Matlab Tools for Computing Canonical Structure Information

ILAS 2019, Rio de Janeiro

Stefan Johansson
joint work with Bo Kågström

Department of Computing Science
Umeå University
Sweden

July 8-12, 2019

UMEÅ UNIVERSITY

The Settings

Consider a matrix pencil

$$G - \lambda H, \qquad \lambda \in \mathbb{C}$$

where $G, H \in \mathbb{C}^{m\times n}$

Goal

Robust Matlab software for computing and analyzing canonical structure information

Code contributions also by A. Dmytryshyn and P. Johansson

Canonical Structure Information

A canonical form (CF) reveals the canonical structure information from which the system characteristics are deduced

- Matrices A
- Matrix pencils $G - \lambda H$
- System pencils $S - \lambda T$
- Matrix polynomials $P(\lambda) = \lambda^d A_d + \ldots + \lambda A_1 + A_0$

Canonical Structure Information

A canonical form (CF) reveals the canonical structure information from which the system characteristics are deduced

- Matrices A – *Jordan CF*
- Matrix pencils $G - \lambda H$ – *Kronecker CF*
- System pencils $S - \lambda T$ – *Brunovsky-type CF*
- Matrix polynomials $P(\lambda)$ – *Smith CF*

A canonical form is the simplest or most symmetrical form a matrix or matrix pencil can be reduced to

Jordan Canonical Form

The *Jordan canonical form* (JCF) reveals the fine eigenstructure of a square matrix A:

$$SAS^{-1} = J, \quad \text{where } S \text{ is non-singular}$$

Jordan Canonical Form

The *Jordan canonical form* (JCF) reveals the fine eigenstructure of a square matrix A:

$$SAS^{-1} = J, \quad \text{where } S \text{ is non-singular}$$

J is a *block-diagonal matrix* where each block is a $k \times k$ *Jordan block* (a regular block) for an eigenvalue μ of A:

$$J_k(\mu) = \begin{bmatrix} \mu & 1 & & \\ & \mu & \ddots & \\ & & \ddots & 1 \\ & & & \mu \end{bmatrix}$$

Geometric multiplicity – Number of Jordan block for μ
Algebraic multiplicity – Total size of all Jordan blocks for μ

Kronecker Canonical Form

For *non-square* matrices and matrix pencils $G - \lambda H$, the canonical form may also consist of *singular blocks*

The *Kronecker canonical form* reveals the fine eigenstructure of a matrix pencil

Kronecker Canonical Form

Any matrix pencil $G - \lambda H$ can be transformed into *Kronecker canonical form* (KCF) using equivalence transformations (U and V non-singular):

$$U(G - \lambda H)V^{-1} =$$
$$\mathrm{diag}(L_{\epsilon_1}, \ldots, L_{\epsilon_p}, J(\mu_1) - \lambda I, \ldots, J(\mu_t) - \lambda I, N_{h_1^{(\infty)}}, \ldots, N_{h_g^{(\infty)}}, L_{\eta_1}^T, \ldots, L_{\eta_q}^T)$$

Kronecker Canonical Form

Any matrix pencil $G - \lambda H$ can be transformed into *Kronecker canonical form* (KCF) using equivalence transformations (U and V non-singular):

$$U(G - \lambda H)V^{-1} =$$

$$\operatorname{diag}(L_{\epsilon_1}, \ldots, L_{\epsilon_p}, J(\mu_1) - \lambda I, \ldots, J(\mu_t) - \lambda I, N_{h_1^{(\infty)}}, \ldots, N_{h_g^{(\infty)}}, L^T_{\eta_1}, \ldots, L^T_{\eta_q})$$

Singular part:

- $L_{\epsilon_1}, \ldots, L_{\epsilon_p}$ – Right singular blocks
- $L^T_{\eta_1}, \ldots, L^T_{\eta_q}$ – Left singular blocks

$$L_k = \begin{bmatrix} -\lambda & 1 & & \\ & \ddots & \ddots & \\ & & -\lambda & 1 \end{bmatrix}$$

of size $k \times (k+1)$

Kronecker Canonical Form

Any matrix pencil $G - \lambda H$ can be transformed into *Kronecker canonical form* (KCF) using equivalence transformations (U and V non-singular):

$$U(G - \lambda H)V^{-1} =$$

$$\mathrm{diag}(L_{\epsilon_1}, \ldots, L_{\epsilon_p}, J(\mu_1) - \lambda I, \ldots, J(\mu_t) - \lambda I, N_{h_1^{(\infty)}}, \ldots, N_{h_g^{(\infty)}}, L_{\eta_1}^T, \ldots, L_{\eta_q}^T)$$

Singular part:

- $L_{\epsilon_1}, \ldots, L_{\epsilon_p}$ – Right singular bloc
- $L_{\eta_1}^T, \ldots, L_{\eta_q}^T$ – Left singular block

$$J_k(\mu_i) - \lambda I = \begin{bmatrix} \mu_i - \lambda & 1 & & \\ & \ddots & \ddots & \\ & & \ddots & 1 \\ & & & \mu_i - \lambda \end{bmatrix}$$

Regular part:

- $J(\mu_1), \ldots, J(\mu_t)$ – Each $J(\mu_i)$ is block-diagonal with *Jordan* blocks corresponding to the finite eigenvalue μ_i
- $N_{h_1^{(\infty)}}, \ldots, N_{h_g^{(\infty)}}$ – Each $N_k = I - \lambda J_k(0)$ and corresponds to the infinite eigenvalue

STAIRCASE FORMS

Using orthogonal or unitary matrices and backward stable algorithms:

Matrices The Jordan-Schur form [Kublanovskaya'66]

Matrix pencils A generalized Schur-staircase form [Van Dooren'79], e.g. the *Guptri staircase form* [Demmel-Kågström'93]

Staircase Forms

Using orthogonal or unitary matrices and backward stable algorithms:

Matrices The Jordan-Schur form [Kublanovskaya'66]

Matrix pencils A generalized Schur-staircase form [Van Dooren'79], e.g. the *Guptri staircase form* [Demmel-Kågström'93]

Computing the canonical structure information is an **ill-posed problem**!

THE GUPTRI STAIRCASE FORM

GUPTRI – Generalized UPper TRIangular form
[Demmel-Kågström'93]

$$P^H(G - \lambda H)Q = \begin{bmatrix} A_r - \lambda B_r & * & * \\ 0 & A_{reg} - \lambda B_{reg} & * \\ 0 & 0 & A_l - \lambda B_l \end{bmatrix}$$

THE GUPTRI STAIRCASE FORM

GUPTRI – Generalized UPper TRIangular form
[Demmel-Kågström'93]

$$P^H(G - \lambda H)Q = \begin{bmatrix} A_r - \lambda B_r & * & * \\ 0 & A_{reg} - \lambda B_{reg} & * \\ 0 & 0 & A_l - \lambda B_l \end{bmatrix}$$

where P and Q are unitary matrices

$A_r - \lambda B_r$ – Right singular part; L blocks
$A_l - \lambda B_l$ – Left singular part; L^T blocks

$A_r - \lambda B_r$ and $A_l - \lambda B_l$ are block-upper triangular

THE GUPTRI STAIRCASE FROM - REGULAR PART

$$A_{reg} - \lambda B_{reg} = \begin{bmatrix} A_z & * & * \\ 0 & A_f & * \\ 0 & 0 & A_i \end{bmatrix} - \lambda \begin{bmatrix} B_z & * & * \\ 0 & B_f & * \\ 0 & 0 & B_i \end{bmatrix}$$

where $A_{reg} - \lambda B_{reg}$ is upper triangular and

$A_z - \lambda B_z$ – The zero eigenvalue $\mu = 0$

$A_f - \lambda B_f$ – Finite non-zero eigenvalues μ

$A_i - \lambda B_i$ – The infinite eigenvalue $\mu = \infty$

The GUPTRI Staircase From - Regular Part

$$A_{reg} - \lambda B_{reg} = \begin{bmatrix} A_z & * & * \\ 0 & A_f & * \\ 0 & 0 & A_i \end{bmatrix} - \lambda \begin{bmatrix} B_z & * & * \\ 0 & B_f & * \\ 0 & 0 & B_i \end{bmatrix}$$

where $A_{reg} - \lambda B_{reg}$ is upper triangular and

$A_z - \lambda B_z$ – The zero eigenvalue $\mu = 0$

- $a_i = 0$ and $b_i \neq 0$

$A_f - \lambda B_f$ – Finite non-zero eigenvalues μ

- $\mu_i = a_i/b_i$ where $a_i \neq 0$ and $b_i \neq 0$

$A_i - \lambda B_i$ – The infinite eigenvalue $\mu = \infty$

- $a_i \neq 0$ and $b_i = 0$

The Guptri Algorithm

Seven steps in total:

1. Compute the RZ-staircase form (RZ-algorithm):
   Right singular structures and zero eigenvalues in $A_{rz} - \lambda B_{rz}$
   - Uses SVD and column compression on the (deflated) pencil $G^{(k)} - \lambda H^{(k)}$, $k = 0, 1, \ldots$
   - Column compress $G^{(k)}$ using $\mathrm{SVD}(G^{(k)}) = (*, \Sigma_A, V_A)$:

     $s_k =$ dim column nullspace of $G^{(k)} = \#(\sigma(G^{(k)}) < tolG)$ (insist on gap)

The Guptri Algorithm

Seven steps in total:

1. Compute the RZ-staircase form (RZ-algorithm):
   Right singular structures and zero eigenvalues in $A_{rz} - \lambda B_{rz}$
2. Separate right singular part from regular part:
   Obtain $A_r - \lambda B_r$ in staircase form
   - Apply RZ-algorithm on $B_{rz} - \mu A_{rz}$
   - Insist on the same right singular structure as in Step 1

THE GUPTRI ALGORITHM

Seven steps in total:

1. Compute the RZ-staircase form (RZ-algorithm):
   Right singular structures and zero eigenvalues in $A_{rz} - \lambda B_{rz}$
2. Separate right singular part from regular part:
   Obtain $A_r - \lambda B_r$ in staircase form
3. Obtain $A_z - \lambda B_z$ in staircase form
   - Apply RZ-algorithm on $A_z - \lambda B_z$
   - Insist on the same regular structure as in Step 1

The Guptri Algorithm

Seven steps in total:

1. Compute the RZ-staircase form (RZ-algorithm):
   Right singular structures and zero eigenvalues in $A_{rz} - \lambda B_{rz}$
2. Separate right singular part from regular part:
   Obtain $A_r - \lambda B_r$ in staircase form
3. Obtain $A_z - \lambda B_z$ in staircase form

4-6. Apply the LI-algorithm (from lower-right corner):
Obtain left singular structures in $A_l - \lambda B_l$ and infinite eigenvalues in $A_i - \lambda B_i$
   - Work on $B - \mu A$ using row compression
   - In total 3 steps

The Guptri Algorithm

Seven steps in total:

1. Compute the RZ-staircase form (RZ-algorithm):
   Right singular structures and zero eigenvalues in $A_{rz} - \lambda B_{rz}$
2. Separate right singular part from regular part:
   Obtain $A_r - \lambda B_r$ in staircase form
3. Obtain $A_z - \lambda B_z$ in staircase form

4-6. Apply the LI-algorithm (from lower-right corner):
   Obtain left singular structures in $A_l - \lambda B_l$ and
   infinite eigenvalues in $A_i - \lambda B_i$

7. Apply RZ-algorithm on the remaining square regular pencil:
   Finite non-zero eigenvalues in $A_f - \lambda B_f$
   - Includes clustering of the eigenvalues

GUPTRI ALGORITHM ILLUSTRATED

Initial matrix:

$$G - \lambda H \qquad =$$

$$\begin{bmatrix} * \end{bmatrix} - \lambda \begin{bmatrix} * \end{bmatrix}$$

Guptri Algorithm Illustrated

Step 1:

$$P_{rz}^{H}(G - \lambda H)Q_{rz} =$$

$$\begin{bmatrix} A_{rz} & * \\ 0 & * \end{bmatrix} - \lambda \begin{bmatrix} B_{rz} & * \\ 0 & * \end{bmatrix}$$

GUPTRI ALGORITHM ILLUSTRATED

Steps 2–3:

$$P_{r-z}^{H} P_{rz}^{H} (G - \lambda H) Q_{rz} Q_{r-z} =$$

$$\begin{bmatrix} A_r & * & * \\ 0 & A_z & * \\ 0 & 0 & * \end{bmatrix} - \lambda \begin{bmatrix} B_r & * & * \\ 0 & B_z & * \\ 0 & 0 & * \end{bmatrix}$$

GUPTRI ALGORITHM ILLUSTRATED

Step 4:

$$P_{li}^H P_{r-z}^H P_{rz}^H (G - \lambda H) Q_{rz} Q_{r-z} Q_{li} \quad =$$

$$\begin{bmatrix} A_r & * & * & * \\ 0 & A_z & * & * \\ 0 & 0 & * & * \\ 0 & 0 & 0 & A_{li} \end{bmatrix} - \lambda \begin{bmatrix} B_r & * & * & * \\ 0 & B_z & * & * \\ 0 & 0 & * & * \\ 0 & 0 & 0 & B_{li} \end{bmatrix}$$

GUPTRI ALGORITHM ILLUSTRATED

Steps 5–6:

$$P_{l-i}^H P_{li}^H P_{r-z}^H P_{rz}^H (G - \lambda H) Q_{rz} Q_{r-z} Q_{li} Q_{l-i} \quad =$$

$$\begin{bmatrix} A_r & * & * & * & * \\ 0 & A_z & * & * & * \\ 0 & 0 & * & * & * \\ 0 & 0 & 0 & A_i & * \\ 0 & 0 & 0 & 0 & A_l \end{bmatrix} - \lambda \begin{bmatrix} B_r & * & * & * & * \\ 0 & B_z & * & * & * \\ 0 & 0 & * & * & * \\ 0 & 0 & 0 & B_i & * \\ 0 & 0 & 0 & 0 & B_l \end{bmatrix}$$

Guptri Algorithm Illustrated

Step 7:

$$P_f^H P_{l-i}^H P_{li}^H P_{r-z}^H P_{rz}^H (G - \lambda H) Q_{rz} Q_{r-z} Q_{li} Q_{l-i} Q_f =$$

$$\begin{bmatrix} A_r & * & * & * & * \\ 0 & A_z & * & * & * \\ 0 & 0 & A_f & * & * \\ 0 & 0 & 0 & A_i & * \\ 0 & 0 & 0 & 0 & A_l \end{bmatrix} - \lambda \begin{bmatrix} B_r & * & * & * & * \\ 0 & B_z & * & * & * \\ 0 & 0 & B_f & * & * \\ 0 & 0 & 0 & B_i & * \\ 0 & 0 & 0 & 0 & B_l \end{bmatrix}$$

where a matrix pair (A_x, B_x) is non-existing (0×0) for an absent invariant

CLUSTERING OF EIGENVALUES - MOTIVATION

Let A be a 5×5 matrix in JCF

$$A = \begin{bmatrix} 3 & 1 & 0 & 0 & 0 \\ 0 & 3 & 1 & 0 & 0 \\ 0 & 0 & 3 & 1 & 0 \\ 0 & 0 & 0 & 3 & 1 \\ \alpha & 0 & 0 & 0 & 3 \end{bmatrix}$$

If $\alpha = 0$, A has one eigenvalue $\mu = 3$ with multiplicity 5

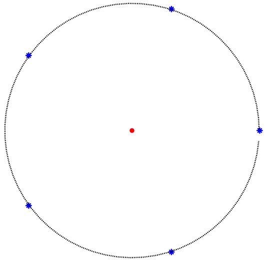
CLUSTERING OF EIGENVALUES - MOTIVATION

Let A be a 5×5 matrix in JCF

$$A = \begin{bmatrix} 3 & 1 & 0 & 0 & 0 \\ 0 & 3 & 1 & 0 & 0 \\ 0 & 0 & 3 & 1 & 0 \\ 0 & 0 & 0 & 3 & 1 \\ \alpha & 0 & 0 & 0 & 3 \end{bmatrix}$$

If $\alpha = 0$, A has one eigenvalue $\mu = 3$ with multiplicity 5

If $\alpha = 10^{-14}$, A has 5 eigenvalues at the distance $\sqrt[n]{\alpha} = \sqrt[5]{10^{-14}}$ from μ



Clustering of Eigenvalues

Hierarchical cluster methods:

- Using Gershgorin circles to cluster the eigenvalues starting with the eigenvalue closest to a goal
  [Gershgorin'31, Kågström-Wiberg'92]

CLUSTERING OF EIGENVALUES

Hierarchical cluster methods:

- Using Gershgorin circles to cluster the eigenvalues starting with the eigenvalue closest to a goal [Gershgorin'31, Kågström-Wiberg'92]
- Using a distance function $D(X,Y) \leq \delta_{max}$ where X and Y are different clusters of eigenvalues, and where a minimum number P_{min} of clusters can be guaranteed [Kintzel'03]

  Possible functions $D(X,Y)$ are:

min	$\min(\text{abs}(X(i) - Y(j)))$, for all i, j
max	$\max(\text{abs}(X(i) - Y(j)))$, for all i, j
average	$1/(rs)(\sum^{r,s} \text{abs}(X(i) - Y(j)))$
mean	$\text{abs}(\sum(X)/r - \sum(Y)/s)$
variance	$\text{var}(X \cup Y)$

  where r and s are #eigenvalues in each cluster

Matrix Canonical Structure (MCS) Toolbox

The MCS Toolbox – A Matlab toolbox for computing and representing canonical structure information

Includes data type objects for

- Canonical Structure Objects
- Canonical Block Objects

Matrix Canonical Structure (MCS) Toolbox

The MCS Toolbox – A Matlab toolbox for computing and representing canonical structure information

Includes data type objects for

- Canonical Structure Objects
- Canonical Block Objects

StratiGraph – Software tool for stratification of closure hierarchies

Canonical Structure Objects

Canonical Structure Objects are data type objects for representing the canonical structure information for a matrix setup

Canonical Structure Objects

Canonical Structure Objects are data type objects for representing the canonical structure information for a matrix setup

Available canonical structure objects:

`pstruct` Matrix pencils under strict equivalence

Canonical Structure Objects

Canonical Structure Objects are data type objects for representing the canonical structure information for a matrix setup

Available canonical structure objects:

`pstruct`	Matrix pencils under strict equivalence
`mstruct`	Matrices under similarity
`mpstruct`	Matrix polynomials under strict equivalence of the corresponding Fiedler linearization
`ssstruct`	State-space system pencils under feedback-injection equivalence
`cmstruct`	Matrices under congruence
`scmstruct`	Matrices under *congruence
`spstruct`	Symmetric matrix pencils under congruence
`sspstruct`	Skew-symmetric matrix pencils under congruence

Canonical Structure Objects

Methods in canonical structure objects for

- accessing and changing the canonical structure
  - `get`, `set`, `compare`, `copy`, etc.
- obtaining sizes and codimension of the matrix representation
  - `size`, `numblk`, `codim`, etc.
- obtaining the corresponding canonical form in matrix form
  - `jnf`, `kcf`, `bcf`, etc.

Canonical Block Objects

Canonical Block Objects are data type objects for representing canonical blocks

Canonical Block Objects

Canonical Block Objects are data type objects for representing canonical blocks

Available canonical block objects:

- `fjblock` Jordan block for a finite eigenvalues
- `zjblock` Jordan block for the zero eigenvalue
- `ijblock` Jordan block for the infinite eigenvalue
- `rsblock` Right singular block
- `lsblock` Left singular block
- ... And corresponding block objects for (skew-)symmetric matrix pencils and matrices under congruence and *congruence

EXAMPLE - CANONICAL STRUCTURE OBJECT

Create a matrix pencil object with KCF $L_2 \oplus L_0^T \oplus 2J_1(3+i2)$

```
>> pstr = pstruct('rsblock',2,'lsblock',0,'fjblock',{[1 1],3+2i})
pstr =
     R2 + L0 + 2*J1(3 + 2i)
```

Example - Canonical Structure Object

Create a matrix pencil object with KCF $L_2 \oplus L_0^T \oplus 2J_1(3+i2)$

```
>> pstr = pstruct('rsblock',2,'lsblock',0,'fjblock',{[1 1],3+2i})
pstr =
     R2 + L0 + 2*J1(3 + 2i)
```

Get the size of the matrix pencil

```
>> pstr.size
ans =
     5     5
```

Example - Canonical Structure Object

Create a matrix pencil object with KCF $L_2 \oplus L_0^T \oplus 2J_1(3+i2)$

```
>> pstr = pstruct('rsblock',2,'lsblock',0,'fjblock',{[1 1],3+2i})
pstr =
     R2 + L0 + 2*J1(3 + 2i)
```

Get the size of the matrix pencil

```
>> pstr.size
ans =
     5     5
```

Get the block object for the Jordan blocks

```
>> fjblk = pstr.get('fjblock','object');
>> fjblk{1}
ans =
     2*J1(3.0000 + 2.0000i)
```

Example - Canonical Structure Object

Recall pstr has the KCF $L_2 \oplus L_0^T \oplus 2J_1(3+i2)$

```
>> pstr = pstruct('rsblock',2,'lsblock',0,'fjblock',{[1 1],3+2i})
pstr =
     R2 + L0 + 2*J1(3 + 2i)
```

Get the KCF in matrix form

```
>> [G,H] = pstr.kcf
G =                                                 H =
     0+0i   1+0i   0+0i   0+0i   0+0i                    1     0     0     0     0
     0+0i   0+0i   1+0i   0+0i   0+0i                    0     1     0     0     0
     0+0i   0+0i   0+0i   0+0i   0+0i                    0     0     0     0     0
     0+0i   0+0i   0+0i   3+2i   0+0i                    0     0     0     1     0
     0+0i   0+0i   0+0i   0+0i   3+2i                    0     0     0     0     1
```

Matrix Canonical Structure (MCS) Toolbox

Numerical routines for computing

- the *staircase form* (the Guptri form) of matrices and matrix pencils
  - `mguptri` and `pguptri`

MATRIX CANONICAL STRUCTURE (MCS) TOOLBOX

Numerical routines for computing

- the *staircase form* (the Guptri form) of matrices and matrix pencils
  - `mguptri` and `pguptri`
- a matrix representation of the *tangent space* of an orbit
- the *codimension*, either from matrices or the canonical structure

Matrix Canonical Structure (MCS) Toolbox

Numerical routines for computing

- the *staircase form* (the Guptri form) of matrices and matrix pencils
  - `mguptri` and `pguptri`
- a matrix representation of the *tangent space* of an orbit
- the *codimension*, either from matrices or the canonical structure

Also includes

- a gallery of sample matrices and matrix pencils for staircase computations

THE GUPTRI FORM FOR MATRIX PENCILS

```
pstr = pguptri(G,H)
```

compute the canonical structure information of $G - \lambda H$
where `pstr` is a matrix pencil object

The Guptri Form For Matrix Pencils

```
pstr = pguptri(G,H)
```

compute the canonical structure information of $G - \lambda H$
where `pstr` is a matrix pencil object

Returning the Guptri staircase form:

```
[A,B,P,Q,dG,dH,pstr] = pguptri(G,H)
```

where P and Q are unitary and A and B are in generalized upper triangular form (Guptri staircase form), such that

$$P(A - \lambda B)Q^H = (G + dG) - \lambda(H + dH)$$

The Guptri Form For Matrix Pencils

```
pstr = pguptri(G,H)
```

compute the canonical structure information of $G - \lambda H$
where `pstr` is a matrix pencil object

Returning the Guptri staircase form:

```
[A,B,P,Q,dG,dH,pstr] = pguptri(G,H)
```

where P and Q are unitary and A and B are in generalized upper triangular form (Guptri staircase form), such that

$$P(A - \lambda B)Q^H = (G + dG) - \lambda(H + dH)$$

$\|(dG, dH)\|_F$ is an upper bound on the distance from $G - \lambda H$ to a matrix pencil with the computed Kronecker structure

UPPER BOUND ON ERROR

If the small singular values $< tol$ are kept in the deflation

$$\|(dG, dH)\| \approx \|(G, H)\| * eps$$

Otherwise

$$\|(dG, dH)\| \approx \|(G, H)\| * tol$$

Specifying Regular Block Order

The block order of the regular part

$$A_{reg} - \lambda B_{reg} = \begin{bmatrix} A_z & * & * \\ 0 & A_f & * \\ 0 & 0 & A_i \end{bmatrix} - \lambda \begin{bmatrix} B_z & * & * \\ 0 & B_f & * \\ 0 & 0 & B_i \end{bmatrix}$$

can be changed

- Reordering of the sub-pencils are done by solving a generalized Sylvester equation

Specifying Regular Block Order

The block order of the regular part

$$A_{reg} - \lambda B_{reg} = \begin{bmatrix} A_z & * & * \\ 0 & A_f & * \\ 0 & 0 & A_i \end{bmatrix} - \lambda \begin{bmatrix} B_z & * & * \\ 0 & B_f & * \\ 0 & 0 & B_i \end{bmatrix}$$

can be changed

- Reordering of the sub-pencils are done by solving a generalized Sylvester equation

Specified with an optional parameter to `guptri`

'zfi'	Zero-Finite-Infinite	**'fiz'**	Finite-Infinite-Zero
'zif'	Zero-Infinite-Finite	**'ifz'**	Infinite-Finite-Zero
'fzi'	Finite-Zero-Infinite	**'izf'**	Infinite-Zero-Finite

CLUSTERING OF EIGENVALUES

Eigenvalue clustering method is specified with (default 'var')

'norm'	Cluster within a norm distance
'gersh'	Cluster using Gershgorin circles
'min'	Cluster using a hierarchical clustering algorithm
'max'	where the clustering is determined by the specified
'avg'	distance function $D(X, Y)$
'mean'	
'var'	

Pguptri - Example

Load the sample matrix pencil 'boley2' [Boley'90] with KCF L_3

```
>> [G,H] = pggallery('boley2');
```

where

$$G - \lambda H = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} - \lambda \begin{bmatrix} 1 & -149 & 537 & -27 \\ 1 & -50 & 180 & -9 \\ 1 & -154 & 546 & -25 \end{bmatrix}$$

PGUPTRI - EXAMPLE

Load the sample matrix pencil 'boley2' [Boley'90] with KCF L_3

```
>> [G,H] = pggallery('boley2');
```

where

$$G - \lambda H = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} - \lambda \begin{bmatrix} 1 & -149 & 537 & -27 \\ 1 & -50 & 180 & -9 \\ 1 & -154 & 546 & -25 \end{bmatrix}$$

Results from pguptri ($tol = 10^{-12}$ and $gap = 1000$)

```
>> [A,B,pstr] = pguptri(G,H)
```

```
A =
   -1.0000 -0.0046   0.0008 -0.0004
         0  0.1966   0.8556 -0.4789
         0       0   0.4884  0.8726

B =
     0 -146.8650 -492.6903   635.9285
     0         0    2.3417    -1.5560
     0         0         0     3.7346

pstr =
     R3
```

Pguptri - Problematic Example

The 3×4 sample matrix pencil 'test1' with KCF $L_0 \oplus L_1 \oplus L_2^T \oplus J_1(10^{-4}i)$ has ill-conditioned matrices G and H, $O(\text{cond}(G)) = O(\text{cond}(H)) = 10^{16}$

pguptri fails:

```
>> [A,B,P,Q,dG,dH,pstr] = pguptri(G,H);
```

```
Warning: The rank is increased by 1 with a total perturbation
  of 1.000e-12.
Warning: Failed to separate the right singular and the zero
  eigenvalue blocks.
Warning: Returned order RZ-F-I-L, where RZ is not separated.
```

returns the KCF $L_0 \oplus J_3(0)$

Pguptri - Problematic Example

The 3×4 sample matrix pencil 'test1' with KCF $L_0 \oplus L_1 \oplus L_2^T \oplus J_1(10^{-4}i)$ has ill-conditioned matrices G and H, $O(\text{cond}(G)) = O(\text{cond}(H)) = 10^{16}$

However,
the correct KCF is computed if the matrices are swapped in the computation

```
>> pstr = pguptri(G,H,'ifz')
pstr =
     R1 + R0 + L0 + J1(-3.7259e-17 + 1.0000e-04i)
```

Download and Contact

MCS Toolbox v0.6 and StratiGraph v4.0 can be downloaded from
`https://www.umu.se/en/stratigraph-mcs/`

MCS Toolbox v0.7 including pguptri is available after summer
or send an email to stefanj@cs.umu.se