remark

If $G(\lambda)$ is a polynomial matrix, then linearizations and strong linearizations of $G(\lambda)$ according to the definitions above are linearizations and strong linearizations of $G(\lambda)$ according to the polynomial definition.

Strong linearizations contain the whole zero/pole structure

Theorem (Amparan, D, Marcaida, Zaballa, 2016)

If

$$L(\lambda) = \begin{bmatrix} A_1\lambda + A_0 & B_1\lambda + B_0 \\ -(C_1\lambda + C_0) & D_1\lambda + D_0 \end{bmatrix} \in \mathbb{C}[\lambda]^{(n+(p+s))\times(n+(m+s))}$$

is a strong linearization of $G(\lambda) \in \mathbb{C}(\lambda)^{p\times m}$ then:

- *The finite eigenvalue structure of $L(\lambda)$ coincides exactly with the finite zero structure of $G(\lambda)$.*
- *The finite eigenvalue structure of $A_1\lambda + A_0$ coincides exactly with the finite pole structure of $G(\lambda)$.*
- *The infinite eigenvalue structure of $L(\lambda)$ allows us to recover exactly the infinite zero/pole structure of $G(\lambda)$ via a uniform shift.*
- *$L(\lambda)$ and $G(\lambda)$ have the same number of left and the same number of right minimal indices.*

- This is a consequence of the theorem in the next slide,
- which requires to know a **state-space realization of the strictly proper part** of the rational matrix.
- Such realizations can be obtained easily in many modern applications and, in any case, there are classical algorithms for computing them.
- Extensions to other scenarios are in progress D, Marcaida, Quintana, Van Dooren (2018) to cope with some pencils used by Güttel, Van Beeumen, Meerbergen, Michiels (2014), Lietaert, Pérez, Vandereycken, Meerbergen, (2018) in the approximation of NEPs with REPs.

There exist many strong linearizations of rational matrices

- This is a consequence of the theorem in the next slide,
- which requires to know a **state-space realization of the strictly proper part** of the rational matrix.
- Such realizations can be obtained easily in many modern applications and, in any case, there are classical algorithms for computing them.
- Extensions to other scenarios are in progress D, Marcaida, Quintana, Van Dooren (2018) to cope with some pencils used by Güttel, Van Beeumen, Meerbergen, Michiels (2014), Lietaert, Pérez, Vandereycken, Meerbergen, (2018) in the approximation of NEPs with REPs.

There exist many strong linearizations of rational matrices

- This is a consequence of the theorem in the next slide,
- which requires to know a **state-space realization of the strictly proper part** of the rational matrix.
- Such realizations can be obtained easily in many modern applications and, in any case, there are classical algorithms for computing them.
- Extensions to other scenarios are in progress D, Marcaida, Quintana, Van Dooren (2018) to cope with some pencils used by Güttel, Van Beeumen, Meerbergen, Michiels (2014), Lietaert, Pérez, Vandereycken, Meerbergen, (2018) in the approximation of NEPs with REPs.

- This is a consequence of the theorem in the next slide,
- which requires to know a **state-space realization of the strictly proper part** of the rational matrix.
- Such realizations can be obtained easily in many modern applications and, in any case, there are classical algorithms for computing them.
- Extensions to other scenarios are in progress D, Marcaida, Quintana, Van Dooren (2018) to cope with some pencils used by Güttel, Van Beeumen, Meerbergen, Michiels (2014), Lietaert, Pérez, Vandereycken, Meerbergen, (2018) in the approximation of NEPs with REPs.

Theorem (Amparan, D, Marcaida, Zaballa, 2016)

Let

$$\begin{bmatrix} M(\lambda) & K_2(\lambda)^T \\ K_1(\lambda) & 0 \end{bmatrix}$$

be a SBMBP and $N_1(\lambda), N_2(\lambda)$ *be minimal bases dual to* $K_1(\lambda), K_2(\lambda)$*. Consider for* $i = 1, 2$ *unimodular matrices*

$$U_i(\lambda) = \begin{bmatrix} K_i(\lambda) \\ \widehat{K}_i \end{bmatrix} \quad \text{and} \quad U_i(\lambda)^{-1} = \begin{bmatrix} \widehat{N}_i(\lambda)^T & N_i(\lambda)^T \end{bmatrix}$$

and a **linear minimal polynomial system matrix**

$$L(\lambda) = \left[\begin{array}{c|cc} (\lambda I_n - A) & B\widehat{K}_1 & 0 \\ \hline -\widehat{K}_2^T C & M(\lambda) & K_2(\lambda)^T \\ 0 & K_1(\lambda) & 0 \end{array}\right].$$

Then $L(\lambda)$ is a strong linearization of the rational matrix

$$G(\lambda) = \underbrace{N_2(\lambda)M(\lambda)N_1(\lambda)^T}_{\text{poly. part}} + \underbrace{C(\lambda I_n - A)^{-1}B}_{\text{strict. proper. part}}.$$

Example 1 of RSBMBL. Strong linearization based on Frobenius companion linearization for polynomials of Su & Bai (2011)

- Given rational matrix:

$$G(\lambda) = D_d\lambda^d + \cdots + D_1\lambda + D_0 + C(\lambda I_n - A)^{-1}B \in \mathbb{C}(\lambda)^{p\times m}.$$

- Strong linearization (originally introduced by Su & Bai (SIMAX, 2011) without minimal order requirement and without strong nature):

$$L(\lambda) = \left[\begin{array}{c|ccccc} \lambda I_n - A & 0 & 0 & \cdots & 0 & B \\ \hline -C & \lambda D_d + D_{d-1} & D_{d-2} & \cdots & D_1 & D_0 \\ 0 & -I_m & \lambda I_m & & & \\ \vdots & & \ddots & \ddots & & \\ \vdots & & & \ddots & \lambda I_m & \\ 0 & & & & -I_m & \lambda I_m \end{array}\right]$$

Example 1 of RSBMBL. Strong linearization based on Frobenius companion linearization for polynomials of Su & Bai (2011)

- Given rational matrix:

$$G(\lambda) = D_d\lambda^d + \cdots + D_1\lambda + D_0 + C(\lambda I_n - A)^{-1}B \in \mathbb{C}(\lambda)^{p\times m}.$$

- Strong linearization (originally introduced by Su & Bai (SIMAX, 2011) without minimal order requirement and without strong nature):

$$L(\lambda) = \left[\begin{array}{c|cccccc} \lambda I_n - A & 0 & 0 & \cdots & 0 & B \\ \hline -C & \lambda D_d + D_{d-1} & D_{d-2} & \cdots & D_1 & D_0 \\ 0 & -I_m & \lambda I_m & & & \\ \vdots & & \ddots & \ddots & & \\ \vdots & & & \ddots & \lambda I_m & \\ 0 & & & & -I_m & \lambda I_m \end{array}\right]$$

Example 2 of RSBMBL. Strong linearization based on another block Kronecker pencil (Amparan, D., Marcaida, Zaballa, 2016)

- Given rational matrix:

$$G(\lambda) = \lambda^5 D_5 + \lambda^4 D_4 + \lambda^3 D_3 + \lambda^2 D_2 + \lambda D_1 + D_0 + C(\lambda I_n - A)^{-1} B \in \mathbb{C}(\lambda)^{p\times m}$$

- Strong linearization:

$$L(\lambda) = \left[\begin{array}{c|ccc|cc} \lambda I_n - A & 0 & 0 & B & 0 & 0 \\ \hline 0 & \lambda P_5 + P_4 & 0 & 0 & -I_p & 0 \\ 0 & 0 & \lambda P_3 + P_2 & 0 & \lambda I_p & -I_p \\ -C & 0 & 0 & \lambda P_1 + P_0 & 0 & \lambda I_p \\ \hline 0 & -I_m & \lambda I_m & 0 & 0 & 0 \\ 0 & 0 & -I_m & \lambda I_m & 0 & 0 \end{array}\right]$$

- Given rational matrix:

$$G(\lambda) = \lambda^5 D_5 + \lambda^4 D_4 + \lambda^3 D_3 + \lambda^2 D_2 + \lambda D_1 + D_0 + C(\lambda I_n - A)^{-1} B \in \mathbb{C}(\lambda)^{p\times m}$$

- Strong linearization:

$$L(\lambda) = \left[\begin{array}{c|ccc|cc} \lambda I_n - A & 0 & 0 & B & 0 & 0 \\ \hline 0 & \lambda P_5 + P_4 & 0 & 0 & -I_p & 0 \\ 0 & 0 & \lambda P_3 + P_2 & 0 & \lambda I_p & -I_p \\ -C & 0 & 0 & \lambda P_1 + P_0 & 0 & \lambda I_p \\ \hline 0 & -I_m & \lambda I_m & 0 & 0 & 0 \\ 0 & 0 & -I_m & \lambda I_m & 0 & 0 \end{array}\right]$$

Step 1. Construct one of the previous (strong) linearizations $L(\lambda)$ of $G(\lambda)$.

Step 2. For computing the zeros (and minimal indices, if singular):

- **Step 2.1** Apply to $L(\lambda)$ the QZ algorithm for not too large regular problems.
- **Step 2.2** Apply to $L(\lambda)$ the Staircase algorithm for not too large singular problems.
- **Step 2.3** Apply to $L(\lambda)$ the structured rational Krylov algorithm R-CORK (D, González-Pizarro, 2018) for large-scale regular problems.

Step 3. If the poles are unknown and desired:

- **Step 3.1** Apply to the (1,1)-block of $L(\lambda)$ the QZ algorithm for not too large regular problems.
- **Step 3.2** Apply to the (1,1)-block of $L(\lambda)$ a rational Krylov algorithm for large-scale pencils.

Outline

- We consider a **general** $m \times n$ **polynomial matrix** of degree d

$$\boxed{P(\lambda) = P_d \lambda^d + \cdots + P_1 \lambda + P_0}, \qquad P_i \in \mathbb{C}^{m \times n},$$

- and we assume that its **complete eigenstructure**
- has been computed by applying a **backward stable algorithm** (QZ for regular, Staircase for singular)
- to a **strong linearization** $\mathcal{L}(\lambda)$ **in the wide class of block Kronecker linearizations** of $P(\lambda)$.

- We consider a **general** $m \times n$ **polynomial matrix** of degree d

$$\boxed{P(\lambda) = P_d\lambda^d + \cdots + P_1\lambda + P_0}, \qquad P_i \in \mathbb{C}^{m\times n},$$

- and we assume that its **complete eigenstructure**
- has been computed by applying a **backward stable algorithm** (QZ for regular, Staircase for singular)
- to a **strong linearization** $\mathcal{L}(\lambda)$ **in the wide class of block Kronecker linearizations** of $P(\lambda)$.

- We consider a **general** $m \times n$ **polynomial matrix** of degree d

$$\boxed{P(\lambda) = P_d\lambda^d + \cdots + P_1\lambda + P_0}, \qquad P_i \in \mathbb{C}^{m\times n},$$

- and we assume that its **complete eigenstructure**
- has been computed by applying a **backward stable algorithm** (QZ for regular, Staircase for singular)
- to a **strong linearization** $\mathcal{L}(\lambda)$ **in the wide class of block Kronecker linearizations** of $P(\lambda)$.

- We consider a **general** $m \times n$ **polynomial matrix** of degree d

$$\boxed{P(\lambda) = P_d \lambda^d + \cdots + P_1 \lambda + P_0}, \qquad P_i \in \mathbb{C}^{m \times n},$$

- and we assume that its **complete eigenstructure**
- has been computed by applying a **backward stable algorithm** (QZ for regular, Staircase for singular)
- to a **strong linearization** $\mathcal{L}(\lambda)$ **in the wide class of block Kronecker linearizations** of $P(\lambda)$.

- *The computed complete eigenstructure of $\mathcal{L}(\lambda)$ is the exact complete eigenstructure of a matrix pencil $\mathcal{L}(\lambda) + \Delta\mathcal{L}(\lambda)$ such that*

$$\frac{\|\Delta\mathcal{L}(\lambda)\|_F}{\|\mathcal{L}(\lambda)\|_F} = O(\mathbf{u}),$$

where $\mathbf{u} \approx 10^{-16}$ is the unit roundoff and

- $\|\cdot\|_F$ is the Frobenius norm, i.e., for any matrix polynomial

$$\|Q_k\lambda^k + \cdots + Q_1\lambda + Q_0\|_F = \sqrt{\|Q_k\|_F^2 + \cdots + \|Q_1\|_F^2 + \|Q_0\|_F^2}\,.$$

- But, does this imply that the computed complete eigenstructure of $P(\lambda)$ is the exact complete eigenstructure of a polynomial matrix of the same degree $P(\lambda) + \Delta P(\lambda)$ such that

$$\frac{\|\Delta P(\lambda)\|_F}{\|P(\lambda)\|_F} = O(\mathbf{u})\ ??$$

Backward stable algorithms on strong linearizations and question

- *The computed complete eigenstructure of $\mathcal{L}(\lambda)$ is the exact complete eigenstructure of a matrix pencil $\mathcal{L}(\lambda) + \Delta\mathcal{L}(\lambda)$ such that*

$$\frac{\|\Delta\mathcal{L}(\lambda)\|_F}{\|\mathcal{L}(\lambda)\|_F} = O(\mathbf{u}),$$

  where $\mathbf{u} \approx 10^{-16}$ is the unit roundoff and

- $\|\cdot\|_F$ is the Frobenius norm, i.e., for any matrix polynomial

$$\|Q_k\lambda^k + \cdots + Q_1\lambda + Q_0\|_F = \sqrt{\|Q_k\|_F^2 + \cdots + \|Q_1\|_F^2 + \|Q_0\|_F^2}\ .$$

- But, does this imply that the computed complete eigenstructure of $P(\lambda)$ is the exact complete eigenstructure of a polynomial matrix of the same degree $P(\lambda) + \Delta P(\lambda)$ such that

$$\frac{\|\Delta P(\lambda)\|_F}{\|P(\lambda)\|_F} = O(\mathbf{u})\ ??$$

- *The computed complete eigenstructure of $\mathcal{L}(\lambda)$ is the exact complete eigenstructure of a matrix pencil $\mathcal{L}(\lambda) + \Delta\mathcal{L}(\lambda)$ such that*

$$\frac{\|\Delta\mathcal{L}(\lambda)\|_F}{\|\mathcal{L}(\lambda)\|_F} = O(\mathbf{u}),$$

where $\mathbf{u} \approx 10^{-16}$ is the unit roundoff and

- $\|\cdot\|_F$ is the Frobenius norm, i.e., for any matrix polynomial

$$\|Q_k\lambda^k + \cdots + Q_1\lambda + Q_0\|_F = \sqrt{\|Q_k\|_F^2 + \cdots + \|Q_1\|_F^2 + \|Q_0\|_F^2}\ .$$

- But, does this imply that the computed complete eigenstructure of $P(\lambda)$ is the exact complete eigenstructure of a polynomial matrix of the same degree $P(\lambda) + \Delta P(\lambda)$ such that

$$\frac{\|\Delta P(\lambda)\|_F}{\|P(\lambda)\|_F} = O(\mathbf{u})\ ??$$

because block Kronecker linearizations are highly structured pencils and perturbations destroy the structure!!

Example: The Frobenius Companion Form

$$C_1(\lambda) := \begin{bmatrix} \lambda P_d + P_{d-1} & P_{d-2} & \cdots & P_1 & P_0 \\ -I_n & \lambda I_n & & & \\ & \ddots & \ddots & & \\ & & \ddots & \lambda I_n & \\ & & & -I_n & \lambda I_n \end{bmatrix}$$

$C_1(\lambda) + \Delta\mathcal{L}(\lambda) =$

$$\begin{bmatrix} \lambda(P_d + E_{11}) + (P_{d-1} + F_{11}) & \lambda E_{12} + P_{d-2} + F_{12} & \cdots & \lambda E_{1,d-1} + P_1 + F_{1,d-1} & \cdots \\ \lambda E_{21} - I_n + F_{21} & \lambda(I_n + E_{22}) + F_{22} & \lambda E_{23} + F_{23} & & \\ \lambda E_{31} + F_{31} & \lambda E_{32} + F_{32} & \ddots & & \\ \vdots & \vdots & \ddots & \lambda(I_n + E_{d-1,d-1}) + F_{d-1,d-1} & \\ \lambda E_{d1} + F_{d1} & \lambda E_{d2} + F_{d2} & & \lambda E_{d,d-1} + F_{d,d-1} - I_n & \cdots \end{bmatrix}$$

because block Kronecker linearizations are highly structured pencils and perturbations destroy the structure!!

Example: The Frobenius Companion Form

$$C_1(\lambda) := \begin{bmatrix} \lambda P_d + P_{d-1} & P_{d-2} & \cdots & P_1 & P_0 \\ -I_n & \lambda I_n & & & \\ & \ddots & \ddots & & \\ & & \ddots & \lambda I_n & \\ & & & -I_n & \lambda I_n \end{bmatrix}$$

$$C_1(\lambda) + \Delta\mathcal{L}(\lambda) =$$

$$\begin{bmatrix} \lambda(P_d + E_{11}) + (P_{d-1} + F_{11}) & \lambda E_{12} + P_{d-2} + F_{12} & \cdots & \lambda E_{1,d-1} + P_1 + F_{1,d-1} & \cdots \\ \lambda E_{21} - I_n + F_{21} & \lambda(I_n + E_{22}) + F_{22} & \lambda E_{23} + F_{23} & & \\ \lambda E_{31} + F_{31} & \lambda E_{32} + F_{32} & \ddots & & \\ \vdots & \vdots & \ddots & \lambda(I_n + E_{d-1,d-1}) + F_{d-1,d-1} & \\ \lambda E_{d1} + F_{d1} & \lambda E_{d2} + F_{d2} & & \lambda E_{d,d-1} + F_{d,d-1} - I_n & \cdots \end{bmatrix}$$

The matrix perturbation problems to be solved

- **Problem 1:** To establish conditions on $\|\Delta\mathcal{L}(\lambda)\|_F$ such that $\mathcal{L}(\lambda) + \Delta\mathcal{L}(\lambda)$ is a strong linearization for some polynomial matrix $P(\lambda) + \Delta P(\lambda)$ of degree d.
- **Problem 2:** To prove a perturbation bound

$$\frac{\|\Delta P(\lambda)\|_F}{\|P(\lambda)\|_F} \leq C_{P,\mathcal{L}} \frac{\|\Delta\mathcal{L}(\lambda)\|_F}{\|\mathcal{L}(\lambda)\|_F},$$

with $C_{P,\mathcal{L}}$ a number depending on $P(\lambda)$ and $\mathcal{L}(\lambda)$.
- For those $P(\lambda)$ and $\mathcal{L}(\lambda)$ s.t. $C_{P,\mathcal{L}}$ is moderate, to use global backward stable algorithms on $\mathcal{L}(\lambda)$ gives global backward stability for $P(\lambda)$.

- **Problem 1:** To establish conditions on $\|\Delta\mathcal{L}(\lambda)\|_F$ such that $\mathcal{L}(\lambda)+\Delta\mathcal{L}(\lambda)$ is a strong linearization for some polynomial matrix $P(\lambda)+\Delta P(\lambda)$ of degree d.
- **Problem 2:** To prove a perturbation bound

$$\frac{\|\Delta P(\lambda)\|_F}{\|P(\lambda)\|_F} \leq C_{P,\mathcal{L}} \frac{\|\Delta\mathcal{L}(\lambda)\|_F}{\|\mathcal{L}(\lambda)\|_F},$$

with $C_{P,\mathcal{L}}$ a number depending on $P(\lambda)$ and $\mathcal{L}(\lambda)$.
- For those $P(\lambda)$ and $\mathcal{L}(\lambda)$ s.t. $C_{P,\mathcal{L}}$ is moderate, to use global backward stable algorithms on $\mathcal{L}(\lambda)$ gives global backward stability for $P(\lambda)$.

- **Problem 1:** To establish conditions on $\|\Delta\mathcal{L}(\lambda)\|_F$ such that $\mathcal{L}(\lambda) + \Delta\mathcal{L}(\lambda)$ is a strong linearization for some polynomial matrix $P(\lambda) + \Delta P(\lambda)$ of degree d.
- **Problem 2:** To prove a perturbation bound

$$\frac{\|\Delta P(\lambda)\|_F}{\|P(\lambda)\|_F} \leq C_{P,\mathcal{L}} \frac{\|\Delta\mathcal{L}(\lambda)\|_F}{\|\mathcal{L}(\lambda)\|_F},$$

  with $C_{P,\mathcal{L}}$ a number depending on $P(\lambda)$ and $\mathcal{L}(\lambda)$.
- For those $P(\lambda)$ and $\mathcal{L}(\lambda)$ s.t. $C_{P,\mathcal{L}}$ is moderate, to use global backward stable algorithms on $\mathcal{L}(\lambda)$ gives global backward stability for $P(\lambda)$.

1. There are just a few previous analysis of this type:
   - Van Dooren & De Wilde (LAA 1983).
   - Edelman & Murakami (Math. Comp. 1995).
   - Lawrence & Corless (SIMAX 2015).
   - Lawrence & Van Barel & Van Dooren (SIMAX 2016).
   - Noferini & Pérez (Math. Comp., 2017).
2. **Our analysis improve considerably these analyses**, because
3. previous analyses are only valid to first order for infinitesimal perturbations, are very particular since are only valid for classical Frobenius linearizations or closely connected linearizations, and often are only valid for regular polynomial matrices.

Previous works on this type of backward error analyses

1. There are just a few previous analysis of this type:
   - Van Dooren & De Wilde (LAA 1983).
   - Edelman & Murakami (Math. Comp. 1995).
   - Lawrence & Corless (SIMAX 2015).
   - Lawrence & Van Barel & Van Dooren (SIMAX 2016).
   - Noferini & Pérez (Math. Comp., 2017).
2. **Our analysis improve considerably these analyses**, because
3. previous analyses are only valid to first order for infinitesimal perturbations, are very particular since are only valid for classical Frobenius linearizations or closely connected linearizations, and often are only valid for regular polynomial matrices.

1. There are just a few previous analysis of this type:
   - Van Dooren & De Wilde (LAA 1983).
   - Edelman & Murakami (Math. Comp. 1995).
   - Lawrence & Corless (SIMAX 2015).
   - Lawrence & Van Barel & Van Dooren (SIMAX 2016).
   - Noferini & Pérez (Math. Comp., 2017).
2. **Our analysis improve considerably these analyses**, because
3. previous analyses are only valid to first order for infinitesimal perturbations, are very particular since are only valid for classical Frobenius linearizations or closely connected linearizations, and often are only valid for regular polynomial matrices.

Theorem (D, Lawrence, Pérez, Van Dooren, Numer. Math., 2018)

Let $\mathcal{L}(\lambda)$ be a block Kronecker pencil for $P(\lambda) = \sum_{i=0}^{d} P_i \lambda^i \in \mathbb{C}[\lambda]^{m\times n}$, i.e.,

$$\mathcal{L}(\lambda) = \left[\begin{array}{c|c} M(\lambda) & L_\eta(\lambda)^T \otimes I_m \\ \hline L_\varepsilon(\lambda) \otimes I_n & 0 \end{array}\right].$$

If $\Delta\mathcal{L}(\lambda)$ is any pencil with the same size as $\mathcal{L}(\lambda)$ and such that

$$\|\Delta\mathcal{L}(\lambda)\|_F < \frac{(\sqrt{2}-1)^2}{d^{5/2}} \frac{1}{1+\|M(\lambda)\|_F},$$

then $\mathcal{L}(\lambda)+\Delta\mathcal{L}(\lambda)$ is a strong linearization of a polynomial matrix $P(\lambda)+\Delta P(\lambda)$ with grade d and such that

$$\frac{\|\Delta P(\lambda)\|_F}{\|P(\lambda)\|_F} \leq 14\, d^{5/2} \frac{\|\mathcal{L}(\lambda)\|_F}{\|P(\lambda)\|_F} \left(1+\|M(\lambda)\|_F+\|M(\lambda)\|_F^2\right) \frac{\|\Delta\mathcal{L}(\lambda)\|_F}{\|\mathcal{L}(\lambda)\|_F}.$$

Discussion of the perturbation bounds for block Kronecker pencils

$$\mathcal{L}(\lambda) = \left[\begin{array}{c|c} M(\lambda) & L_\eta(\lambda)^T \otimes I_m \\ \hline L_\varepsilon(\lambda) \otimes I_n & 0 \end{array} \right].$$

$$\frac{\|\Delta P(\lambda)\|_F}{\|P(\lambda)\|_F} \leq \underbrace{14\, d^{5/2} \frac{\|\mathcal{L}(\lambda)\|_F}{\|P(\lambda)\|_F} (1 + \|M(\lambda)\|_F + \|M(\lambda)\|_F^2)}_{C_{P,\mathcal{L}}} \frac{\|\Delta \mathcal{L}(\lambda)\|_F}{\|\mathcal{L}(\lambda)\|_F}.$$

- It can be proved that if $\|P(\lambda)\|_F \ll 1$ or $\|P(\lambda)\|_F \gg 1$, then $C_{P,\mathcal{L}} \gg 1$,
- and that, if $\|M(\lambda)\|_F \gg 1$, then $C_{P,\mathcal{L}} \gg 1$.
- Therefore, for getting "backward stability" from Block Kronecker linearizations, one needs to normalize the matrix poly $\|P(\lambda)\|_F = 1$ and to use pencils such that $\|M(\lambda)\|_F \approx \|P(\lambda)\|_F$, then

$$\frac{\|\Delta P(\lambda)\|_F}{\|P(\lambda)\|_F} \lesssim d^3 \sqrt{m+n} \frac{\|\Delta \mathcal{L}(\lambda)\|_F}{\|\mathcal{L}(\lambda)\|_F}.$$

For Fiedler, Frobenius, etc linearizations $\|M(\lambda)\|_F = \|P(\lambda)\|_F$.

Discussion of the perturbation bounds for block Kronecker pencils

$$\mathcal{L}(\lambda) = \left[\begin{array}{c|c} M(\lambda) & L_\eta(\lambda)^T \otimes I_m \\ \hline L_\varepsilon(\lambda) \otimes I_n & 0 \end{array}\right].$$

$$\frac{\|\Delta P(\lambda)\|_F}{\|P(\lambda)\|_F} \leq \underbrace{14\, d^{5/2} \frac{\|\mathcal{L}(\lambda)\|_F}{\|P(\lambda)\|_F} \left(1 + \|M(\lambda)\|_F + \|M(\lambda)\|_F^2\right)}_{C_{P,\mathcal{L}}} \frac{\|\Delta \mathcal{L}(\lambda)\|_F}{\|\mathcal{L}(\lambda)\|_F}.$$

- It can be proved that if $\|P(\lambda)\|_F \ll 1$ or $\|P(\lambda)\|_F \gg 1$, then $C_{P,\mathcal{L}} \gg 1$,
- and that, if $\|M(\lambda)\|_F \gg 1$, then $C_{P,\mathcal{L}} \gg 1$.
- Therefore, for getting “backward stability” from Block Kronecker linearizations, one needs to normalize the matrix poly $\|P(\lambda)\|_F = 1$ and to use pencils such that $\|M(\lambda)\|_F \approx \|P(\lambda)\|_F$, then

$$\frac{\|\Delta P(\lambda)\|_F}{\|P(\lambda)\|_F} \lesssim d^3 \sqrt{m+n}\, \frac{\|\Delta \mathcal{L}(\lambda)\|_F}{\|\mathcal{L}(\lambda)\|_F}.$$

For Fiedler, Frobenius, etc linearizations $\|M(\lambda)\|_F = \|P(\lambda)\|_F$.

$$\mathcal{L}(\lambda) = \left[\begin{array}{c|c} M(\lambda) & L_\eta(\lambda)^T \otimes I_m \\ \hline L_\varepsilon(\lambda) \otimes I_n & 0 \end{array} \right].$$

$$\frac{\|\Delta P(\lambda)\|_F}{\|P(\lambda)\|_F} \leq \underbrace{14\, d^{5/2} \frac{\|\mathcal{L}(\lambda)\|_F}{\|P(\lambda)\|_F} \left(1 + \|M(\lambda)\|_F + \|M(\lambda)\|_F^2\right)}_{C_{P,\mathcal{L}}} \frac{\|\Delta \mathcal{L}(\lambda)\|_F}{\|\mathcal{L}(\lambda)\|_F}.$$

- It can be proved that if $\|P(\lambda)\|_F \ll 1$ or $\|P(\lambda)\|_F \gg 1$, then $C_{P,\mathcal{L}} \gg 1$,
- and that, if $\|M(\lambda)\|_F \gg 1$, then $C_{P,\mathcal{L}} \gg 1$.
- Therefore, for getting "backward stability" from Block Kronecker linearizations, one needs to normalize the matrix poly $\|P(\lambda)\|_F = 1$ and to use pencils such that $\|M(\lambda)\|_F \approx \|P(\lambda)\|_F$, then

$$\frac{\|\Delta P(\lambda)\|_F}{\|P(\lambda)\|_F} \lesssim d^3 \sqrt{m+n}\, \frac{\|\Delta \mathcal{L}(\lambda)\|_F}{\|\mathcal{L}(\lambda)\|_F}.$$

For Fiedler, Frobenius, etc linearizations $\|M(\lambda)\|_F = \|P(\lambda)\|_F$.

Discussion of the perturbation bounds for block Kronecker pencils

$$\mathcal{L}(\lambda) = \left[\begin{array}{c|c} M(\lambda) & L_\eta(\lambda)^T \otimes I_m \\ \hline L_\varepsilon(\lambda) \otimes I_n & 0 \end{array}\right].$$

$$\frac{\|\Delta P(\lambda)\|_F}{\|P(\lambda)\|_F} \leq \underbrace{14\, d^{5/2} \frac{\|\mathcal{L}(\lambda)\|_F}{\|P(\lambda)\|_F} \left(1 + \|M(\lambda)\|_F + \|M(\lambda)\|_F^2\right)}_{C_{P,\mathcal{L}}} \frac{\|\Delta \mathcal{L}(\lambda)\|_F}{\|\mathcal{L}(\lambda)\|_F}.$$

- It can be proved that if $\|P(\lambda)\|_F \ll 1$ or $\|P(\lambda)\|_F \gg 1$, then $C_{P,\mathcal{L}} \gg 1$,
- and that, if $\|M(\lambda)\|_F \gg 1$, then $C_{P,\mathcal{L}} \gg 1$.
- Therefore, for getting "backward stability" from Block Kronecker linearizations, one needs to normalize the matrix poly $\|P(\lambda)\|_F = 1$ and to use pencils such that $\|M(\lambda)\|_F \approx \|P(\lambda)\|_F$, then

$$\frac{\|\Delta P(\lambda)\|_F}{\|P(\lambda)\|_F} \lesssim d^3 \sqrt{m+n}\, \frac{\|\Delta \mathcal{L}(\lambda)\|_F}{\|\mathcal{L}(\lambda)\|_F}.$$

For Fiedler, Frobenius, etc linearizations $\|M(\lambda)\|_F = \|P(\lambda)\|_F$.

$$\mathcal{L}(\lambda) = \left[\begin{array}{c|c} M(\lambda) & L_\eta(\lambda)^T \otimes I_m \\ \hline L_\varepsilon(\lambda) \otimes I_n & 0 \end{array}\right].$$

$$\frac{\|\Delta P(\lambda)\|_F}{\|P(\lambda)\|_F} \leq \underbrace{14\, d^{5/2} \frac{\|\mathcal{L}(\lambda)\|_F}{\|P(\lambda)\|_F} (1 + \|M(\lambda)\|_F + \|M(\lambda)\|_F^2)}_{C_{P,\mathcal{L}}} \frac{\|\Delta\mathcal{L}(\lambda)\|_F}{\|\mathcal{L}(\lambda)\|_F}.$$

- It can be proved that if $\|P(\lambda)\|_F \ll 1$ or $\|P(\lambda)\|_F \gg 1$, then $C_{P,\mathcal{L}} \gg 1$,
- and that, if $\|M(\lambda)\|_F \gg 1$, then $C_{P,\mathcal{L}} \gg 1$.
- Therefore, for getting "backward stability" from Block Kronecker linearizations, one needs to normalize the matrix poly $\|P(\lambda)\|_F = 1$ and to use pencils such that $\|M(\lambda)\|_F \approx \|P(\lambda)\|_F$, then

$$\frac{\|\Delta P(\lambda)\|_F}{\|P(\lambda)\|_F} \lesssim d^3 \sqrt{m+n}\, \frac{\|\Delta\mathcal{L}(\lambda)\|_F}{\|\mathcal{L}(\lambda)\|_F}.$$

For Fiedler, Frobenius, etc linearizations $\|M(\lambda)\|_F = \|P(\lambda)\|_F$.

$$\mathcal{L}(\lambda) = \left[\begin{array}{c|c} M(\lambda) & L_\eta(\lambda)^T \otimes I_m \\ \hline L_\varepsilon(\lambda) \otimes I_n & 0 \end{array}\right].$$

$$\frac{\|\Delta P(\lambda)\|_F}{\|P(\lambda)\|_F} \leq \underbrace{14\, d^{5/2} \frac{\|\mathcal{L}(\lambda)\|_F}{\|P(\lambda)\|_F} \left(1 + \|M(\lambda)\|_F + \|M(\lambda)\|_F^2\right)}_{C_{P,\mathcal{L}}} \frac{\|\Delta\mathcal{L}(\lambda)\|_F}{\|\mathcal{L}(\lambda)\|_F}.$$

- It can be proved that if $\|P(\lambda)\|_F \ll 1$ or $\|P(\lambda)\|_F \gg 1$, then $C_{P,\mathcal{L}} \gg 1$,
- and that, if $\|M(\lambda)\|_F \gg 1$, then $C_{P,\mathcal{L}} \gg 1$.
- Therefore, for getting "backward stability" from Block Kronecker linearizations, one needs to normalize the matrix poly $\|P(\lambda)\|_F = 1$ and to use pencils such that $\|M(\lambda)\|_F \approx \|P(\lambda)\|_F$, then

$$\frac{\|\Delta P(\lambda)\|_F}{\|P(\lambda)\|_F} \lesssim d^3 \sqrt{m+n}\, \frac{\|\Delta\mathcal{L}(\lambda)\|_F}{\|\mathcal{L}(\lambda)\|_F}.$$

For Fiedler, Frobenius, etc linearizations $\|M(\lambda)\|_F = \|P(\lambda)\|_F$.

Outline

- We consider a **general** $p \times m$ **rational matrix** expressed as

$$\boxed{G(\lambda) = D_d\lambda^d + \cdots + D_1\lambda + D_0 + C(\lambda I_n - A)^{-1}B},$$

- and we assume that its **complete ZERO and minimal index structure**
- has been computed by applying a **backward stable algorithm** (QZ for regular, Staircase for singular)
- to a **strong linearization** $\mathcal{L}(\lambda)$ **in the wide class of Rational block Kronecker linearizations** of $G(\lambda)$.
- **Is this process backward stable from the point of view of rational matrices?**
- This question is completely open in the literature.
- Joint **WORK IN PROGRESS** with M.C. Quintana and P. Van Dooren. I present just an idea of preliminary results

- We consider a **general** $p \times m$ **rational matrix** expressed as

$$\boxed{G(\lambda) = D_d\lambda^d + \cdots + D_1\lambda + D_0 + C(\lambda I_n - A)^{-1}B},$$

- and we assume that its **complete ZERO and minimal index structure**
- has been computed by applying a **backward stable algorithm** (QZ for regular, Staircase for singular)
- to a **strong linearization** $\mathcal{L}(\lambda)$ **in the wide class of Rational block Kronecker linearizations** of $G(\lambda)$.
- **Is this process backward stable from the point of view of rational matrices?**
- This question is completely open in the literature.
- Joint **WORK IN PROGRESS** with M.C. Quintana and P. Van Dooren. I present just an idea of preliminary results

The setting

- We consider a **general** $p \times m$ **rational matrix** expressed as

$$G(\lambda) = D_d\lambda^d + \cdots + D_1\lambda + D_0 + C(\lambda I_n - A)^{-1}B \quad ,$$

- and we assume that its **complete ZERO and minimal index structure**
- has been computed by applying a **backward stable algorithm** (QZ for regular, Staircase for singular)
- to a **strong linearization** $\mathcal{L}(\lambda)$ **in the wide class of Rational block Kronecker linearizations** of $G(\lambda)$.
- **Is this process backward stable from the point of view of rational matrices?**
- This question is completely open in the literature.
- Joint **WORK IN PROGRESS** with M.C. Quintana and P. Van Dooren. I present just an idea of preliminary results

The setting

- We consider a **general** $p \times m$ **rational matrix** expressed as

$$G(\lambda) = D_d\lambda^d + \cdots + D_1\lambda + D_0 + C(\lambda I_n - A)^{-1}B \quad ,$$

- and we assume that its **complete ZERO and minimal index structure**
- has been computed by applying a **backward stable algorithm** (QZ for regular, Staircase for singular)
- to a **strong linearization** $\mathcal{L}(\lambda)$ **in the wide class of Rational block Kronecker linearizations** of $G(\lambda)$.
- **Is this process backward stable from the point of view of rational matrices?**
- This question is completely open in the literature.
- Joint **WORK IN PROGRESS** with M.C. Quintana and P. Van Dooren. I present just an idea of preliminary results

- We consider a **general** $p \times m$ **rational matrix** expressed as

$$G(\lambda) = D_d\lambda^d + \cdots + D_1\lambda + D_0 + C(\lambda I_n - A)^{-1}B \quad ,$$

- and we assume that its **complete ZERO and minimal index structure**
- has been computed by applying a **backward stable algorithm** (QZ for regular, Staircase for singular)
- to a **strong linearization** $\mathcal{L}(\lambda)$ **in the wide class of Rational block Kronecker linearizations** of $G(\lambda)$.
- **Is this process backward stable from the point of view of rational matrices?**
- This question is completely open in the literature.
- Joint **WORK IN PROGRESS** with M.C. Quintana and P. Van Dooren. I present just an idea of preliminary results

- We consider a **general** $p \times m$ **rational matrix** expressed as

$$G(\lambda) = D_d\lambda^d + \cdots + D_1\lambda + D_0 + C(\lambda I_n - A)^{-1}B \quad ,$$

- and we assume that its **complete ZERO and minimal index structure**
- has been computed by applying a **backward stable algorithm** (QZ for regular, Staircase for singular)
- to a **strong linearization** $\mathcal{L}(\lambda)$ **in the wide class of Rational block Kronecker linearizations** of $G(\lambda)$.
- **Is this process backward stable from the point of view of rational matrices?**
- This question is completely open in the literature.
- Joint **WORK IN PROGRESS** with M.C. Quintana and P. Van Dooren. I present just an idea of preliminary results

The setting

- We consider a **general** $p \times m$ **rational matrix** expressed as

$$\boxed{G(\lambda) = D_d \lambda^d + \cdots + D_1 \lambda + D_0 + C(\lambda I_n - A)^{-1} B} \,,$$

- and we assume that its **complete ZERO and minimal index structure**
- has been computed by applying a **backward stable algorithm** (QZ for regular, Staircase for singular)
- to a **strong linearization** $\mathcal{L}(\lambda)$ **in the wide class of Rational block Kronecker linearizations** of $G(\lambda)$.
- **Is this process backward stable from the point of view of rational matrices?**
- This question is completely open in the literature.
- Joint **WORK IN PROGRESS** with M.C. Quintana and P. Van Dooren. I present just an idea of preliminary results

These are

$$\mathcal{L}(\lambda) = \left[\begin{array}{c|cc} \lambda I_n - A & B(e_{\varepsilon+1}^T \otimes I_m) & 0 \\ \hline -(e_{\eta+1} \otimes I_p)C & M(\lambda) & L_\eta(\lambda)^T \otimes I_p \\ 0 & L_\varepsilon(\lambda) \otimes I_m & 0 \end{array}\right].$$

An **example** we have already seen is for

$$G(\lambda) = \lambda^5 D_5 + \lambda^4 D_4 + \lambda^3 D_3 + \lambda^2 D_2 + \lambda D_1 + D_0 + C(\lambda I_n - A)^{-1} B$$

the strong linerization

$$\mathcal{L}(\lambda) = \left[\begin{array}{c|ccc|cc} \lambda I_n - A & 0 & 0 & B & 0 & 0 \\ \hline 0 & \lambda P_5 + P_4 & 0 & 0 & -I_p & 0 \\ 0 & 0 & \lambda P_3 + P_2 & 0 & \lambda I_p & -I_p \\ -C & 0 & 0 & \lambda P_1 + P_0 & 0 & \lambda I_p \\ \hline 0 & -I_m & \lambda I_m & 0 & 0 & 0 \\ 0 & 0 & -I_m & \lambda I_m & 0 & 0 \end{array}\right]$$

These are

$$\mathcal{L}(\lambda)=\left[\begin{array}{c|cc} \lambda I_n - A & B(e_{\varepsilon+1}^T\otimes I_m) & 0 \\ \hline -(e_{\eta+1}\otimes I_p)C & M(\lambda) & L_\eta(\lambda)^T\otimes I_p \\ 0 & L_\varepsilon(\lambda)\otimes I_m & 0 \end{array}\right].$$

An **example** we have already seen is for

$$G(\lambda)=\lambda^5 D_5+\lambda^4 D_4+\lambda^3 D_3+\lambda^2 D_2+\lambda D_1+D_0+C(\lambda I_n-A)^{-1}B$$

the strong linerization

$$\mathcal{L}(\lambda)=\left[\begin{array}{c|ccc|cc} \lambda I_n - A & 0 & 0 & B & 0 & 0 \\ \hline 0 & \lambda P_5+P_4 & 0 & 0 & -I_p & 0 \\ 0 & 0 & \lambda P_3+P_2 & 0 & \lambda I_p & -I_p \\ -C & 0 & 0 & \lambda P_1+P_0 & 0 & \lambda I_p \\ \hline 0 & -I_m & \lambda I_m & 0 & 0 & 0 \\ 0 & 0 & -I_m & \lambda I_m & 0 & 0 \end{array}\right]$$

Some auxiliary definitions

- Given $G(\lambda) = \sum_{i=0}^{d} \lambda^i D_i + C(\lambda I_n - A)^{-1} B$, we define

$$\|G(\lambda)\|_F = \sqrt{\sum_{i=0}^{d} \|D_i\|_F^2 + \|C\|_F^2 + \|I_n\|_F^2 + \|A\|_F^2 + \|B\|_F^2},$$

- which is the norm of the polynomial system matrix of $G(\lambda)$

$$P(\lambda) = \begin{bmatrix} \lambda I_n - A & B \\ -C & \sum_{i=0}^{d} \lambda^i D_i \end{bmatrix}.$$

- Given a perturbation of $G(\lambda)$, $\widehat{G}(\lambda) = \sum_{i=0}^{d} \lambda^i \widehat{D}_i + \widehat{C}(\lambda I_n - \widehat{A})^{-1} \widehat{B}$, we define (it is a definition!!)

$$\|G(\lambda) - \widehat{G}(\lambda)\|_F := \|\Delta G(\lambda)\|_F$$

$$:= \sqrt{\sum_{i=0}^{d} \|D_i - \widehat{D}_i\|_F^2 + \|C - \widehat{C}\|_F^2 + \|A - \widehat{A}\|_F^2 + \|B - \widehat{B}\|_F^2},$$

which is the norm of the difference of the polynomial system matrices of $G(\lambda)$ and $\widehat{G}(\lambda)$.

Some auxiliary definitions

- Given $G(\lambda) = \sum_{i=0}^{d} \lambda^i D_i + C(\lambda I_n - A)^{-1} B$, we define

$$\|G(\lambda)\|_F = \sqrt{\sum_{i=0}^{d} \|D_i\|_F^2 + \|C\|_F^2 + \|I_n\|_F^2 + \|A\|_F^2 + \|B\|_F^2},$$

- which is the norm of the polynomial system matrix of $G(\lambda)$

$$P(\lambda) = \begin{bmatrix} \lambda I_n - A & B \\ -C & \sum_{i=0}^{d} \lambda^i D_i \end{bmatrix}.$$

- Given a perturbation of $G(\lambda)$, $\widehat{G}(\lambda) = \sum_{i=0}^{d} \lambda^i \widehat{D}_i + \widehat{C}(\lambda I_n - \widehat{A})^{-1} \widehat{B}$, we define (it is a definition!!)

$$\|G(\lambda) - \widehat{G}(\lambda)\|_F := \|\Delta G(\lambda)\|_F$$

$$:= \sqrt{\sum_{i=0}^{d} \|D_i - \widehat{D}_i\|_F^2 + \|C - \widehat{C}\|_F^2 + \|A - \widehat{A}\|_F^2 + \|B - \widehat{B}\|_F^2},$$

which is the norm of the difference of the polynomial system matrices of $G(\lambda)$ and $\widehat{G}(\lambda)$.

Some auxiliary definitions

- Given $G(\lambda) = \sum_{i=0}^{d} \lambda^i D_i + C(\lambda I_n - A)^{-1} B$, we define

$$\|G(\lambda)\|_F = \sqrt{\sum_{i=0}^{d} \|D_i\|_F^2 + \|C\|_F^2 + \|I_n\|_F^2 + \|A\|_F^2 + \|B\|_F^2},$$

- which is the norm of the polynomial system matrix of $G(\lambda)$

$$P(\lambda) = \begin{bmatrix} \lambda I_n - A & B \\ -C & \sum_{i=0}^{d} \lambda^i D_i \end{bmatrix}.$$

- Given a perturbation of $G(\lambda)$, $\widehat{G}(\lambda) = \sum_{i=0}^{d} \lambda^i \widehat{D}_i + \widehat{C}(\lambda I_n - \widehat{A})^{-1} \widehat{B}$, we define (it is a definition!!)

$$\|G(\lambda) - \widehat{G}(\lambda)\|_F := \|\Delta G(\lambda)\|_F$$

$$:= \sqrt{\sum_{i=0}^{d} \|D_i - \widehat{D}_i\|_F^2 + \|C - \widehat{C}\|_F^2 + \|A - \widehat{A}\|_F^2 + \|B - \widehat{B}\|_F^2},$$

which is the norm of the difference of the polynomial system matrices of $G(\lambda)$ and $\widehat{G}(\lambda)$.

Theorem (D, Quintana, Van Dooren, in progress, 2018)

Let $\mathcal{L}(\lambda)$ be a rational block Kronecker strong linearization of

$$G(\lambda) = \sum_{i=0}^{d} \lambda^i D_i + C(\lambda I_n - A)^{-1} B \in \mathbb{C}(\lambda)^{p\times m}.$$

If $\Delta\mathcal{L}(\lambda)$ is any sufficiently small pencil with the same size as $\mathcal{L}(\lambda)$, then the EIGENVALUE AND MINIMAL INDEX STRUCTURE OF $\mathcal{L}(\lambda) + \Delta\mathcal{L}(\lambda)$ corresponds exactly to the ZERO AND MINIMAL INDEX STRUCTURE of a rational matrix

$$\widehat{G}(\lambda) = \sum_{i=0}^{d} \lambda^i \widehat{D}_i + \widehat{C}(\lambda I_n - \widehat{A})^{-1} \widehat{B} \in \mathbb{C}(\lambda)^{p\times m},$$

such that, to first order in $\|\Delta\mathcal{L}(\lambda)\|_F$,

$$\frac{\|\Delta G(\lambda)\|_F}{\|G(\lambda)\|_F} \leq p(d)\, \frac{\|\mathcal{L}(\lambda)\|_F}{\|G(\lambda)\|_F}\, \mathbf{C_G}\, (1 + \|M(\lambda)\|_F + \|M(\lambda)\|_F^2)\, \frac{\|\Delta\mathcal{L}(\lambda)\|_F}{\|\mathcal{L}(\lambda)\|_F},$$

where

$$\mathbf{C_G} = \|C\|_2 + \|A\|_2^{\max\{\varepsilon,\eta\}} + \|B\|_2\,.$$

There is a penalty with respect to the polynomial case!!!

- $\mathbf{C_G} = \|C\|_2 + \|A\|_2^{\max\{\varepsilon,\eta\}} + \|B\|_2$ depends on the particular state-space realization of the strictly proper part that is used, which **is natural** since there are infinitely many of such realizations:
- $G(\lambda) = \sum_{i=0}^{d} \lambda^i D_i + CT^{-1}(\lambda I_n - TAT^{-1})^{-1}TB.$
- This effect **has been observed** in numerical tests!! (next slide)
- However, for block Kronecker strong linearizations such that $\|M(\lambda)\|_F \approx \|D(\lambda)\|_F$, we have proved that:
  1. There exists a scaling, $G_s(\lambda_s) = d_r\, G(d_\lambda \lambda)$, and
  2. and a balancing diagonal T,
- that transform the original REP into another REP such that

$$\frac{\|\mathcal{L}(\lambda)\|_F}{\|G(\lambda)\|_F}\, \mathbf{C_G}\, (1 + \|M(\lambda)\|_F + \|M(\lambda)\|_F^2) \approx f(d,p,m),$$

with $f(d,p,m)$ a slowing increasing function of d, p, and m.

There is a penalty with respect to the polynomial case!!!

- $\mathbf{C_G} = \|C\|_2 + \|A\|_2^{\max\{\varepsilon,\eta\}} + \|B\|_2$ depends on the particular state-space realization of the strictly proper part that is used, which **is natural** since there are infinitely many of such realizations:
- $G(\lambda) = \sum_{i=0}^{d} \lambda^i D_i + CT^{-1}(\lambda I_n - TAT^{-1})^{-1}TB$.
- This effect **has been observed** in numerical tests!! (next slide)
- However, for block Kronecker strong linearizations such that $\|M(\lambda)\|_F \approx \|D(\lambda)\|_F$, we have proved that:
  1. There exists a scaling, $G_s(\lambda_s) = d_r\, G(d_\lambda \lambda)$, and
  2. and a balancing diagonal T,
- that transform the original REP into another REP such that

$$\frac{\|\mathcal{L}(\lambda)\|_F}{\|G(\lambda)\|_F}\, \mathbf{C_G}\, (1 + \|M(\lambda)\|_F + \|M(\lambda)\|_F^2) \approx f(d,p,m),$$

with $f(d,p,m)$ a slowing increasing function of d, p, and m.

There is a penalty with respect to the polynomial case!!!

- $\mathbf{C_G} = \|C\|_2 + \|A\|_2^{\max\{\varepsilon,\eta\}} + \|B\|_2$ depends on the particular state-space realization of the strictly proper part that is used, which **is natural** since there are infinitely many of such realizations:
- $G(\lambda) = \sum_{i=0}^{d} \lambda^i D_i + CT^{-1}(\lambda I_n - TAT^{-1})^{-1}TB$.
- This effect **has been observed** in numerical tests!! (next slide)
- However, for block Kronecker strong linearizations such that $\|M(\lambda)\|_F \approx \|D(\lambda)\|_F$, we have proved that:
  1. There exists a scaling, $G_s(\lambda_s) = d_r\, G(d_\lambda \lambda)$, and
  2. and a balancing diagonal T,
- that transform the original REP into another REP such that

$$\frac{\|\mathcal{L}(\lambda)\|_F}{\|G(\lambda)\|_F}\, \mathbf{C_G}\, (1 + \|M(\lambda)\|_F + \|M(\lambda)\|_F^2) \approx f(d,p,m),$$

with $f(d,p,m)$ a slowing increasing function of $d, p,$ and m.

There is a penalty with respect to the polynomial case!!!

- $\mathbf{C_G} = \|C\|_2 + \|A\|_2^{\max\{\varepsilon,\eta\}} + \|B\|_2$ depends on the particular state-space realization of the strictly proper part that is used, which **is natural** since there are infinitely many of such realizations:
- $G(\lambda) = \sum_{i=0}^{d} \lambda^i D_i + CT^{-1}(\lambda I_n - TAT^{-1})^{-1}TB$.
- This effect **has been observed** in numerical tests!! (next slide)
- However, for block Kronecker strong linearizations such that $\|M(\lambda)\|_F \approx \|D(\lambda)\|_F$, we have proved that:
  1. There exists a scaling, $G_s(\lambda_s) = d_r\, G(d_\lambda \lambda)$, and
  2. and a balancing diagonal T,
- that transform the original REP into another REP such that

$$\frac{\|\mathcal{L}(\lambda)\|_F}{\|G(\lambda)\|_F}\, \mathbf{C_G}\, (1 + \|M(\lambda)\|_F + \|M(\lambda)\|_F^2) \approx f(d,p,m),$$

with $f(d,p,m)$ a slowing increasing function of $d, p,$ and m.

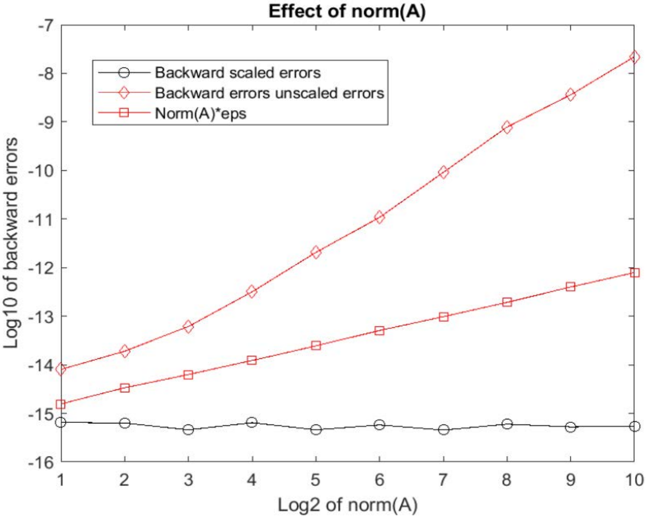
There is a penalty with respect to the polynomial case!!!

- $\mathbf{C_G} = \|C\|_2 + \|A\|_2^{\max\{\varepsilon,\eta\}} + \|B\|_2$ depends on the particular state-space realization of the strictly proper part that is used, which **is natural** since there are infinitely many of such realizations:
- $G(\lambda) = \sum_{i=0}^{d} \lambda^i D_i + CT^{-1}(\lambda I_n - TAT^{-1})^{-1}TB$.
- This effect **has been observed** in numerical tests!! (next slide)
- However, for block Kronecker strong linearizations such that $\|M(\lambda)\|_F \approx \|D(\lambda)\|_F$, we have proved that:
  1. There exists a scaling, $G_s(\lambda_s) = d_r\, G(d_\lambda \lambda)$, and
  2. and a balancing diagonal T,
- that transform the original REP into another REP such that

$$\frac{\|\mathcal{L}(\lambda)\|_F}{\|G(\lambda)\|_F}\, \mathbf{C_G}\, (1 + \|M(\lambda)\|_F + \|M(\lambda)\|_F^2) \approx f(d,p,m),$$

with $f(d,p,m)$ a slowing increasing function of $d, p,$ and m.

Numerical test on backward errors for zeros of REPs



Outline

Conclusions

- There are many matrix eigenvalue problems in addition to the basic one that are attracting a lot of attention in the last 15 years.
- There are still many open problems in this area: development of algorithms, approximation of NEPs by REPs, theoretical understanding of REPs, and stability analyses.
- We have developed new classes of linearizations of PEPs that unify and extend the previous ones and, for the first time in the literature, a theory of **strong** linearizations of REPs.
- We have have performed a backward stability analysis of PEPs solved with linearizations that improve previous analyses in generality and quality, but more general analyses, including PEPs represented in other bases, are necessary.
- We have performed for the first time in the literature a backward stability analysis of REPs solved with linearizations, which confirms (from another perspective) that REPs are more difficult than PEPs, but this is just the beginning of these analyses...