Equalizing what should be equal
–
Solving the even eigenvalue problem by transforming an even URV form to even Schur form

Christian Schröder
TU Berlin
Research Center MATHEON

Conference on
Computational Methods with Applications

Harrachov, 21 August 2007

The Even Eigenvalue Problem

- special case of generalized eigenvalue problem

$$Mx = \lambda Nx$$

$M = M^T$ symmetric , $N = -N^T$ skew symmetric

- $M, N \in \mathbb{C}^{n,n}$, dense, $(\cdot)^T$ denotes the complex transpose ($*$-case, real case are similar, but different)
- is called: *even* eigenvalue problem, as

$$P(\lambda) := \lambda N - M = P(-\lambda)^T$$

- related to Hamiltonian eigenvalue problem $\Rightarrow$ similar methods for this talks topic in particular: [Wat06][1]

[1] David Watkins, *On the reduction of a Hamiltonian matrix to Hamiltonian Schur form*, Electron. Trans. Numer. Anal. 23, 2006

Application: The LQ optimal control problem

- dynamic system (matrices E, A, B given):

$$E\dot{x}(t) = Ax(t) + Bu(t), \quad x(0) = x_0$$

  chosing $u(t)$ determines $x(t)$

- problem: chose $u(t)$ which minimizes $\int_0^\infty x(t)^T Qx(t) + 2x(t)^T Su(t) + u(t)^T Ru(t) dt$

Application: The LQ optimal control problem

- dynamic system (matrices E, A, B given):

$$E\dot{x}(t) = Ax(t) + Bu(t), \quad x(0) = x_0$$

  chosing $u(t)$ determines $x(t)$

- problem: chose $u(t)$ which minimizes $\int_0^\infty x(t)^T Qx(t) + 2x(t)^T Su(t) + u(t)^T Ru(t) dt$
- yields eigenvalue problem for

$$\lambda \underbrace{\begin{bmatrix} 0 & -E & 0 \\ E^T & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}}_{N=-N^T} + \underbrace{\begin{bmatrix} 0 & A & B \\ A^T & Q & S \\ B^T & S^T & R \end{bmatrix}}_{M=M^T}$$

- needed: deflating subspace for eigenvalues with negative real part

Why not use standard algorithms?

- system features spectral symmetry:

$$Mx = \lambda N x \quad \overset{(\cdot)^T}{\Longleftrightarrow} \quad x^T M = (-\lambda) x^T N$$

  i.e., also $-\lambda$ is eigenvalue $\Rightarrow$ pairs $\pm\lambda$

- structure is important for applications (positive/negative real part)

- general algorithms (like the QZ algorithm) destroy this eigenvalue pairing due to rounding errors

Special case: skew triangular

- If M, N are skew triangular, i.e., $m_{ij} = 0$ whenever $i + j \leq n$,

$$M = \triangle, \quad N = \triangle,$$

- then

$$Me_1 = m_{n1} e_n, \quad Ne_1 = n_{n1} e_n$$

so, e_1 is eigenvector, e_n is image vector, $\frac{m_{n1}}{n_{n1}}$ is eigenvalue,

Special case: skew triangular

- If M, N are skew triangular, i.e., $m_{ij} = 0$ whenever $i + j \leq n$,

$$M = $$  $$,\quad N = $$  $$,$$

- then

$$Me_1 = m_{n1}e_n, \quad Ne_1 = n_{n1}e_n$$

  so, e_1 is eigenvector, e_n is image vector, $\frac{m_{n1}}{n_{n1}}$ is eigenvalue,

- and more general
  $[e_1, e_2, ..., e_k]$ span right deflating subspace,
  $[e_{n-k+1}, ..., e_n]$ span left deflating subspace,
  $\frac{m_{n-i+1,i}}{n_{n-i+1,i}}$ are eigenvalues $(i = 1, \ldots, k)$.

- So, our problem is already solved.

- What if M, N are not skew triangular?

What if M, N are not skew triangular?

- Answer: make them,

[2] C. Schröder, *URV decomposition based structured methods for palindromic and even eigenvalue problems*, MATHEON Preprint 375, 2007

What if M, N are not skew triangular?

- Answer: make them, by finding unitary Q (i.e., $Q^*Q = I$) such that

  $$Q^T M Q = \triangle, \quad Q^T N Q = \triangle.$$

  Called *even Schur form*; Existence: always; Algorithm: many, but no completely satisfying one

[2] C. Schröder, *URV decomposition based structured methods for palindromic and even eigenvalue problems*, MATHEON Preprint 375, 2007

What if M, N are not skew triangular?

- Answer: make them, by finding unitary Q (i.e., $Q^*Q = I$) such that

$$Q^T M Q = \triangle, \quad Q^T N Q = \triangle.$$

  Called *even Schur form*; Existence: always; Algorithm: many, but no completely satisfying one

- What we can compute: unitary U, V such that

$$U^T N U = \triangle, \quad \underbrace{U^T M V}_{R} = \triangle, \quad V^T N V = \triangle$$

  called *even URV form* (as $M = \bar{U} R V^*$); algorithm: [Sch07][2]; provides eigenvalues, eigenvectors, not subspaces

[2] C. Schröder, *URV decomposition based structured methods for palindromic and even eigenvalue problems*, MATHEON Preprint 375, 2007

What if M, N are not skew triangular?

- Answer: make them, by finding unitary Q (i.e., $Q^*Q = I$) such that

$$Q^TMQ = \lrtriangle, \quad Q^TNQ = \lrtriangle.$$

  Called *even Schur form*; Existence: always; Algorithm: many, but no completely satisfying one
- What we can compute: unitary U, V such that

$$U^TNU = \lrtriangle, \quad \underbrace{U^TMV}_{R} = \lrtriangle, \quad V^TNV = \lrtriangle$$

  called *even URV form* (as $M = \bar{U}RV^*$); algorithm: [Sch07][2]; provides eigenvalues, eigenvectors, not subspaces
- note reduces to even Schur form if $U = V$

[2]C. Schröder, *URV decomposition based structured methods for palindromic and even eigenvalue problems*, MATHEON Preprint 375, 2007

What if M, N are not skew triangular?

- Answer: make them, by finding unitary Q (i.e., $Q^*Q = I$) such that

$$Q^T M Q = \triangle, \quad Q^T N Q = \triangle.$$

Called *even Schur form*; Existence: always; Algorithm: many, but no completely satisfying one

- What we can compute: unitary U, V such that

$$U^T N U = \triangle, \quad \underbrace{U^T M V}_{R} = \triangle, \quad V^T N V = \triangle$$

called *even URV form* (as $M = \bar{U} R V^*$); algorithm: [Sch07][2]; provides eigenvalues, eigenvectors, not subspaces

- note reduces to even Schur form if $U = V$
- Idea: transform a URV form to Schur form $\Rightarrow$ Equalizing what should be equal

[2] C. Schröder, *URV decomposition based structured methods for palindromic and even eigenvalue problems*, MATHEON Preprint 375, 2007

Today: first step

- Given an even URV form

$$U^TNU = T = \text{◿}, \quad U^TMV = R = \text{◿}, \quad V^TNV = P = \text{◿}$$

modify U, V to $\tilde{U} = U\Delta_U$, $\tilde{V} = V\Delta_V$

$$\tilde{U}^TN\tilde{U} = \tilde{T} \qquad , \quad \tilde{U}^TM\tilde{V} = \tilde{R} \qquad , \quad \tilde{V}^TN\tilde{V} = \tilde{P} \qquad ,$$

Today: first step

- Given an even URV form

$$U^T N U = T = \triangle, \quad U^T M V = R = \triangle, \quad V^T N V = P = \triangle$$

modify U, V to $\tilde{U} = U\Delta_U$, $\tilde{V} = V\Delta_V$

$$\tilde{U}^T N \tilde{U} = \tilde{T} \qquad , \quad \tilde{U}^T M \tilde{V} = \tilde{R} \qquad , \quad \tilde{V}^T N \tilde{V} = \tilde{P} \qquad ,$$

such that 1) the first and last columns of $\tilde{U}$ and $\tilde{V}$ coincide

$$\tilde{u}_1 = \tilde{v}_1, \quad \tilde{u}_n = \tilde{v}_n,$$

2) $\tilde{T}, \tilde{R}, \tilde{P}$ are still skew triangular.

Today: first step

- Given an even URV form

$$U^T N U = T = \lrtriangle, \quad U^T M V = R = \lrtriangle, \quad V^T N V = P = \lrtriangle$$

modify U, V to $\tilde{U} = U\Delta_U$, $\tilde{V} = V\Delta_V$

$$\tilde{U}^T N \tilde{U} = \tilde{T} \qquad , \quad \tilde{U}^T M \tilde{V} = \tilde{R} \qquad , \quad \tilde{V}^T N \tilde{V} = \tilde{P} \qquad ,$$

such that 1) the first and last columns of $\tilde{U}$ and $\tilde{V}$ coincide

$$\tilde{u}_1 = \tilde{v}_1, \quad \tilde{u}_n = \tilde{v}_n,$$

2) $\tilde{T}, \tilde{R}, \tilde{P}$ are still skew triangular.

- remaining columns can be treated recursively
- Lets concentrate on first goal.

Goal 1: Equalizing first/last column of U, V

Note: $\tilde{u}_1 = \tilde{v}_1$ must be eigenvector, $\bar{\tilde{u}}_n = \bar{\tilde{v}}_n$ must be image vector.
Procedure: obtain eigen/imagevector x, y,

then chose Δ_U, Δ_V such that $\tilde{u}_1 = \tilde{v}_1 = c_1 \cdot x, \quad \bar{\tilde{u}}_n = \bar{\tilde{v}}_n = c_2 \cdot y$

Goal 1 achived,

Goal 1: Equalizing first/last column of U, V

Note: $\tilde{u}_1 = \tilde{v}_1$ must be eigenvector, $\bar{\tilde{u}}_n = \bar{\tilde{v}}_n$ must be image vector.
Procedure: obtain eigen/imagevector x, y, from URV form

$$\begin{aligned} M[u_1, v_1] &= [\bar{u}_n, \bar{v}_n] \begin{bmatrix} 0 & r_{n1} \\ r_{1n} & 0 \end{bmatrix} \\ N[u_1, v_1] &= [\bar{u}_n, \bar{v}_n] \begin{bmatrix} t_{n1} & 0 \\ 0 & p_{n1} \end{bmatrix} \end{aligned}$$

$\Rightarrow \mathrm{span}(u_1, v_1)$ right deflating subspace
$\Rightarrow \mathrm{span}(\bar{u}_n, \bar{v}_n)$ left deflating subspace of (M, N)
$\rightsquigarrow$ contain x, y with $Mx = \alpha y, \quad Nx = \beta y$.
then chose Δ_U, Δ_V such that $\tilde{u}_1 = \tilde{v}_1 = c_1 \cdot x, \quad \bar{\tilde{u}}_n = \bar{\tilde{v}}_n = c_2 \cdot y$

Goal 1 achived,

Goal 1: Equalizing first/last column of U, V

Note: $\tilde{u}_1 = \tilde{v}_1$ must be eigenvector, $\bar{\tilde{u}}_n = \bar{\tilde{v}}_n$ must be image vector.
Procedure: obtain eigen/imagevector x, y, from URV form

$$\begin{aligned} M[u_1, v_1] &= [\bar{u}_n, \bar{v}_n] \begin{bmatrix} 0 & r_{n1} \\ r_{1n} & 0 \end{bmatrix} \\ N[u_1, v_1] &= [\bar{u}_n, \bar{v}_n] \begin{bmatrix} t_{n1} & 0 \\ 0 & p_{n1} \end{bmatrix} \end{aligned}$$

$\Rightarrow \mathrm{span}(u_1, v_1)$ right deflating subspace
$\Rightarrow \mathrm{span}(\bar{u}_n, \bar{v}_n)$ left deflating subspace of (M, N)
$\rightsquigarrow$ contain x, y with $Mx = \alpha y, \quad Nx = \beta y$.
then chose Δ_U, Δ_V such that $\tilde{u}_1 = \tilde{v}_1 = c_1 \cdot x, \quad \bar{\tilde{u}}_n = \bar{\tilde{v}}_n = c_2 \cdot y$
this implys that

$$\Delta_U e_1 = x_u := U^* x, \quad \Delta_U e_n = y_u := U^T y$$

$\Rightarrow$ choose Δ_U as series of Givens rotations that transform x_u to e_1 and y_u to e_n, (same for Δ_V)
Goal 1 achived,

Goal 1: Equalizing first/last column of U, V

Note: $\tilde{u}_1 = \tilde{v}_1$ must be eigenvector, $\bar{\tilde{u}}_n = \bar{\tilde{v}}_n$ must be image vector.
Procedure: obtain eigen/imagevector x, y, from URV form

$$
\begin{aligned}
M[u_1, v_1] &= [\bar{u}_n, \bar{v}_n] \begin{bmatrix} 0 & r_{n1} \\ r_{1n} & 0 \end{bmatrix} \\
N[u_1, v_1] &= [\bar{u}_n, \bar{v}_n] \begin{bmatrix} t_{n1} & 0 \\ 0 & p_{n1} \end{bmatrix}
\end{aligned}
$$

$\Rightarrow \operatorname{span}(u_1, v_1)$ right deflating subspace
$\Rightarrow \operatorname{span}(\bar{u}_n, \bar{v}_n)$ left deflating subspace of (M, N)
$\rightsquigarrow$ contain x, y with $Mx = \alpha y, \quad Nx = \beta y$.
then chose Δ_U, Δ_V such that $\tilde{u}_1 = \tilde{v}_1 = c_1 \cdot x, \quad \bar{\tilde{u}}_n = \bar{\tilde{v}}_n = c_2 \cdot y$
this implys that

$$
\Delta_U e_1 = x_u := U^* x, \quad \Delta_U e_n = y_u := U^T y
$$

$\Rightarrow$ choose Δ_U as series of Givens rotations that transform x_u to e_1 and y_u to e_n, (same for Δ_V)
Goal 1 achived, what about goal 2?

Goal 2: Invariance of URV form

To understand, why R, T, P stay skew triangular, the following relation is essential

$$\begin{array}{rcl} Nx & = & \beta y \\ U^T NUU^* x & = & U^T y \\ Tx_u & = & y_u \\ \Delta_U^T T \Delta_U \Delta_U^* x_u & = & \Delta_U^T y_u \end{array} \qquad (1)$$

So, (1) stays valid under an update of form

$$\begin{array}{rcl} x_u & \leftarrow & \Delta_U^* x_u \\ y_u & \leftarrow & \Delta_U^T y_u \\ T & \leftarrow & \Delta_U^T T \Delta_U \end{array}$$

for any unitary Δ_U.

A 6-by-6 example

Given $M, N \in \mathbb{C}^{6,6}$
we compute URV form, and x_u, x_v, y_u, y_v

$$T, R, P, [x_u, y_u], [x_v, y_v] =$$

$$\begin{bmatrix} & & & & & \mathbf{x} \\ & & & & \mathbf{x} & x \\ & & & \mathbf{x} & x & x \\ & & \mathbf{x} & 0 & x & x \\ & \mathbf{x} & x & x & 0 & x \\ \mathbf{x} & x & x & x & x & 0 \end{bmatrix}, \begin{bmatrix} & & & & & \mathbf{x} \\ & & & & \mathbf{x} & x \\ & & & \mathbf{x} & x & x \\ & & \mathbf{x} & x & x & x \\ & \mathbf{x} & x & x & x & x \\ \mathbf{x} & x & x & x & x & x \end{bmatrix}, \begin{bmatrix} & & & & & \mathbf{x} \\ & & & & \mathbf{x} & x \\ & & & \mathbf{x} & x & x \\ & & \mathbf{x} & 0 & x & x \\ & \mathbf{x} & x & x & 0 & x \\ \mathbf{x} & x & x & x & x & 0 \end{bmatrix}, \begin{bmatrix} x & \mathbf{x} \\ x & x \\ x & x \\ x & x \\ x & x \\ \mathbf{x} & x \end{bmatrix}, \begin{bmatrix} x & \mathbf{x} \\ x & x \\ x & x \\ x & x \\ x & x \\ \mathbf{x} & x \end{bmatrix}.$$

assumption 1: M, N nonsingular $\Rightarrow$ skew diagonal entries of T, R, P nonzero, α, β non-zero
assumption 2: last entries of x_u, x_v and first entries of y_u, y_v are nonzero
goal: $x_u, x_v \longrightarrow e_1$ $y_u, y_v \longrightarrow e_6$.

A 6-by-6 example

$$T, R, P, [x_u, y_u], [x_v, y_v] =$$

$$\begin{bmatrix} & & & & & x \\ & & & & x & x \\ & & & x & x & x \\ & & x & 0 & x & x \\ & x & x & x & 0 & x \\ x & x & x & x & x & 0 \end{bmatrix}, \begin{bmatrix} & & & & & x \\ & & & & x & x \\ & & & x & x & x \\ & & x & x & x & x \\ & x & x & x & x & x \\ x & x & x & x & x & x \end{bmatrix}, \begin{bmatrix} & & & & & x \\ & & & & x & x \\ & & & x & x & x \\ & & x & 0 & x & x \\ & x & x & x & 0 & x \\ x & x & x & x & x & 0 \end{bmatrix}, \begin{bmatrix} x & x \\ x & x \\ x & x \\ x & x \\ x & x \\ x & x \end{bmatrix}, \begin{bmatrix} x & x \\ x & x \\ x & x \\ x & x \\ x & x \\ x & x \end{bmatrix}.$$

eleminate by Givens rotations: last entries of x_u, x_v,
first entries of y_u, y_v

A 6-by-6 example

$$T, R, P, [x_u, y_u], [x_v, y_v] =$$

$$\begin{bmatrix} & & & & ? & y \\ & & & & y & y \\ & & & x & y & y \\ & & x & 0 & y & y \\ ? & y & y & y & 0 & y \\ y & y & y & y & y & 0 \end{bmatrix}, \begin{bmatrix} & & & & ? & y \\ & & & & y & y \\ & & & x & y & y \\ & & x & x & y & y \\ ? & y & y & y & y & y \\ y & y & y & y & y & y \end{bmatrix}, \begin{bmatrix} & & & & ? & y \\ & & & & y & y \\ & & & x & y & y \\ & & x & 0 & y & y \\ ? & y & y & y & 0 & y \\ y & y & y & y & y & 0 \end{bmatrix}, \begin{bmatrix} y & \mathbf{0} \\ y & y \\ x & x \\ x & x \\ y & y \\ \mathbf{0} & y \end{bmatrix}, \begin{bmatrix} y & \mathbf{0} \\ y & y \\ x & x \\ x & x \\ y & y \\ \mathbf{0} & y \end{bmatrix}$$

fill-in at $(1,5)$ and $(5,1)$ in R, T, P !!!!

A 6-by-6 example

$$T, R, P, [x_u, y_u], [x_v, y_v] =$$

$$\begin{bmatrix} & & & & ? & y \\ & & & & y & y \\ & & & x & y & y \\ & & x & 0 & y & y \\ ? & y & y & y & 0 & y \\ y & y & y & y & y & 0 \end{bmatrix}, \begin{bmatrix} & & & & ? & y \\ & & & & y & y \\ & & & x & y & y \\ & & x & x & y & y \\ ? & y & y & y & y & y \\ y & y & y & y & y & y \end{bmatrix}, \begin{bmatrix} & & & & ? & y \\ & & & & y & y \\ & & & x & y & y \\ & & x & 0 & y & y \\ ? & y & y & y & 0 & y \\ y & y & y & y & y & 0 \end{bmatrix}, \begin{bmatrix} y & \mathbf{0} \\ y & y \\ x & x \\ x & x \\ y & y \\ \mathbf{0} & y \end{bmatrix}, \begin{bmatrix} y & \mathbf{0} \\ y & y \\ x & x \\ x & x \\ y & y \\ \mathbf{0} & y \end{bmatrix}$$

fill-in at $(1, 5)$ and $(5, 1)$ in R, T, P !!!! really?
remember, $Tx_u = \beta y_u$ still holds.
first row $\longrightarrow t_{1,5} \cdot x_{u,5} = \beta y_{u,1} = 0$, so $t_{1,5} = 0 = -t_{5,1}$.
Simillar for other question marks

A 6-by-6 example

$$T, R, P, [x_u, y_u], [x_v, y_v] =$$

$$\begin{bmatrix} & & & & ? & y \\ & & & & y & y \\ & & & x & y & y \\ & & x & 0 & y & y \\ ? & y & y & y & 0 & y \\ y & y & y & y & y & 0 \end{bmatrix}, \begin{bmatrix} & & & & ? & y \\ & & & & y & y \\ & & & x & y & y \\ & & x & x & y & y \\ ? & y & y & y & y & y \\ y & y & y & y & y & y \end{bmatrix}, \begin{bmatrix} & & & & ? & y \\ & & & & y & y \\ & & & x & y & y \\ & & x & 0 & y & y \\ ? & y & y & y & 0 & y \\ y & y & y & y & y & 0 \end{bmatrix}, \begin{bmatrix} y & \mathbf{0} \\ y & y \\ x & x \\ x & x \\ y & y \\ \mathbf{0} & y \end{bmatrix}, \begin{bmatrix} y & \mathbf{0} \\ y & y \\ x & x \\ x & x \\ y & y \\ \mathbf{0} & y \end{bmatrix}$$

fill-in at $(1,5)$ and $(5,1)$ in R, T, P !!!! really?
remember, $Tx_u = \beta y_u$ still holds.
first row $\rightarrow t_{1,5} \cdot x_{u,5} = \beta y_{u,1} = 0$, so $t_{1,5} = 0 = -t_{5,1}$.
Simillar for other question marks

next, eleminate by Givens rotations: second last entries of x_u, x_v,
second entries of y_u, y_v

A 6-by-6 example

$$T, R, P, [x_u, y_u], [x_v, y_v] =$$

$$\begin{bmatrix} & & & & & x \\ & & & ? & y & y \\ & & & y & y & y \\ & ? & y & 0 & y & y \\ & y & y & y & 0 & y \\ x & y & y & y & y & 0 \end{bmatrix}, \begin{bmatrix} & & & & & x \\ & & & ? & y & y \\ & & & y & y & y \\ & ? & y & y & y & y \\ & y & y & y & y & y \\ x & y & y & y & y & x \end{bmatrix}, \begin{bmatrix} & & & & & x \\ & & & ? & y & y \\ & & & y & y & y \\ & ? & y & 0 & y & y \\ & y & y & y & 0 & y \\ x & y & y & y & y & 0 \end{bmatrix}, \begin{bmatrix} x & 0 \\ y & \mathbf{0} \\ y & y \\ y & y \\ \mathbf{0} & y \\ 0 & x \end{bmatrix}, \begin{bmatrix} x & 0 \\ y & \mathbf{0} \\ y & y \\ y & y \\ \mathbf{0} & y \\ 0 & x \end{bmatrix}$$

Again, fill-in in T, R, P.

A 6-by-6 example

$$T, R, P, [x_u, y_u], [x_v, y_v] =$$

$$\begin{bmatrix} & & & & & x \\ & & & ? & y & y \\ & & & y & y & y \\ & ? & y & 0 & y & y \\ & y & y & y & 0 & y \\ x & y & y & y & y & 0 \end{bmatrix}, \begin{bmatrix} & & & & & x \\ & & & ? & y & y \\ & & & y & y & y \\ & ? & y & y & y & y \\ & y & y & y & y & y \\ x & y & y & y & y & x \end{bmatrix}, \begin{bmatrix} & & & & & x \\ & & & ? & y & y \\ & & & y & y & y \\ & ? & y & 0 & y & y \\ & y & y & y & 0 & y \\ x & y & y & y & y & 0 \end{bmatrix}, \begin{bmatrix} x & 0 \\ y & \mathbf{0} \\ y & y \\ y & y \\ \mathbf{0} & y \\ 0 & x \end{bmatrix}, \begin{bmatrix} x & 0 \\ y & \mathbf{0} \\ y & y \\ y & y \\ \mathbf{0} & y \\ 0 & x \end{bmatrix}$$

Again, fill-in in T, R, P.

again, fake!!!

second row of $Tx_u = y_u$: $\quad T_{2,4}x_{u,4} = 0$

A 6-by-6 example

$$T, R, P, [x_u, y_u], [x_v, y_v] =$$

$$\begin{bmatrix} & & & & & x \\ & & & ? & y & y \\ & & & y & y & y \\ & ? & y & 0 & y & y \\ & y & y & y & 0 & y \\ x & y & y & y & y & 0 \end{bmatrix}, \begin{bmatrix} & & & & & x \\ & & & ? & y & y \\ & & & y & y & y \\ & ? & y & y & y & y \\ & y & y & y & y & y \\ x & y & y & y & y & x \end{bmatrix}, \begin{bmatrix} & & & & & x \\ & & & ? & y & y \\ & & & y & y & y \\ & ? & y & 0 & y & y \\ & y & y & y & 0 & y \\ x & y & y & y & y & 0 \end{bmatrix}, \begin{bmatrix} x & 0 \\ y & \mathbf{0} \\ y & y \\ y & y \\ \mathbf{0} & y \\ 0 & x \end{bmatrix}, \begin{bmatrix} x & 0 \\ y & \mathbf{0} \\ y & y \\ y & y \\ \mathbf{0} & y \\ 0 & x \end{bmatrix}$$

Again, fill-in in T, R, P.
again, fake!!!
second row of $Tx_u = y_u$: $T_{2,4}x_{u,4} = 0$

next, eleminate by Givens rotations: third entries of y_u, y_v

A 6-by-6 example

$$T, R, P, [x_u, y_u], [x_v, y_v] =$$

$$\begin{bmatrix} & & & & & x \\ & & & & x & x \\ & & 0 & y & y & y \\ & & y & 0 & y & y \\ & x & y & y & 0 & x \\ x & x & y & y & x & 0 \end{bmatrix}, \begin{bmatrix} & & & & & x \\ & & & & x & x \\ & & ? & y & y & y \\ & & y & y & y & y \\ & x & y & y & x & x \\ x & x & y & y & x & x \end{bmatrix}, \begin{bmatrix} & & & & & x \\ & & & & x & x \\ & & 0 & y & y & y \\ & & y & 0 & y & y \\ & x & y & y & 0 & x \\ x & x & y & y & x & 0 \end{bmatrix}, \begin{bmatrix} x & 0 \\ x & 0 \\ y & \mathbf{0} \\ ? & y \\ 0 & x \\ 0 & x \end{bmatrix}, \begin{bmatrix} x & 0 \\ x & 0 \\ y & \mathbf{0} \\ ? & y \\ 0 & x \\ 0 & x \end{bmatrix}$$

- fake non-zeros in x_u, x_v, because $x_u^T y_u = 0$
- no fill-in in T, P, because skew symmetry
- fake fill-in in R

A 6-by-6 example

$$T, R, P, [x_u, y_u], [x_v, y_v] =$$

$$\begin{bmatrix} & & & & & x \\ & & & & x & x \\ & & 0 & y & y & y \\ & & y & 0 & y & y \\ & x & y & y & 0 & x \\ x & x & y & y & x & 0 \end{bmatrix}, \begin{bmatrix} & & & & & x \\ & & & & x & x \\ & & ? & y & y & y \\ & & y & y & y & y \\ & x & y & y & x & x \\ x & x & y & y & x & x \end{bmatrix}, \begin{bmatrix} & & & & & x \\ & & & & x & x \\ & & 0 & y & y & y \\ & & y & 0 & y & y \\ & x & y & y & 0 & x \\ x & x & y & y & x & 0 \end{bmatrix}, \begin{bmatrix} x & 0 \\ x & 0 \\ y & \mathbf{0} \\ ? & y \\ 0 & x \\ 0 & x \end{bmatrix}, \begin{bmatrix} x & 0 \\ x & 0 \\ y & \mathbf{0} \\ ? & y \\ 0 & x \\ 0 & x \end{bmatrix}$$

- fake non-zeros in x_u, x_v, because $x_u^T y_u = 0$
- no fill-in in T, P, because skew symmetry
- fake fill-in in R

remaining steps as before

A 6-by-6 example

$$T, R, P, [x_u, y_u], [x_v, y_v] =$$

$$\begin{bmatrix} & & & & & x \\ & & & ? & y & y \\ & & & y & y & y \\ & ? & y & 0 & y & y \\ & y & y & y & 0 & y \\ x & y & y & y & y & 0 \end{bmatrix}, \begin{bmatrix} & & & & & x \\ & & & ? & y & y \\ & & & y & y & y \\ & ? & y & y & y & y \\ & y & y & y & y & y \\ x & y & y & y & y & x \end{bmatrix}, \begin{bmatrix} & & & & & x \\ & & & ? & y & y \\ & & & y & y & y \\ & ? & y & 0 & y & y \\ & y & y & y & 0 & y \\ x & y & y & y & y & 0 \end{bmatrix}, \begin{bmatrix} x & 0 \\ y & 0 \\ \mathbf{0} & 0 \\ 0 & \mathbf{0} \\ 0 & y \\ 0 & x \end{bmatrix}, \begin{bmatrix} x & 0 \\ y & 0 \\ \mathbf{0} & 0 \\ 0 & \mathbf{0} \\ 0 & y \\ 0 & x \end{bmatrix}$$

and finally

A 6-by-6 example

$$T, R, P, [x_u, y_u], [x_v, y_v] =$$

$$\begin{bmatrix} & & & & ? & y \\ & & & & y & y \\ & & & x & y & y \\ & & x & 0 & y & y \\ ? & y & y & y & 0 & y \\ y & y & y & y & y & 0 \end{bmatrix}, \begin{bmatrix} & & & & ? & y \\ & & & & y & y \\ & & & x & y & y \\ & & x & x & y & y \\ ? & y & y & y & y & y \\ y & y & y & y & y & y \end{bmatrix}, \begin{bmatrix} & & & & ? & y \\ & & & & y & y \\ & & & x & y & y \\ & & x & 0 & y & y \\ ? & y & y & y & 0 & y \\ y & y & y & y & y & 0 \end{bmatrix}, \begin{bmatrix} y & 0 \\ \mathbf{0} & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & \mathbf{0} \\ 0 & y \end{bmatrix}, \begin{bmatrix} y & 0 \\ \mathbf{0} & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & \mathbf{0} \\ 0 & y \end{bmatrix}$$

As before, $? = 0$.

A 6-by-6 example

$T, R, P, [x_u, y_u], [x_v, y_v] =$

$$\begin{bmatrix} & & & & ? & y \\ & & & & y & y \\ & & & x & y & y \\ & & x & 0 & y & y \\ ? & y & y & y & 0 & y \\ y & y & y & y & y & 0 \end{bmatrix}, \begin{bmatrix} & & & & ? & y \\ & & & & y & y \\ & & & x & y & y \\ & & x & x & y & y \\ ? & y & y & y & y & y \\ y & y & y & y & y & y \end{bmatrix}, \begin{bmatrix} & & & & ? & y \\ & & & & y & y \\ & & & x & y & y \\ & & x & 0 & y & y \\ ? & y & y & y & 0 & y \\ y & y & y & y & y & 0 \end{bmatrix}, \begin{bmatrix} y & 0 \\ \mathbf{0} & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & \mathbf{0} \\ 0 & y \end{bmatrix}, \begin{bmatrix} y & 0 \\ \mathbf{0} & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & \mathbf{0} \\ 0 & y \end{bmatrix}$$

As before, $? = 0$.

At this point: $x_u, x_v = const \cdot e_1$, and $y_u, y_v = const \cdot e_6$

$\Rightarrow$ the first columns of U and V are multiples of each other

$\Rightarrow$ after scaling they coincide

$\Rightarrow$ Done!

Summary

- Algorithm to compute even Schur form
- not from scratch, but
- by transforming an even URV form
- preprint soon to come

Summary

- Algorithm to compute even Schur form
- not from scratch, but
- by transforming an even URV form
- preprint soon to come

Comments

- can be extended to non-singular M, N (deflation, adaption)
- can be extended to real arithmetic
  (2-by-2 blocks for conjugate quadruples $\pm\lambda, \pm\bar{\lambda}$)

Summary

- Algorithm to compute even Schur form
- not from scratch, but
- by transforming an even URV form
- preprint soon to come

Comments

- can be extended to non-singular M, N (deflation, adaption)
- can be extended to real arithmetic
  (2-by-2 blocks for conjugate quadruples $\pm\lambda, \pm\bar{\lambda}$)

Thanks for your attention. Any questions? Enjoy the dinner!