On Generalized Schur Algorithms

Jens-Peter M. Zemke
zemke@tu-harburg.de

Institut für Numerische Simulation
Technische Universität Hamburg-Harburg

13.12.2006

TUHH
Technische Universität Hamburg-Harburg

Outline

Classification and normal forms of functions

Outline

Classification and normal forms of functions

- Schur functions
- Jacobi transformation
- Cayley transform
- Carathéodory functions
- Matrix decomposition

Schur algorithm; modern form

- Reformulation of Schur's expansion
- Displacement structure
- Cholesky decomposition
- The Schur algorithm

Outline

Outline

Schur functions

Schur [Schur, 1917/1918] investigated analytic functions bounded by one in modulus in the open unit disc $\mathbb{D}$. These functions are nowadays known as Schur functions.

Schur functions

Schur [Schur, 1917/1918] investigated analytic functions bounded by one in modulus in the open unit disc $\mathbb{D}$. These functions are nowadays known as Schur functions.

Schur denoted the set of all such functions by $\mathfrak{E}$. In modern notation, when $s \in \mathfrak{E}$,

$$s(z) = \sum_{k=0}^{\infty} \mathfrak{s}_k z^k, \qquad |s(z)| \leqslant 1, \quad |z| < 1. \tag{1}$$

Schur functions

Schur [Schur, 1917/1918] investigated analytic functions bounded by one in modulus in the open unit disc $\mathbb{D}$. These functions are nowadays known as Schur functions.

Schur denoted the set of all such functions by $\mathfrak{E}$. In modern notation, when $s \in \mathfrak{E}$,

$$s(z) = \sum_{k=0}^{\infty} \mathfrak{s}_k z^k, \qquad |s(z)| \leqslant 1, \quad |z| < 1. \tag{1}$$

In network theory, Schur functions are known as scattering functions.

Schur functions

Schur [Schur, 1917/1918] investigated analytic functions bounded by one in modulus in the open unit disc $\mathbb{D}$. These functions are nowadays known as Schur functions.

Schur denoted the set of all such functions by $\mathfrak{E}$. In modern notation, when $s \in \mathfrak{E}$,

$$s(z) = \sum_{k=0}^{\infty} \mathfrak{s}_k z^k, \qquad |s(z)| \leqslant 1, \quad |z| < 1. \tag{1}$$

In network theory, Schur functions are known as scattering functions.

It turns out that the coefficients $\mathfrak{s}_k$ of the power series are not that useful when investigating Schur functions. Instead, the so-called Schur coefficients better describe their properties, and thus, play the dominant rôle.

Expansion of Schur functions

Schur used that for $\alpha \in \mathbb{C}$ with $|\alpha| < 1$ the "linear" transformation (Blaschke factor, no normalization; Moebius transformation)

$$\mathfrak{B}(z) = \frac{z - \alpha}{1 - \overline{\alpha} z} \tag{2}$$

maps $\mathbb{D}$ to itself.

Expansion of Schur functions

Schur used that for $\alpha \in \mathbb{C}$ with $|\alpha| < 1$ the "linear" transformation (Blaschke factor, no normalization; Moebius transformation)

$$\mathfrak{B}(z) = \frac{z - \alpha}{1 - \overline{\alpha} z} \tag{2}$$

maps $\mathbb{D}$ to itself and by Schwarz's Lemma (set $f(z) = zs(z)$)

$$\{|s(z)| \leqslant 1, \quad |z| < 1\} \quad \Leftrightarrow \quad \{|zs(z)| \leqslant 1, \quad |z| < 1\}. \tag{3}$$

Expansion of Schur functions

Schur used that for $\alpha \in \mathbb{C}$ with $|\alpha| < 1$ the “linear” transformation (Blaschke factor, no normalization; Moebius transformation)

$$\mathfrak{B}(z) = \frac{z-\alpha}{1-\overline{\alpha}z} \tag{2}$$

maps $\mathbb{D}$ to itself and by Schwarz’s Lemma (set $f(z) = zs(z)$)

$$\{|s(z)| \leqslant 1, \quad |z| < 1\} \quad \Leftrightarrow \quad \{|zs(z)| \leqslant 1, \quad |z| < 1\}, \tag{3}$$

to propose a recursive expansion (kettenbruchartiger Algorithmus) of a Schur function $s \in \mathfrak{E}$.

Expansion of Schur functions

Schur used that for $\alpha \in \mathbb{C}$ with $|\alpha| < 1$ the "linear" transformation (Blaschke factor, no normalization; Moebius transformation)

$$\mathfrak{B}(z) = \frac{z - \alpha}{1 - \overline{\alpha} z} \tag{2}$$

maps $\mathbb{D}$ to itself and by Schwarz's Lemma (set $f(z) = zs(z)$)

$$\{|s(z)| \leqslant 1, \quad |z| < 1\} \quad \Leftrightarrow \quad \{|zs(z)| \leqslant 1, \quad |z| < 1\}, \tag{3}$$

to propose a recursive expansion (kettenbruchartiger Algorithmus) of a Schur function $s \in \mathfrak{E}$. Set $s_0 = s$ and perform

$$s_{i+1}(z) = \frac{1}{z} \frac{s_i(z) - \gamma_i}{1 - \overline{\gamma}_i s_i(z)}, \qquad \gamma_i = s_i(0). \tag{4}$$

Expansion of Schur functions

Schur used that for $\alpha \in \mathbb{C}$ with $|\alpha| < 1$ the "linear" transformation (Blaschke factor, no normalization; Moebius transformation)

$$\mathfrak{B}(z) = \frac{z-\alpha}{1-\overline{\alpha}z} \tag{2}$$

maps $\mathbb{D}$ to itself and by Schwarz's Lemma (set $f(z) = zs(z)$)

$$\{|s(z)| \leqslant 1, \quad |z| < 1\} \quad \Leftrightarrow \quad \{|zs(z)| \leqslant 1, \quad |z| < 1\}, \tag{3}$$

to propose a recursive expansion (kettenbruchartiger Algorithmus) of a Schur function $s \in \mathfrak{E}$. Set $s_0 = s$ and perform

$$s_{i+1}(z) = \frac{1}{z}\frac{s_i(z)-\gamma_i}{1-\overline{\gamma}_i s_i(z)}, \quad \gamma_i = s_i(0). \tag{4}$$

Then all $\{s_i\}_{i=0}^m$ satisfy $|s_i(z)| \leqslant 1$.

Expansion of Schur functions

Schur used that for $\alpha \in \mathbb{C}$ with $|\alpha| < 1$ the "linear" transformation (Blaschke factor, no normalization; Moebius transformation)

$$\mathfrak{B}(z) = \frac{z - \alpha}{1 - \overline{\alpha} z} \tag{2}$$

maps $\mathbb{D}$ to itself and by Schwarz's Lemma (set $f(z) = zs(z)$)

$$\{|s(z)| \leqslant 1, \quad |z| < 1\} \quad \Leftrightarrow \quad \{|zs(z)| \leqslant 1, \quad |z| < 1\}, \tag{3}$$

to propose a recursive expansion (kettenbruchartiger Algorithmus) of a Schur function $s \in \mathfrak{E}$. Set $s_0 = s$ and perform

$$s_{i+1}(z) = \frac{1}{z} \frac{s_i(z) - \gamma_i}{1 - \overline{\gamma}_i s_i(z)}, \quad \gamma_i = s_i(0). \tag{4}$$

Then all $\{s_i\}_{i=0}^m$ satisfy $|s_i(z)| \leqslant 1$. The constants γ_i are the aforementioned Schur coefficients or reflection coefficients.

Rational expansion of Schur functions

Schur developed recursions for the expansion by writing the function s in terms of a rational function,

$$s(z) = \frac{a(z)}{b(z)} = \frac{\sum_{k=0}^{\infty} a_k z^k}{\sum_{k=0}^{\infty} b_k z^k}, \qquad b_0 \neq 0. \tag{5}$$

Rational expansion of Schur functions

Schur developed recursions for the expansion by writing the function s in terms of a rational function,

$$s(z) = \frac{a(z)}{b(z)} = \frac{\sum_{k=0}^{\infty} a_k z^k}{\sum_{k=0}^{\infty} b_k z^k}, \qquad b_0 \neq 0. \tag{5}$$

Using determinal identities Schur proved that the functions $\{s_i\}_{i=0}^{\infty}$ can be expressed according to

$$s_i(z) = -\frac{D_i(z)}{\Delta_i(z)}. \tag{6}$$

Rational expansion of Schur functions

Schur developed recursions for the expansion by writing the function s in terms of a rational function,

$$s(z) = \frac{a(z)}{b(z)} = \frac{\sum_{k=0}^{\infty} a_k z^k}{\sum_{k=0}^{\infty} b_k z^k}, \qquad b_0 \neq 0. \tag{5}$$

Using determinal identities Schur proved that the functions $\{s_i\}_{i=0}^{\infty}$ can be expressed according to

$$s_i(z) = -\frac{D_i(z)}{\Delta_i(z)}, \tag{6}$$

where he additionally defined analytic functions

$$g_i(z) = \sum_{k=0}^{\infty} a_{i+k} z^k \quad \text{and} \quad h_i(z) = \sum_{k=0}^{\infty} b_{i+k} z^k \tag{7}$$

Rational expansion of Schur functions

Schur developed recursions for the expansion by writing the function s in terms of a rational function,

$$s(z) = \frac{a(z)}{b(z)} = \frac{\sum_{k=0}^{\infty} a_k z^k}{\sum_{k=0}^{\infty} b_k z^k}, \qquad b_0 \neq 0. \tag{5}$$

Using determinal identities Schur proved that the functions $\{s_i\}_{i=0}^{\infty}$ can be expressed according to

$$s_i(z) = -\frac{D_i(z)}{\Delta_i(z)}, \tag{6}$$

where he additionally defined analytic functions

$$g_i(z) = \sum_{k=0}^{\infty} a_{i+k} z^k \quad \text{and} \quad h_i(z) = \sum_{k=0}^{\infty} b_{i+k} z^k \tag{7}$$

and gave determinal expressions for $D_i(z)$ and $\Delta_i(z)$ (the next two pages . . .)

Explicit solution of the recursion

(remember that $g_i(z) = \sum_{k=0}^{\infty} a_{i+k} z^k$ and $h_i(z) = \sum_{k=0}^{\infty} b_{i+k} z^k$)

$$D_i(z) = \begin{vmatrix} 0 & 0 & \cdots & 0 & a_0 & a_1 & \cdots & a_{i-1} & g_i(z) \\ \overline{b}_0 & 0 & \cdots & 0 & 0 & a_0 & \cdots & a_{i-2} & g_{i-1}(z) \\ \overline{b}_1 & \overline{b}_0 & \cdots & 0 & 0 & 0 & \cdots & a_{i-3} & g_{i-2}(z) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \overline{b}_{i-2} & \overline{b}_{i-3} & \cdots & \overline{b}_0 & 0 & 0 & \cdots & a_0 & g_1(z) \\ 0 & 0 & \cdots & 0 & b_0 & b_1 & \cdots & b_{i-1} & h_i(z) \\ \overline{a}_0 & 0 & \cdots & 0 & 0 & b_0 & \cdots & b_{i-2} & h_{i-1}(z) \\ \overline{a}_1 & \overline{a}_0 & \cdots & 0 & 0 & 0 & \cdots & b_{i-3} & h_{i-2}(z) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \overline{a}_{i-2} & \overline{a}_{i-3} & \cdots & \overline{a}_0 & 0 & 0 & \cdots & b_0 & h_1(z) \end{vmatrix} \qquad (8)$$

Explicit solution of the recursion

(remember that $g_i(z) = \sum_{k=0}^{\infty} a_{i+k} z^k$ and $h_i(z) = \sum_{k=0}^{\infty} b_{i+k} z^k$)

$$\Delta_i(z) = \begin{vmatrix} \overline{b}_0 & 0 & \cdots & 0 & a_0 & a_1 & \cdots & a_{i-2} & g_{i-1}(z) \\ \overline{b}_1 & \overline{b}_0 & \cdots & 0 & 0 & a_0 & \cdots & a_{i-3} & g_{i-2}(z) \\ \overline{b}_2 & \overline{b}_1 & \cdots & 0 & 0 & 0 & \cdots & a_{i-4} & g_{i-3}(z) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \overline{b}_{i-1} & \overline{b}_{i-2} & \cdots & \overline{b}_0 & 0 & 0 & \cdots & 0 & g_0(z) \\ \overline{a}_0 & 0 & \cdots & 0 & b_0 & b_1 & \cdots & b_{i-2} & h_{i-1}(z) \\ \overline{a}_1 & \overline{a}_0 & \cdots & 0 & 0 & b_0 & \cdots & b_{i-3} & h_{i-2}(z) \\ \overline{a}_2 & \overline{a}_1 & \cdots & 0 & 0 & 0 & \cdots & b_{i-4} & h_{i-3}(z) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \overline{a}_{i-1} & \overline{a}_{i-2} & \cdots & \overline{a}_0 & 0 & 0 & \cdots & 0 & h_0(z) \end{vmatrix} \tag{9}$$

A Toeplitz reformulation

Following [Toeplitz, 1911] Schur associated matrices to every power series as follows. Let a and b be the power series defined as

$$a(z) = \sum_{k=0}^{\infty} a_k z^k, \quad b(z) = \sum_{k=0}^{\infty} b_k z^k. \tag{10}$$

A Toeplitz reformulation

Following [Toeplitz, 1911] Schur associated matrices to every power series as follows. Let a and b be the power series defined as

$$a(z) = \sum_{k=0}^{\infty} a_k z^k, \quad b(z) = \sum_{k=0}^{\infty} b_k z^k. \tag{10}$$

Then infinite upper triangular Toeplitz matrices A and B associated to a and b are defined by

$$A = \begin{pmatrix} a_0 & a_1 & \cdots \\ & a_0 & \ddots \\ & & \ddots \end{pmatrix}, \quad B = \begin{pmatrix} b_0 & b_1 & \cdots \\ & b_0 & \ddots \\ & & \ddots \end{pmatrix}. \tag{11}$$

A Toeplitz reformulation

Following [Toeplitz, 1911] Schur associated matrices to every power series as follows. Let a and b be the power series defined as

$$a(z) = \sum_{k=0}^{\infty} a_k z^k, \quad b(z) = \sum_{k=0}^{\infty} b_k z^k. \tag{10}$$

Then infinite upper triangular Toeplitz matrices A and B associated to a and b are defined by

$$A = \begin{pmatrix} a_0 & a_1 & \cdots \\ & a_0 & \ddots \\ & & \ddots \end{pmatrix}, \quad B = \begin{pmatrix} b_0 & b_1 & \cdots \\ & b_0 & \ddots \\ & & \ddots \end{pmatrix}, \tag{11}$$

and infinite Hermitean matrices $\mathfrak{A}$ and $\mathfrak{B}$ associated to a and b are defined by $\mathfrak{A} = A^H A$ and $\mathfrak{B} = B^H B$.

The matrix counterpart

The Schur coefficients are given by

$$\gamma_i = s_i(0) \tag{12}$$

The matrix counterpart

The Schur coefficients are given by the fractions

$$\gamma_i = s_i(0) = -\frac{D_i(0)}{\Delta_i(0)} \tag{12}$$

The matrix counterpart

The Schur coefficients are given by the fractions

$$\gamma_i = s_i(0) = -\frac{D_i(0)}{\Delta_i(0)} \equiv -\frac{d_i}{\delta_i}. \tag{12}$$

The matrix counterpart

The Schur coefficients are given by the fractions

$$\gamma_i = s_i(0) = -\frac{D_i(0)}{\Delta_i(0)} \equiv -\frac{d_i}{\delta_i}. \tag{12}$$

Due to the underlying relations, the reflection coefficients satisfy

$$1 - |\gamma_i|^2 = \frac{\delta_{i-1}\delta_{i+1}}{\delta_i^2}, \quad \delta_0 \equiv 1,\ \delta_{-1} \equiv \frac{1}{b_0^2}. \tag{13}$$

The matrix counterpart

The Schur coefficients are given by the fractions

$$\gamma_i = s_i(0) = -\frac{D_i(0)}{\Delta_i(0)} \equiv -\frac{d_i}{\delta_i}. \tag{12}$$

Due to the underlying relations, the reflection coefficients satisfy

$$1 - |\gamma_i|^2 = \frac{\delta_{i-1}\delta_{i+1}}{\delta_i^2}, \quad \delta_0 \equiv 1,\ \delta_{-1} \equiv \frac{1}{b_0^2}. \tag{13}$$

Based on the commutativity of Toeplitz matrices Schur proved that the determinants δ_i are also the $i \times i$ leading principal determinants of the infinite Hermitean matrix (Hermitean form)

$$\mathfrak{H} = \mathfrak{B} - \mathfrak{A}. \tag{14}$$

Outline

Jacobi transformation

[Schur, 1917/1918] remarks on pages 217–218:

> *"Der Übergang [..] entspricht also dem ersten Schritt bei der Jacobischen Transformation der Form* $\mathfrak{H}(x_0, x_1, \ldots, x_\nu)$*."*

Jacobi transformation

[Schur, 1917/1918] remarks on pages 217–218:

> *“Der Übergang [..] entspricht also dem ersten Schritt bei der Jacobischen Transformation der Form $\mathfrak{H}(x_0, x_1, \ldots, x_\nu)$.”*

Schur’s treatment reminds of [Toeplitz, 1907] who cites Jacobi, but remarks:

> *“Allgemein für Bilinearformen wird diese Transformation von Jacobi aufgestellt; für quadratische Formen wird sie schon von Lagrange und Gauß verwendet.”*

Jacobi transformation

[Schur, 1917/1918] remarks on pages 217–218:

> *"Der Übergang [..] entspricht also dem ersten Schritt bei der Jacobischen Transformation der Form $\mathfrak{H}(x_0, x_1, \ldots, x_\nu)$."*

Schur's treatment reminds of [Toeplitz, 1907] who cites Jacobi, but remarks:

> *"Allgemein für Bilinearformen wird diese Transformation von Jacobi aufgestellt; für quadratische Formen wird sie schon von Lagrange und Gauß verwendet."*

[Lev-Ari & Kailath, 1986] state "incorrectly" (i.e., simplified) that the mentioned "Jacobi Transformation" [Jacobi, 1857] is the nested computation of the LDLT decomposition of H (a finite section of $\mathfrak{H}$),

$$H = LDL^H, \tag{15}$$

with L unit diagonal lower triangular and D diagonal.

Jacobi transformation; modern style

In [Kailath & Sayed, 1999] the authors “correct” the decomposition and call “Schur’s” variant of Jacobi’s transformation the Gauß-Schur reduction.

Jacobi transformation; modern style

In [Kailath & Sayed, 1999] the authors “correct” the decomposition and call “Schur’s” variant of Jacobi’s transformation the Gauß-Schur reduction.

In this variant H is decomposed into

$$H = LD^{-1}L^H, \quad l_{ii} = d_{ii}, \tag{16}$$

where L is lower triangular and D is diagonal.

Jacobi transformation; modern style

In [Kailath & Sayed, 1999] the authors “correct” the decomposition and call “Schur’s” variant of Jacobi’s transformation the Gauß-Schur reduction.

In this variant H is decomposed into

$$H = LD^{-1}L^H, \quad l_{ii} = d_{ii}, \tag{16}$$

where L is lower triangular and D is diagonal.

Actually, [Jacobi, 1857] computes an LDMT decomposition using $A^{(0)} = A$ and the iteration

$$A^{(i)} = A^{(i-1)} - \frac{a_{:i}^{(i-1)} a_{i:}^{(i-1)}}{a_{ii}^{(i-1)}}. \tag{17}$$

Jacobi transformation; modern style

In [Kailath & Sayed, 1999] the authors "correct" the decomposition and call "Schur's" variant of Jacobi's transformation the Gauß-Schur reduction.

In this variant H is decomposed into

$$H = LD^{-1}L^H, \quad l_{ii} = d_{ii}, \tag{16}$$

where L is lower triangular and D is diagonal.

Actually, [Jacobi, 1857] computes an LDMT decomposition using $A^{(0)} = A$ and the iteration

$$A^{(i)} = A^{(i-1)} - \frac{a_{:i}^{(i-1)} a_{i:}^{(i-1)}}{a_{ii}^{(i-1)}}. \tag{17}$$

Additionally, the resulting quantities are expressed (as usually those times) in terms of determinants.

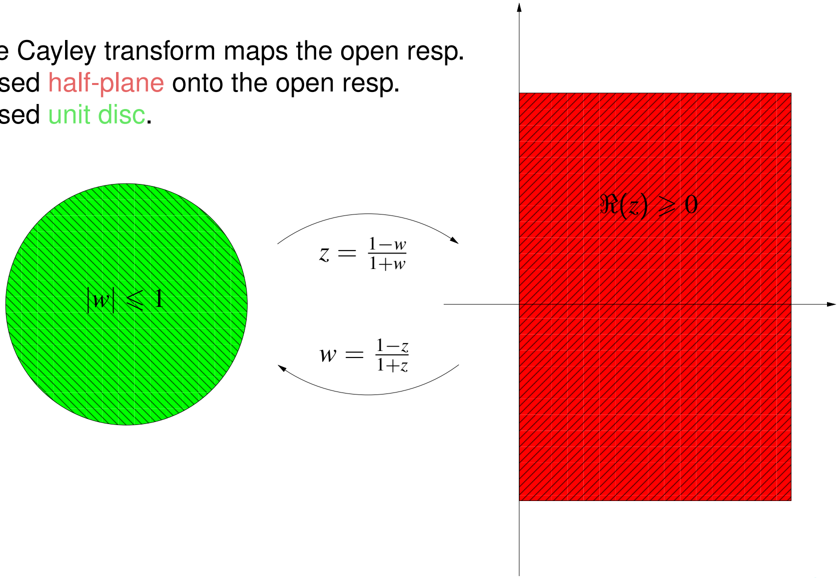
Outline

Cayley transform

The Cayley transform maps the open resp. closed half-plane onto the open resp. closed unit disc.

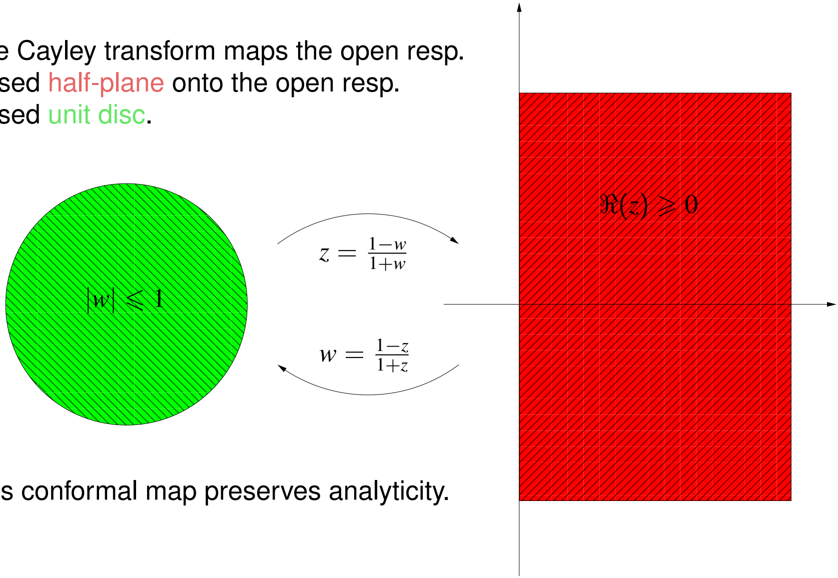
Cayley transform

The Cayley transform maps the open resp. closed half-plane onto the open resp. closed unit disc.



Cayley transform

The Cayley transform maps the open resp. closed half-plane onto the open resp. closed unit disc.



This conformal map preserves analyticity.

Outline

Carathéodory functions

Schur's investigations are closely related to Toeplitz' and Carathéodory's work on functions with positive real part,

$$c(z) = \sum_{k=0}^{\infty} c_k z^k, \qquad \Re(c(z)) > 0, \quad |z| < 1. \tag{18}$$

Carathéodory functions

Schur's investigations are closely related to Toeplitz' and Carathéodory's work on functions with positive real part,

$$c(z) = \sum_{k=0}^{\infty} c_k z^k, \qquad \Re(c(z)) > 0, \quad |z| < 1. \tag{18}$$

These functions are known as Carathéodory functions.

Carathéodory functions

Schur's investigations are closely related to Toeplitz' and Carathéodory's work on functions with positive real part,

$$c(z) = \sum_{k=0}^{\infty} c_k z^k, \qquad \Re(c(z)) > 0, \quad |z| < 1. \tag{18}$$

These functions are known as Carathéodory functions. A function is a Carathéodory function, if and only if "all" Hermitean Toeplitz matrices

$$T_m = \begin{pmatrix} \overline{c}_0 + c_0 & c_1 & \cdots & c_m \\ \overline{c}_1 & \overline{c}_0 + c_0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & c_1 \\ \overline{c}_m & \cdots & \overline{c}_1 & \overline{c}_0 + c_0 \end{pmatrix} \tag{19}$$

are positive definite, $\{T_m > 0\}_{m=0}^n$, $\det(T_m) = 0$, $m > n$.

Schur's proof

Carathéodory functions can be transformed to Schur functions via a Cayley transform that maps the positive real complex (right) half-plane onto the interior of the unit disc,

$$s(z) = \frac{1 - c(z)}{1 + c(z)}. \tag{20}$$

Schur's proof

Carathéodory functions can be transformed to Schur functions via a Cayley transform that maps the positive real complex (right) half-plane onto the interior of the unit disc,

$$s(z) = \frac{1 - c(z)}{1 + c(z)}. \tag{20}$$

If the Carathéodory function is given in rational form,

$$c(z) = \frac{a(z)}{b(z)} = \frac{\sum_{k=0}^{\infty} a_k z^k}{\sum_{k=0}^{\infty} b_k z^k}, \qquad b_0 \neq 0, \tag{21}$$

Schur's proof

Carathéodory functions can be transformed to Schur functions via a Cayley transform that maps the positive real complex (right) half-plane onto the interior of the unit disc,

$$s(z) = \frac{1 - c(z)}{1 + c(z)}. \tag{20}$$

If the Carathéodory function is given in rational form,

$$c(z) = \frac{a(z)}{b(z)} = \frac{\sum_{k=0}^{\infty} a_k z^k}{\sum_{k=0}^{\infty} b_k z^k}, \qquad b_0 \neq 0, \tag{21}$$

then the associated Schur function takes the form

$$s(z) = \frac{b(z) - a(z)}{b(z) + a(z)}. \tag{22}$$

Schur's proof

Since the Cayley transformed Carathéodory function is given by

$$s(z) = \frac{b(z) - a(z)}{b(z) + a(z)}, \tag{23}$$

Schur's proof

Since the Cayley transformed Carathéodory function is given by

$$s(z) = \frac{b(z) - a(z)}{b(z) + a(z)}, \tag{23}$$

the associated Hermitean form is given by

$$\mathfrak{H} = (B + A)^H (B + A) - (B - A)^H (B - A) = 2(B^H A + A^H B), \tag{24}$$

Schur's proof

Since the Cayley transformed Carathéodory function is given by

$$s(z) = \frac{b(z) - a(z)}{b(z) + a(z)}, \tag{23}$$

the associated Hermitean form is given by

$$\mathfrak{H} = (B + A)^H (B + A) - (B - A)^H (B - A) = 2(B^H A + A^H B), \tag{24}$$

and taking $b(z) \equiv 1$ gives a multiple of the aforementioned Toeplitz matrix,

$$\mathfrak{H} = 2(B^H A + A^H B) = 2(I^H A + A^H I) \tag{25}$$

$$= 2(A + A^H) = 2(C + C^H). \tag{26}$$

Outline

Utilizing these relations

The test whether a function is a Schur function is based on the Schur coefficients (reflection coefficients).

Utilizing these relations

The test whether a function is a Schur function is based on the Schur coefficients (reflection coefficients).

Since the Schur functions are closely linked to Carathéodory functions, we might suspect a certain relation between

$$|\gamma_i| < 1 \quad \text{and} \quad \delta_i > 0. \tag{27}$$

Utilizing these relations

The test whether a function is a Schur function is based on the Schur coefficients (reflection coefficients).

Since the Schur functions are closely linked to Carathéodory functions, we might suspect a certain relation between

$$|\gamma_i| < 1 \quad \text{and} \quad \delta_i > 0, \tag{27}$$

and the diagonal elements c_i of the Cholesky factor of the decomposition of the associated Toeplitz matrices,

$$T_m = C_m C_m^H. \tag{28}$$

Utilizing these relations

The test whether a function is a Schur function is based on the Schur coefficients (reflection coefficients).

Since the Schur functions are closely linked to Carathéodory functions, we might suspect a certain relation between

$$|\gamma_i| < 1 \quad \text{and} \quad \delta_i > 0, \tag{27}$$

and the diagonal elements c_i of the Cholesky factor of the decomposition of the associated Toeplitz matrices,

$$T_m = C_m C_m^H. \tag{28}$$

It turns out that the Schur algorithm (as a byproduct) cheaply computes the Cholesky decomposition.

Outline

Schur's expansion

With

$$s_i(z) = \frac{a_i(z)}{b_i(z)}, \qquad G_i(z) = \begin{pmatrix} b_i(z) & a_i(z) \end{pmatrix} \tag{29}$$

we can state a "linearized" version of Schur's expansion.

Schur's expansion

With

$$s_i(z) = \frac{a_i(z)}{b_i(z)}, \qquad G_i(z) = \begin{pmatrix} b_i(z) & a_i(z) \end{pmatrix} \tag{29}$$

we can state a "linearized" version of Schur's expansion,

$$G_{i+1}(z) = G_i(z)\ \phi_i \begin{pmatrix} 1 & -\gamma_i \\ -\overline{\gamma}_i & 1 \end{pmatrix} \begin{pmatrix} z & 0 \\ 0 & 1 \end{pmatrix}, \quad \phi_i = \text{arbitrary.} \tag{30}$$

Schur's expansion

With

$$s_i(z) = \frac{a_i(z)}{b_i(z)}, \qquad G_i(z) = \begin{pmatrix} b_i(z) & a_i(z) \end{pmatrix} \tag{29}$$

we can state a "linearized" version of Schur's expansion,

$$G_{i+1}(z) = G_i(z)\ \phi_i \begin{pmatrix} 1 & -\gamma_i \\ -\overline{\gamma}_i & 1 \end{pmatrix} \begin{pmatrix} z & 0 \\ 0 & 1 \end{pmatrix}, \quad \phi_i = \text{arbitrary}, \tag{30}$$

since

$$s_{i+1}(z) = \frac{a_{i+1}(z)}{b_{i+1}(z)} \tag{31}$$

$$\tag{32}$$

Schur's expansion

With

$$s_i(z) = \frac{a_i(z)}{b_i(z)}, \qquad G_i(z) = \begin{pmatrix} b_i(z) & a_i(z) \end{pmatrix} \tag{29}$$

we can state a "linearized" version of Schur's expansion,

$$G_{i+1}(z) = G_i(z)\ \phi_i \begin{pmatrix} 1 & -\gamma_i \\ -\overline{\gamma}_i & 1 \end{pmatrix} \begin{pmatrix} z & 0 \\ 0 & 1 \end{pmatrix}, \quad \phi_i = \text{arbitrary}, \tag{30}$$

since

$$s_{i+1}(z) = \frac{a_{i+1}(z)}{b_{i+1}(z)} = \frac{1}{z}\,\frac{s_i(z) - \gamma_i}{1 - \overline{\gamma}_i s_i(z)} \tag{31}$$

$$\tag{32}$$

Schur's expansion

With

$$s_i(z) = \frac{a_i(z)}{b_i(z)}, \qquad G_i(z) = \begin{pmatrix} b_i(z) & a_i(z) \end{pmatrix} \tag{29}$$

we can state a "linearized" version of Schur's expansion,

$$G_{i+1}(z) = G_i(z)\ \phi_i \begin{pmatrix} 1 & -\gamma_i \\ -\overline{\gamma}_i & 1 \end{pmatrix} \begin{pmatrix} z & 0 \\ 0 & 1 \end{pmatrix}, \quad \phi_i = \text{arbitrary}, \tag{30}$$

since

$$s_{i+1}(z) = \frac{a_{i+1}(z)}{b_{i+1}(z)} = \frac{1}{z}\,\frac{s_i(z) - \gamma_i}{1 - \overline{\gamma}_i s_i(z)} \cdot \frac{b_i(z)}{b_i(z)} \tag{31}$$

$$\tag{32}$$

Schur's expansion

With

$$s_i(z) = \frac{a_i(z)}{b_i(z)}, \qquad G_i(z) = \begin{pmatrix} b_i(z) & a_i(z) \end{pmatrix} \tag{29}$$

we can state a "linearized" version of Schur's expansion,

$$G_{i+1}(z) = G_i(z)\ \phi_i \begin{pmatrix} 1 & -\gamma_i \\ -\overline{\gamma}_i & 1 \end{pmatrix} \begin{pmatrix} z & 0 \\ 0 & 1 \end{pmatrix}, \quad \phi_i = \text{arbitrary}, \tag{30}$$

since

$$s_{i+1}(z) = \frac{a_{i+1}(z)}{b_{i+1}(z)} = \frac{1}{z}\frac{s_i(z) - \gamma_i}{1 - \overline{\gamma}_i s_i(z)} \cdot \frac{b_i(z)}{b_i(z)} \tag{31}$$

$$= \frac{1}{z}\frac{a_i(z) - \gamma_i b_i(z)}{b_i(z) - \overline{\gamma}_i a_i(z)} \tag{32}$$

Schur's expansion

With

$$s_i(z) = \frac{a_i(z)}{b_i(z)}, \qquad G_i(z) = \begin{pmatrix} b_i(z) & a_i(z) \end{pmatrix} \tag{29}$$

we can state a "linearized" version of Schur's expansion,

$$G_{i+1}(z) = G_i(z)\ \phi_i \begin{pmatrix} 1 & -\gamma_i \\ -\overline{\gamma}_i & 1 \end{pmatrix} \begin{pmatrix} z & 0 \\ 0 & 1 \end{pmatrix}, \quad \phi_i = \text{arbitrary}, \tag{30}$$

since

$$s_{i+1}(z) = \frac{a_{i+1}(z)}{b_{i+1}(z)} = \frac{1}{z}\frac{s_i(z) - \gamma_i}{1 - \overline{\gamma}_i s_i(z)} \cdot \frac{b_i(z)}{b_i(z)} \tag{31}$$

$$= \frac{1}{z}\frac{a_i(z) - \gamma_i b_i(z)}{b_i(z) - \overline{\gamma}_i a_i(z)} = \frac{G_i(z)\begin{pmatrix} -\gamma_i \\ 1 \end{pmatrix} \cdot 1}{G_i(z)\begin{pmatrix} 1 \\ -\overline{\gamma}_i \end{pmatrix} \cdot z}. \tag{32}$$

Schur's expansion

With

$$s_i(z) = \frac{a_i(z)}{b_i(z)}, \qquad G_i(z) = \begin{pmatrix} b_i(z) & a_i(z) \end{pmatrix} \tag{29}$$

we can state a "linearized" version of Schur's expansion,

$$G_{i+1}(z) = G_i(z)\ \phi_i \begin{pmatrix} 1 & -\gamma_i \\ -\overline{\gamma}_i & 1 \end{pmatrix} \begin{pmatrix} z & 0 \\ 0 & 1 \end{pmatrix}, \qquad \color{red}{\phi_i = \text{arbitrary?}} \tag{30}$$

since

$$s_{i+1}(z) = \frac{a_{i+1}(z)}{b_{i+1}(z)} = \frac{1}{z}\frac{s_i(z) - \gamma_i}{1 - \overline{\gamma}_i s_i(z)} \cdot \frac{b_i(z)}{b_i(z)} \tag{31}$$

$$= \frac{1}{z}\frac{a_i(z) - \gamma_i b_i(z)}{b_i(z) - \overline{\gamma}_i a_i(z)} = \frac{G_i(z)\begin{pmatrix} -\gamma_i \\ 1 \end{pmatrix} \cdot 1}{G_i(z)\begin{pmatrix} 1 \\ -\overline{\gamma}_i \end{pmatrix} \cdot z}. \tag{32}$$

Schur's expansion

With

$$s_i(z) = \frac{a_i(z)}{b_i(z)}, \qquad G_i(z) = \begin{pmatrix} b_i(z) & a_i(z) \end{pmatrix} \tag{29}$$

we can state a "linearized" version of Schur's expansion,

$$G_{i+1}(z) = G_i(z)\ \phi_i \begin{pmatrix} 1 & -\gamma_i \\ -\overline{\gamma}_i & 1 \end{pmatrix} \begin{pmatrix} z & 0 \\ 0 & 1 \end{pmatrix}, \qquad \phi_i = \frac{1}{\sqrt{1-|\gamma_i|^2}}, \tag{30}$$

since

$$s_{i+1}(z) = \frac{a_{i+1}(z)}{b_{i+1}(z)} = \frac{1}{z}\,\frac{s_i(z)-\gamma_i}{1-\overline{\gamma}_i s_i(z)} \cdot \frac{b_i(z)}{b_i(z)} \tag{31}$$

$$= \frac{1}{z}\,\frac{a_i(z)-\gamma_i b_i(z)}{b_i(z)-\overline{\gamma}_i a_i(z)} = \frac{G_i(z)\begin{pmatrix} -\gamma_i \\ 1 \end{pmatrix} \cdot 1}{G_i(z)\begin{pmatrix} 1 \\ -\overline{\gamma}_i \end{pmatrix} \cdot z}. \tag{32}$$

Reversing the approach

Suppose we are given a positive definite finite section H of a symmetric infinite matrix

$$\mathfrak{H} = \mathfrak{B} - \mathfrak{A} \tag{33}$$

and want to compute the Cholesky decomposition.

Reversing the approach

Suppose we are given a positive definite finite section H of a symmetric infinite matrix

$$\mathfrak{H} = \mathfrak{B} - \mathfrak{A} \tag{33}$$

and want to compute the Cholesky decomposition.

H has a special property, namely that a displacement of the entries along the diagonal preserves major part of the structure.

Reversing the approach

Suppose we are given a positive definite finite section H of a symmetric infinite matrix

$$\mathfrak{H} = \mathfrak{B} - \mathfrak{A} \tag{33}$$

and want to compute the Cholesky decomposition.

H has a special property, namely that a displacement of the entries along the diagonal preserves major part of the structure.

Notation is changed slightly: Let $L(a)$ denote a lower triangular Toeplitz matrix with entries $a \in \mathbb{C}^n$ in the first column. Then

$$H = L(b)L(b)^H - L(a)L(a)^H \tag{34}$$

is Toeplitz. This change corresponds to complex conjugation.

Reversing the approach

Suppose we are given a positive definite finite section H of a symmetric infinite matrix

$$\mathfrak{H} = \mathfrak{B} - \mathfrak{A} \tag{33}$$

and want to compute the Cholesky decomposition.

H has a special property, namely that a displacement of the entries along the diagonal preserves major part of the structure.

Notation is changed slightly: Let $L(a)$ denote a lower triangular Toeplitz matrix with entries $a \in \mathbb{C}^n$ in the first column. Then

$$H = L(b)L(b)^H - L(a)L(a)^H \tag{34}$$

is Toeplitz. This change corresponds to complex conjugation.

The displacement can be described by shifting the entries of a and b, i.e., the elements of the power series a and b.

Outline

Displacement structure

Let $Z = \mathrm{diag}(\text{ones}(n-1,1),-1)$ denote the displacement by one position. Then

$$H - ZHZ^H \tag{35}$$

Displacement structure

Let $Z = \mathrm{diag}(\text{ones}(n-1, 1), -1)$ denote the displacement by one position. Then

$$\nabla H \equiv H - ZHZ^H \tag{35}$$

Displacement structure

Let $Z = \mathrm{diag}(\text{ones}(n-1,1),-1)$ denote the displacement by one position. Then

$$\nabla H \equiv H - ZHZ^H = bb^H - aa^H \tag{35}$$

Displacement structure

Let $Z = \mathrm{diag}(\text{ones}(n-1,1),-1)$ denote the displacement by one position. Then

$$\nabla H \equiv H - ZHZ^H = bb^H - aa^H = GJG^H \tag{35}$$

with

$$G = \begin{pmatrix} b & a \end{pmatrix}, \qquad J = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \tag{36}$$

Displacement structure

Let $Z = \mathrm{diag}(\text{ones}(n-1,1),-1)$ denote the displacement by one position. Then

$$\nabla H \equiv H - ZHZ^H = bb^H - aa^H = GJG^H \tag{35}$$

with

$$G = \begin{pmatrix} b & a \end{pmatrix}, \qquad J = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \tag{36}$$

Using the fact that Z is nilpotent and utilizing a Neumann series,

$$\mathrm{vec}(H) = (I - Z \otimes Z)^{-1}\mathrm{vec}(GJG) = \sum_{j=0}^{n-1} (Z \otimes Z)^j \mathrm{vec}(GJG^H) \tag{37}$$

Displacement structure

Let $Z = \mathrm{diag}(\text{ones}(n-1,1),-1)$ denote the displacement by one position. Then

$$\nabla H \equiv H - ZHZ^H = bb^H - aa^H = GJG^H \tag{35}$$

with

$$G = \begin{pmatrix} b & a \end{pmatrix}, \qquad J = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \tag{36}$$

Using the fact that Z is nilpotent and utilizing a Neumann series,

$$\mathrm{vec}(H) = (I - Z \otimes Z)^{-1}\mathrm{vec}(GJG) = \sum_{j=0}^{n-1} (Z \otimes Z)^j \mathrm{vec}(GJG^H) \tag{37}$$

and therefore we can recover H using the generator G,

$$H = \sum_{j=0}^{n-1} Z^j GJG^H (Z^T)^j. \tag{38}$$

Displacement structure

Generators are unique up to post-multiplication by J-unitary matrices $\Theta \in \mathbb{C}$, defined by $\Theta J \Theta^H = J$, including hyperbolic rotations $\Theta(\gamma)$,

$$\Theta(\gamma) = \frac{1}{\sqrt{1 - |\gamma|^2}} \begin{pmatrix} 1 & -\gamma \\ -\overline{\gamma} & 1 \end{pmatrix}. \tag{39}$$

Displacement structure

Generators are unique up to post-multiplication by J-unitary matrices $\Theta \in \mathbb{C}$, defined by $\Theta J \Theta^H = J$, including hyperbolic rotations $\Theta(\gamma)$,

$$\Theta(\gamma) = \frac{1}{\sqrt{1-|\gamma|^2}} \begin{pmatrix} 1 & -\gamma \\ -\overline{\gamma} & 1 \end{pmatrix}. \tag{39}$$

This is obvious, since with $\tilde{G} = G\Theta$ for a J-unitary matrix Θ

$$\tilde{G} J \tilde{G}^H = G \Theta J \Theta^H G^H = G J G^H. \tag{40}$$

Displacement structure

Generators are unique up to post-multiplication by J-unitary matrices $\Theta \in \mathbb{C}$, defined by $\Theta J \Theta^H = J$, including hyperbolic rotations $\Theta(\gamma)$,

$$\Theta(\gamma) = \frac{1}{\sqrt{1 - |\gamma|^2}} \begin{pmatrix} 1 & -\gamma \\ -\overline{\gamma} & 1 \end{pmatrix}. \tag{39}$$

This is obvious, since with $\tilde{G} = G\Theta$ for a J-unitary matrix Θ

$$\tilde{G} J \tilde{G}^H = G \Theta J \Theta^H G^H = G J G^H. \tag{40}$$

Since H is assumed positive definite ($0 < h_{11} = |b^T e_1|^2 - |a^T e_1|^2$), we can always chose a generator G in “proper form”

$$G = \begin{pmatrix} b & a \end{pmatrix} = \begin{pmatrix} \star & {\color{red}0} \\ \star & \star \\ \vdots & \vdots \\ \star & \star \end{pmatrix}. \tag{41}$$

Outline

Cholesky decomposition

The generator can be used to compactly ($2n$ entries) represent a HPD Toeplitz matrix (n^2 entries). It is slightly more expensive than parameterization by first row or column (n entries) but enables a cheap triangular decomposition.

Cholesky decomposition

The generator can be used to compactly ($2n$ entries) represent a HPD Toeplitz matrix (n^2 entries). It is slightly more expensive than parameterization by first row or column (n entries) but enables a cheap triangular decomposition.

The Cholesky decomposition of HPD H can be computed as follows. Set $H_1 = H$ and repeat for $i = 1, \dots, n$

$$H_{i+1} = H_i - c_i c_i^H, \tag{42}$$

where $c_{ii} = \sqrt{h_{ii}^{(i)}}$ and $c_i = H_i e_i / c_{ii}$.

Cholesky decomposition

The generator can be used to compactly ($2n$ entries) represent a HPD Toeplitz matrix (n^2 entries). It is slightly more expensive than parameterization by first row or column (n entries) but enables a cheap triangular decomposition.

The Cholesky decomposition of HPD H can be computed as follows. Set $H_1 = H$ and repeat for $i = 1, \ldots, n$

$$H_{i+1} = H_i - c_i c_i^H, \tag{42}$$

where $c_{ii} = \sqrt{h_{ii}^{(i)}}$ and $c_i = H_i e_i / c_{ii}$.

This is the Jacobi-style Cholesky decomposition working with rank-one updates that introduce a new zero row and vector in H_{i+1} compared to H_i.

Cholesky decomposition

The generator can be used to compactly ($2n$ entries) represent a HPD Toeplitz matrix (n^2 entries). It is slightly more expensive than parameterization by first row or column (n entries) but enables a cheap triangular decomposition.

The Cholesky decomposition of HPD H can be computed as follows. Set $H_1 = H$ and repeat for $i = 1, \ldots, n$

$$H_{i+1} = H_i - c_i c_i^H, \tag{42}$$

where $c_{ii} = \sqrt{h_{ii}^{(i)}}$ and $c_i = H_i e_i / c_{ii}$.

This is the Jacobi-style Cholesky decomposition working with rank-one updates that introduce a new zero row and vector in H_{i+1} compared to H_i.

This style is not very economic and used for demonstration only.

Cholesky decomposition

We have seen that a HPD Toeplitz matrix (scaled; now denoted by T),

$$T = \begin{pmatrix} 1 & \bar{t}_1 & \cdots & \bar{t}_n \\ t_1 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \bar{t}_1 \\ t_n & \cdots & t_1 & 1 \end{pmatrix} \tag{43}$$

may be represented by its generator G.

Cholesky decomposition

We have seen that a HPD Toeplitz matrix (scaled; now denoted by T),

$$T = \begin{pmatrix} 1 & \bar{t}_1 & \cdots & \bar{t}_n \\ t_1 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \bar{t}_1 \\ t_n & \cdots & t_1 & 1 \end{pmatrix} \tag{43}$$

may be represented by its generator G using the identity

$$T = \sum_{j=0}^{n} Z^j G J G^H (Z^T)^j = GJG + ZGJ(ZG)^H + \cdots + Z^n GJ(Z^n G)^H. \tag{44}$$

Cholesky decomposition

We have seen that a HPD Toeplitz matrix (scaled; now denoted by T),

$$T = \begin{pmatrix} 1 & \bar{t}_1 & \cdots & \bar{t}_n \\ t_1 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \bar{t}_1 \\ t_n & \cdots & t_1 & 1 \end{pmatrix} \tag{43}$$

may be represented by its generator G using the identity

$$T = \sum_{j=0}^{n} Z^j G J G^H (Z^T)^j = GJG + ZGJ(ZG)^H + \cdots + Z^n GJ(Z^n G)^H. \tag{44}$$

It is not hard to see that $G = \begin{pmatrix} 1 & 0 \\ t_1 & t_1 \\ \vdots & \vdots \\ t_n & t_n \end{pmatrix}$ is a generator in proper form.

Cholesky decomposition

The only portion of

$$T_1 = T = \sum_{j=0}^{n} Z^j GJG^H (Z^T)^j = GJG + ZGJ(ZG)^H + \cdots + Z^n GJ(Z^n G)^H \qquad (45)$$

contributing to the first row and column is given by

$$c_1 c_1^H = bb^H. \qquad (46)$$

Cholesky decomposition

The only portion of

$$T_1 = T = \sum_{j=0}^{n} Z^j GJG^H (Z^T)^j = GJG + ZGJ(ZG)^H + \cdots + Z^n GJ(Z^n G)^H \qquad (45)$$

contributing to the first row and column is given by

$$c_1 c_1^H = bb^H. \qquad (46)$$

We inductively use this property.

Cholesky decomposition

The only portion of

$$T_1 = T = \sum_{j=0}^{n} Z^j GJG^H (Z^T)^j = GJG + ZGJ(ZG)^H + \cdots + Z^n GJ(Z^n G)^H \qquad (45)$$

contributing to the first row and column is given by

$$c_1 c_1^H = bb^H. \qquad (46)$$

We inductively use this property. It is easy to rewrite the representation by introducing an intermediate generator $\tilde{G}$ of T_2,

$$T_2 = T_1 - c_1 c_1^H \qquad (47)$$

Cholesky decomposition

The only portion of

$$T_1 = T = \sum_{j=0}^{n} Z^j GJG^H (Z^T)^j = GJG + ZGJ(ZG)^H + \cdots + Z^n GJ(Z^n G)^H \tag{45}$$

contributing to the first row and column is given by

$$c_1 c_1^H = bb^H. \tag{46}$$

We inductively use this property. It is easy to rewrite the representation by introducing an intermediate generator $\tilde{G}$ of T_2,

$$T_2 = T_1 - c_1 c_1^H = \sum_{j=0}^{n} Z^j GJG^H (Z^T)^j - bb^H \tag{47}$$

Cholesky decomposition

The only portion of

$$T_1 = T = \sum_{j=0}^{n} Z^j GJG^H (Z^T)^j = GJG + ZGJ(ZG)^H + \cdots + Z^n GJ(Z^n G)^H \qquad (45)$$

contributing to the first row and column is given by

$$c_1 c_1^H = bb^H. \qquad (46)$$

We inductively use this property. It is easy to rewrite the representation by introducing an intermediate generator $\tilde{G}$ of T_2,

$$T_2 = T_1 - c_1 c_1^H = \sum_{j=0}^{n} Z^j GJG^H (Z^T)^j - bb^H = \sum_{j=0}^{n} Z^j \tilde{G} J \tilde{G}^H (Z^T)^j, \qquad (47)$$

Cholesky decomposition

The only portion of

$$T_1 = T = \sum_{j=0}^{n} Z^j GJG^H (Z^T)^j = GJG + ZGJ(ZG)^H + \cdots + Z^n GJ(Z^n G)^H \qquad (45)$$

contributing to the first row and column is given by

$$c_1 c_1^H = bb^H. \qquad (46)$$

We inductively use this property. It is easy to rewrite the representation by introducing an intermediate generator $\tilde{G}$ of T_2,

$$T_2 = T_1 - c_1 c_1^H = \sum_{j=0}^{n} Z^j GJG^H (Z^T)^j - bb^H = \sum_{j=0}^{n} Z^j \tilde{G} J \tilde{G}^H (Z^T)^j, \qquad (47)$$

where $\tilde{G}$ is defined by

$$\tilde{G} = \begin{pmatrix} Zb & a \end{pmatrix}. \qquad (48)$$

Cholesky decomposition

This generator is usually not in “proper” form, so we compute another generator G_1 by applying a hyperbolic rotation to $\tilde{G}$,

$$G_1 = \tilde{G}\Theta(\gamma). \tag{49}$$

Cholesky decomposition

This generator is usually not in “proper” form, so we compute another generator G_1 by applying a hyperbolic rotation to $\tilde{G}$,

$$G_1 = \tilde{G}\Theta(\gamma), \tag{49}$$

where $\gamma = a(2)/b(1)$ is chosen to annihilate the element in position 2 in a.

Cholesky decomposition

This generator is usually not in “proper” form, so we compute another generator G_1 by applying a hyperbolic rotation to $\tilde{G}$,

$$G_1 = \tilde{G}\Theta(\gamma), \tag{49}$$

where $\gamma = a(2)/b(1)$ is chosen to annihilate the element in position 2 in a.

Thus far, we have succesfully removed the first row and column and computed the first column of the Cholesky factor. The new generator has already “proper” form.

Cholesky decomposition

This generator is usually not in “proper” form, so we compute another generator G_1 by applying a hyperbolic rotation to $\tilde{G}$,

$$G_1 = \tilde{G}\Theta(\gamma), \tag{49}$$

where $\gamma = a(2)/b(1)$ is chosen to annihilate the element in position 2 in a.

Thus far, we have succesfully removed the first row and column and computed the first column of the Cholesky factor. The new generator has already “proper” form.

This process can be repeated until all rows and columns have been deflated and all columns of the Cholesky factor have been computed.

Cholesky decomposition

This generator is usually not in “proper” form, so we compute another generator G_1 by applying a hyperbolic rotation to $\tilde{G}$,

$$G_1 = \tilde{G}\Theta(\gamma), \tag{49}$$

where $\gamma = a(2)/b(1)$ is chosen to annihilate the element in position 2 in a.

Thus far, we have succesfully removed the first row and column and computed the first column of the Cholesky factor. The new generator has already “proper” form.

This process can be repeated until all rows and columns have been deflated and all columns of the Cholesky factor have been computed.

Stripping of leading zero blocks in the matrix and the generator we can go through the steps and compute only the nonzero elements of the columns of the Cholesky factor.

Cholesky decomposition

It turns out that we have recovered the Schur algorithm, since this is just another way of describing Schur's classical algorithm

$$s_{i+1}(z) = \frac{1}{z}\frac{s_i(z)-\gamma_i}{1-\overline{\gamma}_i s_i(z)}, \qquad \gamma_i = s_i(0). \tag{50}$$

Cholesky decomposition

It turns out that we have recovered the Schur algorithm, since this is just another way of describing Schur's classical algorithm

$$s_{i+1}(z) = \frac{1}{z} \frac{s_i(z) - \gamma_i}{1 - \overline{\gamma}_i s_i(z)}, \quad \gamma_i = s_i(0), \tag{50}$$

for the case of rational functions

$$s_i(z) = \frac{a_i(z)}{b_i(z)}. \tag{51}$$

Cholesky decomposition

It turns out that we have recovered the Schur algorithm, since this is just another way of describing Schur's classical algorithm

$$s_{i+1}(z) = \frac{1}{z}\frac{s_i(z)-\gamma_i}{1-\overline{\gamma}_i s_i(z)}, \qquad \gamma_i = s_i(0), \tag{50}$$

for the case of rational functions

$$s_i(z) = \frac{a_i(z)}{b_i(z)}, \tag{51}$$

linearized in form of a coupled iteration

$$G_{i+1}(z) = G_i(z)\ \phi_i \begin{pmatrix} 1 & -\gamma_i \\ -\overline{\gamma}_i & 1 \end{pmatrix} \begin{pmatrix} z & 0 \\ 0 & 1 \end{pmatrix}, \qquad \phi_i = \frac{1}{\sqrt{1-|\gamma_i|^2}}, \tag{52}$$

with generators $G_i(z) = \begin{pmatrix} b_i(z) & a_i(z) \end{pmatrix}$.

Outline

The Schur algorithm

Stated in terms of vectors $a_i, b_i \in \mathbb{C}^n$ representing the coefficients of the respective power series and using the shift operator Z in place of multiplication by z we have shown that the Schur algorithm takes the following form:

The Schur algorithm

Stated in terms of vectors $a_i, b_i \in \mathbb{C}^n$ representing the coefficients of the respective power series and using the shift operator Z in place of multiplication by z we have shown that the Schur algorithm takes the following form:

Start with a generator $G = \begin{pmatrix} b_1 & a_1 \end{pmatrix}$ of HPD Toeplitz T in proper form.
Iterate: for $i = 1, \ldots, n-1$:

$$c_i \leftarrow b_i$$

$$\gamma_i \leftarrow \frac{e_{i+1}^T a_i}{e_i^T b_i}$$

$$\begin{pmatrix} b_{i+1} & a_{i+1} \end{pmatrix} \leftarrow \begin{pmatrix} Zb_i & a_i \end{pmatrix} \frac{1}{\sqrt{1-|\gamma_i|^2}} \begin{pmatrix} 1 & -\gamma_i \\ -\overline{\gamma}_i & 1 \end{pmatrix}$$

endfor

Set $c_n \leftarrow b_n$

The Schur algorithm

Stated in terms of vectors $a_i, b_i \in \mathbb{C}^n$ representing the coefficients of the respective power series and using the shift operator Z in place of multiplication by z we have shown that the Schur algorithm takes the following form:

Start with a generator $G = \begin{pmatrix} b_1 & a_1 \end{pmatrix}$ of HPD Toeplitz T in proper form.

Iterate: for $i = 1, \ldots, n-1$:

$$c_i \leftarrow b_i$$

$$\gamma_i \leftarrow \frac{e_{i+1}^T a_i}{e_i^T b_i}$$

$$\begin{pmatrix} b_{i+1} & a_{i+1} \end{pmatrix} \leftarrow \begin{pmatrix} Zb_i & a_i \end{pmatrix} \frac{1}{\sqrt{1-|\gamma_i|^2}} \begin{pmatrix} 1 & -\gamma_i \\ -\overline{\gamma}_i & 1 \end{pmatrix}$$

endfor

Set $c_n \leftarrow b_n$

By previous considerations C iteratively defined by $Ce_i = c_i$ is the Cholesky factor of T.

The Schur algorithm; variants

One can easily use other triangular decompositions:

The Schur algorithm; variants

One can easily use other triangular decompositions:

Cholesky decomposition: this is the “standard” Schur algorithm

The Schur algorithm; variants

One can easily use other triangular decompositions:

Cholesky decomposition: this is the “standard” Schur algorithm

LDLH decomposition: a natural extension of the Schur algorithm to treat indefinite Toeplitz matrices which admit such a decomposition

The Schur algorithm; variants

One can easily use other triangular decompositions:

Cholesky decomposition: this is the “standard” Schur algorithm

LDLH decomposition: a natural extension of the Schur algorithm to treat indefinite Toeplitz matrices which admit such a decomposition

LU decomposition: another possible variant

The Schur algorithm; variants

One can easily use other triangular decompositions:

Cholesky decomposition: this is the “standard” Schur algorithm

LDLH decomposition: a natural extension of the Schur algorithm to treat indefinite Toeplitz matrices which admit such a decomposition

LU decomposition: another possible variant

The use of augmented & composed systems is used to develop variants to compute:

The Schur algorithm; variants

One can easily use other triangular decompositions:

Cholesky decomposition: this is the “standard” Schur algorithm

LDLH decomposition: a natural extension of the Schur algorithm to treat indefinite Toeplitz matrices which admit such a decomposition

LU decomposition: another possible variant

The use of augmented & composed systems is used to develop variants to compute:

- the R factor of the QR decomposition

The Schur algorithm; variants

One can easily use other triangular decompositions:

Cholesky decomposition: this is the “standard” Schur algorithm

LDLH decomposition: a natural extension of the Schur algorithm to treat indefinite Toeplitz matrices which admit such a decomposition

LU decomposition: another possible variant

The use of augmented & composed systems is used to develop variants to compute:

- the R factor of the QR decomposition
- the inverse of T

The Schur algorithm; variants

One can easily use other triangular decompositions:

Cholesky decomposition: this is the “standard” Schur algorithm

LDLH decomposition: a natural extension of the Schur algorithm to treat indefinite Toeplitz matrices which admit such a decomposition

LU decomposition: another possible variant

The use of augmented & composed systems is used to develop variants to compute:

- the R factor of the QR decomposition
- the inverse of T
- …

Outline

Displacement structure

Let $R \in \mathbb{C}^{n\times n}$ be Hermitean, $R = R^H$. The displacement of R with respect to $F \in \mathbb{C}$ is defined by

$$\nabla_F R = R - FRF^H. \tag{53}$$

Displacement structure

Let $R \in \mathbb{C}^{n\times n}$ be Hermitean, $R = R^H$. The displacement of R with respect to $F \in \mathbb{C}$ is defined by

$$\nabla_F R = R - FRF^H. \tag{53}$$

The matrix R is said to have low displacement rank if

$$\operatorname{rank}(\nabla_F) \ll n. \tag{54}$$

Displacement structure

Let $R \in \mathbb{C}^{n \times n}$ be Hermitean, $R = R^H$. The displacement of R with respect to $F \in \mathbb{C}$ is defined by

$$\nabla_F R = R - FRF^H. \tag{53}$$

The matrix R is said to have low displacement rank if

$$\operatorname{rank}(\nabla_F) \ll n. \tag{54}$$

In this case R is said to have displacement structure with respect to F.

Displacement structure

Let $R \in \mathbb{C}^{n\times n}$ be Hermitean, $R = R^H$. The displacement of R with respect to $F \in \mathbb{C}$ is defined by

$$\nabla_F R = R - FRF^H. \tag{53}$$

The matrix R is said to have low displacement rank if

$$\operatorname{rank}(\nabla_F) \ll n. \tag{54}$$

In this case R is said to have displacement structure with respect to F.

An example are symmetric real Toeplitz matrices T having displacement structure with respect to the shift matrix

$$F = Z = \begin{pmatrix} 0 & & & \\ 1 & \ddots & & \\ & \ddots & 0 & \\ & & 1 & 0 \end{pmatrix}. \tag{55}$$

Displacement structure

Other types of displacement include the Stein type displacement

$$\nabla_{\{F,A\}} R = R - FRA^H. \tag{56}$$

Displacement structure

Other types of displacement include the Stein type displacement

$$\nabla_{\{F,A\}} R = R - FRA^H. \tag{56}$$

Stein's theorem: X is convergent, if and only if there exists a positive definite A such that $A - X^H AX$ is positive definite.

Displacement structure

Other types of displacement include the Stein type displacement

$$\nabla_{\{F,A\}} R = R - FRA^H, \tag{56}$$

and the Sylvester type displacement

$$\nabla_{\{F,A\}} R = FR - RA^H. \tag{57}$$

Stein's theorem: X is convergent, if and only if there exists a positive definite A such that $A - X^H AX$ is positive definite.

Displacement structure

Other types of displacement include the Stein type displacement

$$\nabla_{\{F,A\}}R = R - FRA^H, \quad (56)$$

and the Sylvester type displacement

$$\nabla_{\{F,A\}}R = FR - RA^H. \quad (57)$$

Both are covered by the general displacement

$$\nabla_{\{\Omega,\Delta,F,A\}}R = \Omega R \Delta^H - FRA^H. \quad (58)$$

Stein's theorem: X is convergent, if and only if there exists a positive definite A such that $A - X^H AX$ is positive definite.

Displacement structure

Other types of displacement include the Stein type displacement

$$\nabla_{\{F,A\}} R = R - FRA^H, \tag{56}$$

and the Sylvester type displacement

$$\nabla_{\{F,A\}} R = FR - RA^H. \tag{57}$$

Both are covered by the general displacement

$$\nabla_{\{\Omega,\Delta,F,A\}} R = \Omega R \Delta^H - FRA^H. \tag{58}$$

Matrices with Stein type low displacement rank are termed Toeplitz-like, those with Sylvester type low displacement rank Hankel-like.

Stein's theorem: X is convergent, if and only if there exists a positive definite A such that $A - X^H AX$ is positive definite.

Displacement structure

The so-called Pick matrix A is defined by

$$A = \left(\frac{x_i x_j^H - y_i y_j^H}{1 - f_i f_j^H} \right)_{i,j=1}^{n}, \tag{59}$$

where $x_i \in \mathbb{C}^{1\times p}$ and $y_i \in \mathbb{C}^{1\times q}$ are complex row vectors and f_i are complex points inside the open unit disc.

Displacement structure

The so-called Pick matrix A is defined by

$$A = \left(\frac{x_i x_j^H - y_i y_j^H}{1 - f_i f_j^H} \right)_{i,j=1}^{n}, \tag{59}$$

where $x_i \in \mathbb{C}^{1\times p}$ and $y_i \in \mathbb{C}^{1\times q}$ are complex row vectors and f_i are complex points inside the open unit disc.

Such A has displacement rank $r = p + q$ with respect to

$$F = \operatorname{diag}(f_1, \ldots, f_n). \tag{60}$$

Displacement structure

The so-called Pick matrix A is defined by

$$A = \left(\frac{x_i x_j^H - y_i y_j^H}{1 - f_i f_j^H} \right)_{i,j=1}^{n}, \tag{59}$$

where $x_i \in \mathbb{C}^{1\times p}$ and $y_i \in \mathbb{C}^{1\times q}$ are complex row vectors and f_i are complex points inside the open unit disc.

Such A has displacement rank $r = p + q$ with respect to

$$F = \operatorname{diag}(f_1, \ldots, f_n), \tag{60}$$

since

$$A - FAF^H = \begin{pmatrix} x_1 & y_1 \\ \vdots & \vdots \\ x_n & y_n \end{pmatrix} \begin{pmatrix} I_p & O_{p,q} \\ O_{q,p} & -I_q \end{pmatrix} \begin{pmatrix} x_1 & y_1 \\ \vdots & \vdots \\ x_n & y_n \end{pmatrix}^H. \tag{61}$$

Displacement structure

A nonsymmetric example is given by a Vandermonde matrix

$$V = \begin{pmatrix} 1 & \alpha_1 & \alpha_1^2 & \cdots & \alpha_1^n \\ 1 & \alpha_2 & \alpha_2^2 & \cdots & \alpha_2^n \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \alpha_n & \alpha_n^2 & \cdots & \alpha_n^n \end{pmatrix}. \tag{62}$$

Displacement structure

A nonsymmetric example is given by a Vandermonde matrix

$$V = \begin{pmatrix} 1 & \alpha_1 & \alpha_1^2 & \cdots & \alpha_1^n \\ 1 & \alpha_2 & \alpha_2^2 & \cdots & \alpha_2^n \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \alpha_n & \alpha_n^2 & \cdots & \alpha_n^n \end{pmatrix}. \qquad (62)$$

Such matrix V has displacement rank one with respect to

$$F = \operatorname{diag}(\alpha_1, \dots, \alpha_n) \qquad (63)$$

and the shift matrix Z introduced before.

Displacement structure

A nonsymmetric example is given by a Vandermonde matrix

$$V = \begin{pmatrix} 1 & \alpha_1 & \alpha_1^2 & \cdots & \alpha_1^n \\ 1 & \alpha_2 & \alpha_2^2 & \cdots & \alpha_2^n \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \alpha_n & \alpha_n^2 & \cdots & \alpha_n^n \end{pmatrix}. \tag{62}$$

Such matrix V has displacement rank one with respect to

$$F = \operatorname{diag}(\alpha_1, \ldots, \alpha_n) \tag{63}$$

and the shift matrix Z introduced before, since

$$V - FVZ^T = ee_1^T, \tag{64}$$

where e denotes the vector of all ones and e_1 is the first standard unit vector.

Displacement structure

Another example, this time with respect to a Sylvester type displacement is a Cauchy matrix C defined by

$$C = \begin{pmatrix} \frac{1}{x_1 - y_1} & \cdots & \frac{1}{x_1 - y_n} \\ \vdots & \ddots & \vdots \\ \frac{1}{x_n - y_1} & \cdots & \frac{1}{x_n - y_n} \end{pmatrix}. \tag{65}$$

Displacement structure

Another example, this time with respect to a Sylvester type displacement is a Cauchy matrix C defined by

$$C = \begin{pmatrix} \frac{1}{x_1 - y_1} & \cdots & \frac{1}{x_1 - y_n} \\ \vdots & \ddots & \vdots \\ \frac{1}{x_n - y_1} & \cdots & \frac{1}{x_n - y_n} \end{pmatrix}. \tag{65}$$

Cauchy matrices have Sylvester type displacement rank one,

$$\nabla_{\{\mathrm{diag}(x), \mathrm{diag}(y)\}} C = \mathrm{diag}(x) \cdot C - C \cdot \mathrm{diag}(y) = ee^T. \tag{66}$$

Displacement structure

Another example, this time with respect to a Sylvester type displacement is a Cauchy matrix C defined by

$$C = \begin{pmatrix} \frac{1}{x_1 - y_1} & \cdots & \frac{1}{x_1 - y_n} \\ \vdots & \ddots & \vdots \\ \frac{1}{x_n - y_1} & \cdots & \frac{1}{x_n - y_n} \end{pmatrix}. \tag{65}$$

Cauchy matrices have Sylvester type displacement rank one,

$$\nabla_{\{\mathrm{diag}(x),\mathrm{diag}(y)\}} C = \mathrm{diag}(x) \cdot C - C \cdot \mathrm{diag}(y) = ee^T. \tag{66}$$

A well-known example of a Cauchy matrix is the famous Hilbert matrix H with entries

$$h_{ij} = \frac{1}{i + j - 1}. \tag{67}$$

Outline

Recovering R

For simplicity we assume the Hermitean Stein displacement case

$$\nabla R = R - FRF^H = GJG^H, \qquad J = J^H, \quad J^2 = I, \tag{68}$$

where $G \in \mathbb{C}^{n\times r}$ has full rank.

Recovering R

For simplicity we assume the Hermitean Stein displacement case

$$\nabla R = R - FRF^H = GJG^H, \qquad J = J^H, \quad J^2 = I, \tag{68}$$

where $G \in \mathbb{C}^{n\times r}$ has full rank.

We assume further that F is lower triangular and that $f = \mathrm{diag}(F)$ satisfies

$$\bar{f}_i f_j \neq 1 \quad \text{for all} \quad i, j. \tag{69}$$

Recovering R

For simplicity we assume the Hermitean Stein displacement case

$$\nabla R = R - FRF^H = GJG^H, \qquad J = J^H, \quad J^2 = I, \tag{68}$$

where $G \in \mathbb{C}^{n\times r}$ has full rank.

We assume further that F is lower triangular and that $f = \mathrm{diag}(F)$ satisfies

$$\bar{f}_i f_j \neq 1 \quad \text{for all} \quad i,j. \tag{69}$$

Then we can theoretically recover R from its generator pair (G,J) as follows:

$$\mathrm{vec}(R) = \left(I - \overline{F} \otimes F\right)^{-1} \mathrm{vec}(GJG^H). \tag{70}$$

Recovering R

For simplicity we assume the Hermitean Stein displacement case

$$\nabla R = R - FRF^H = GJG^H, \qquad J = J^H, \quad J^2 = I, \tag{68}$$

where $G \in \mathbb{C}^{n\times r}$ has full rank.

We assume further that F is lower triangular and that $f = \operatorname{diag}(F)$ satisfies

$$\bar{f}_i f_j \neq 1 \quad \text{for all} \quad i,j. \tag{69}$$

Then we can theoretically recover R from its generator pair (G,J) as follows:

$$\operatorname{vec}(R) = \left(I - \overline{F} \otimes F\right)^{-1} \operatorname{vec}(GJG^H). \tag{70}$$

The assumption that F is lower triangular leads to a lower triangular

$$I - \overline{F} \otimes F. \tag{71}$$

Recovering R

We have observed that the huge matrix in the Kronecker-type representation (and thus its inverse) is lower triangular.

Recovering R

We have observed that the huge matrix in the Kronecker-type representation (and thus its inverse) is lower triangular.

We could use forward substitution to compute elements of R. We remark that the inverse of a lower triangular matrix is "nested", in the sense that principal submatrices of the inverse are the inverses of the corresponding principal submatrices of the original matrix.

Recovering R

We have observed that the huge matrix in the Kronecker-type representation (and thus its inverse) is lower triangular.

We could use forward substitution to compute elements of R. We remark that the inverse of a lower triangular matrix is “nested”, in the sense that principal submatrices of the inverse are the inverses of the corresponding principal submatrices of the original matrix.

We can not that simply reconstruct the generators of the successive submatrices constructed by the Jacobi-style Cholesky or LDLH decomposition.

Recovering R

We have observed that the huge matrix in the Kronecker-type representation (and thus its inverse) is lower triangular.

We could use forward substitution to compute elements of R. We remark that the inverse of a lower triangular matrix is “nested”, in the sense that principal submatrices of the inverse are the inverses of the corresponding principal submatrices of the original matrix.

We can not that simply reconstruct the generators of the successive submatrices constructed by the Jacobi-style Cholesky or LDLH decomposition.

The generalization can be based on some observations using a few related block matrix decompositions.

Fundamental properties

Displacement structure is preserved under inversion: there exists a full rank matrix $H \in \mathbb{C}^{r\times n}$ such that

$$R^{-1} - F^H R^{-1} F = H^H J H. \tag{72}$$

Fundamental properties

Displacement structure is preserved under inversion: there exists a full rank matrix $H \in \mathbb{C}^{r\times n}$ such that

$$R^{-1} - F^H R^{-1} F = H^H J H. \tag{72}$$

Proof: The block matrix triangular decompositions

$$\begin{pmatrix} R & F \\ F^H & R^{-1} \end{pmatrix} = \begin{pmatrix} I & O \\ F^H R^{-1} & I \end{pmatrix} \begin{pmatrix} R & O \\ O & R^{-1} - F^H R^{-1} F \end{pmatrix} \begin{pmatrix} I & O \\ F^H R^{-1} & I \end{pmatrix}^H \tag{73}$$

$$= \begin{pmatrix} I & FR \\ O & I \end{pmatrix} \begin{pmatrix} R - FRF^H & O \\ O & R^{-1} \end{pmatrix} \begin{pmatrix} I & FR \\ O & I \end{pmatrix}^H \tag{74}$$

Fundamental properties

Displacement structure is preserved under inversion: there exists a full rank matrix $H \in \mathbb{C}^{r\times n}$ such that

$$R^{-1} - F^H R^{-1} F = H^H J H. \tag{72}$$

Proof: The block matrix triangular decompositions

$$\begin{pmatrix} R & F \\ F^H & R^{-1} \end{pmatrix} = \begin{pmatrix} I & O \\ F^H R^{-1} & I \end{pmatrix} \begin{pmatrix} R & O \\ O & R^{-1} - F^H R^{-1} F \end{pmatrix} \begin{pmatrix} I & O \\ F^H R^{-1} & I \end{pmatrix}^H \tag{73}$$

$$= \begin{pmatrix} I & FR \\ O & I \end{pmatrix} \begin{pmatrix} R - FRF^H & O \\ O & R^{-1} \end{pmatrix} \begin{pmatrix} I & FR \\ O & I \end{pmatrix}^H \tag{74}$$

show that by Sylvester's law of inertia

$$\text{inertia}(R^{-1} - F^H R^{-1} F) = \text{inertia}(R - FRF^H). \tag{75}$$

Fundamental properties

We partition

$$R = \begin{pmatrix} R_{11} & R_{12} \\ R_{21} & R_{22} \end{pmatrix} \quad \text{and} \quad F = \begin{pmatrix} F_{12} & O \\ F_{21} & F_{22} \end{pmatrix}. \tag{76}$$

Fundamental properties

We partition

$$R = \begin{pmatrix} R_{11} & R_{12} \\ R_{21} & R_{22} \end{pmatrix} \quad \text{and} \quad F = \begin{pmatrix} F_{12} & O \\ F_{21} & F_{22} \end{pmatrix}. \tag{76}$$

Schur complementation preserves displacement structure: Let the Schur complement be denoted by $S_{22} := R_{22} - R_{21}R_{11}^{-1}R_{12}$.

Fundamental properties

We partition

$$R = \begin{pmatrix} R_{11} & R_{12} \\ R_{21} & R_{22} \end{pmatrix} \quad \text{and} \quad F = \begin{pmatrix} F_{12} & O \\ F_{21} & F_{22} \end{pmatrix}. \tag{76}$$

Schur complementation preserves displacement structure: Let the Schur complement be denoted by $S_{22} := R_{22} - R_{21}R_{11}^{-1}R_{12}$. Then

$$\operatorname{rank}(R_{11} - F_{11}R_{11}F_{11}^H) \leqslant \operatorname{rank}(R - FRF^H), \tag{77}$$

$$\operatorname{rank}(S_{22} - F_{22}S_{22}F_{22}^H) \leqslant \operatorname{rank}(R - FRF^H). \tag{78}$$

Fundamental properties

We partition

$$R = \begin{pmatrix} R_{11} & R_{12} \\ R_{21} & R_{22} \end{pmatrix} \quad \text{and} \quad F = \begin{pmatrix} F_{12} & O \\ F_{21} & F_{22} \end{pmatrix}. \tag{76}$$

Schur complementation preserves displacement structure: Let the Schur complement be denoted by $S_{22} := R_{22} - R_{21}R_{11}^{-1}R_{12}$. Then

$$\operatorname{rank}(R_{11} - F_{11}R_{11}F_{11}^H) \leqslant \operatorname{rank}(R - FRF^H), \tag{77}$$

$$\operatorname{rank}(S_{22} - F_{22}S_{22}F_{22}^H) \leqslant \operatorname{rank}(R - FRF^H). \tag{78}$$

Proof: this follows upon the observation that the inverse of the Schur complement is the lower block in the accordingly partitioned inverse of R.

Fundamental properties

We partition

$$R = \begin{pmatrix} R_{11} & R_{12} \\ R_{21} & R_{22} \end{pmatrix} \quad \text{and} \quad F = \begin{pmatrix} F_{12} & O \\ F_{21} & F_{22} \end{pmatrix}. \tag{76}$$

Schur complementation preserves displacement structure: Let the Schur complement be denoted by $S_{22} := R_{22} - R_{21}R_{11}^{-1}R_{12}$. Then

$$\operatorname{rank}(R_{11} - F_{11}R_{11}F_{11}^H) \leqslant \operatorname{rank}(R - FRF^H), \tag{77}$$

$$\operatorname{rank}(S_{22} - F_{22}S_{22}F_{22}^H) \leqslant \operatorname{rank}(R - FRF^H). \tag{78}$$

Proof: this follows upon the observation that the inverse of the Schur complement is the lower block in the accordingly partitioned inverse of R.

By the first result

$$\operatorname{rank}(S_{22} - F_{22}S_{22}F_{22}^H) = \operatorname{rank}(S_{22}^{-1} - F_{22}^H S_{22}^{-1} F_{22}). \tag{79}$$

Fundamental properties

We now use the Jacobi-style decompositions

$$R = LD^{-1}L^H, \quad U \equiv L^{-H}D, \quad R^{-1} = UD^{-1}U^H. \tag{80}$$

Fundamental properties

We now use the Jacobi-style decompositions

$$R = LD^{-1}L^H, \quad U \equiv L^{-H}D, \quad R^{-1} = UD^{-1}U^H. \tag{80}$$

Key Array Equation: there exists $\Omega \in \mathbb{C}^{2n\times 2n}$ such that

$$\begin{pmatrix} FL & G \\ U & O \end{pmatrix} \Omega = \begin{pmatrix} L & O \\ F^H U & H^H \end{pmatrix}, \qquad \Omega(D^{-1} \oplus J)\Omega = (D^{-1} \oplus J). \tag{81}$$

Fundamental properties

We now use the Jacobi-style decompositions

$$R = LD^{-1}L^H, \quad U \equiv L^{-H}D, \quad R^{-1} = UD^{-1}U^H. \tag{80}$$

Key Array Equation: there exists $\Omega \in \mathbb{C}^{2n\times 2n}$ such that

$$\begin{pmatrix} FL & G \\ U & O \end{pmatrix} \Omega = \begin{pmatrix} L & O \\ F^H U & H^H \end{pmatrix}, \qquad \Omega(D^{-1} \oplus J)\Omega = (D^{-1} \oplus J). \tag{81}$$

Proof:

$$\begin{pmatrix} R & F \\ F^H & R^{-1} \end{pmatrix} = \begin{pmatrix} FL & G \\ U & O \end{pmatrix} \begin{pmatrix} D^{-1} & O \\ O & J \end{pmatrix} \begin{pmatrix} FL & G \\ U & O \end{pmatrix}^H \tag{82}$$

$$= \begin{pmatrix} L & O \\ F^H U & H^H \end{pmatrix} \begin{pmatrix} D^{-1} & O \\ O & J \end{pmatrix} \begin{pmatrix} L & O \\ F^H U & H^H \end{pmatrix}^H. \tag{83}$$

Fundamental properties

We now use the Jacobi-style decompositions

$$R = LD^{-1}L^H, \quad U \equiv L^{-H}D, \quad R^{-1} = UD^{-1}U^H. \tag{80}$$

Key Array Equation: there exists $\Omega \in \mathbb{C}^{2n\times 2n}$ such that

$$\begin{pmatrix} FL & G \\ U & O \end{pmatrix} \Omega = \begin{pmatrix} L & O \\ F^H U & H^H \end{pmatrix}, \qquad \Omega(D^{-1} \oplus J)\Omega = (D^{-1} \oplus J). \tag{81}$$

Proof:

$$\begin{pmatrix} R & F \\ F^H & R^{-1} \end{pmatrix} = \begin{pmatrix} FL & G \\ U & O \end{pmatrix} \begin{pmatrix} D^{-1} & O \\ O & J \end{pmatrix} \begin{pmatrix} FL & G \\ U & O \end{pmatrix}^H \tag{82}$$

$$= \begin{pmatrix} L & O \\ F^H U & H^H \end{pmatrix} \begin{pmatrix} D^{-1} & O \\ O & J \end{pmatrix} \begin{pmatrix} L & O \\ F^H U & H^H \end{pmatrix}^H. \tag{83}$$

We restrict interest to the leading block row.

Outline

Sketch of an algorithm

The key array equation looks as

$$\begin{pmatrix} FL & G \end{pmatrix} \Omega = \begin{pmatrix} L & O \end{pmatrix}, \qquad \Omega(D^{-1} \oplus J)\Omega = (D^{-1} \oplus J). \tag{84}$$

Sketch of an algorithm

The key array equation looks as

$$\begin{pmatrix} FL & G \end{pmatrix} \Omega = \begin{pmatrix} L & O \end{pmatrix}, \qquad \Omega(D^{-1} \oplus J)\Omega = (D^{-1} \oplus J). \tag{84}$$

The algorithm works recursively. We use a $D^{-1} \oplus J$-unitary matrix Ω_0 to obtain a partial triangularization like

$$\begin{pmatrix} FL & G \end{pmatrix} \Omega_0 = \begin{pmatrix} l_0 & o^T & o^T \\ & F_1 L_1 & G_1 \end{pmatrix}. \tag{85}$$

Sketch of an algorithm

The key array equation looks as

$$\begin{pmatrix} FL & G \end{pmatrix} \Omega = \begin{pmatrix} L & O \end{pmatrix}, \qquad \Omega(D^{-1} \oplus J)\Omega = (D^{-1} \oplus J). \tag{84}$$

The algorithm works recursively. We use a $D^{-1} \oplus J$-unitary matrix Ω_0 to obtain a partial triangularization like

$$\begin{pmatrix} FL & G \end{pmatrix} \Omega_0 = \begin{pmatrix} l_0 & o^T & o^T \\ & F_1 L_1 & G_1 \end{pmatrix}. \tag{85}$$

Then we have to show that with $R_1 = L_1 D_1^{-1} L_1^H$, $D_1 = \operatorname{diag}(d_1, \dots, d_n)$,

$$R_1 - F_1 R_1 F_1^H = G_1 J G_1^H. \tag{86}$$

Sketch of an algorithm

The key array equation looks as

$$\begin{pmatrix} FL & G \end{pmatrix} \Omega = \begin{pmatrix} L & O \end{pmatrix}, \qquad \Omega(D^{-1} \oplus J)\Omega = (D^{-1} \oplus J). \tag{84}$$

The algorithm works recursively. We use a $D^{-1} \oplus J$-unitary matrix Ω_0 to obtain a partial triangularization like

$$\begin{pmatrix} FL & G \end{pmatrix} \Omega_0 = \begin{pmatrix} l_0 & o^T & o^T \\ & F_1 L_1 & G_1 \end{pmatrix}. \tag{85}$$

Then we have to show that with $R_1 = L_1 D_1^{-1} L_1^H$, $D_1 = \operatorname{diag}(d_1, \dots, d_n)$,

$$R_1 - F_1 R_1 F_1^H = G_1 J G_1^H. \tag{86}$$

We suppose this is to be correct and proceed to compute

$$\begin{pmatrix} FL & G \end{pmatrix} \Omega_0 \Omega_1 \cdots \Omega_{i-1} = \begin{pmatrix} \hat{L}_i & O & O \\ \tilde{L}_i & F_i L_i & G_i \end{pmatrix}. \tag{87}$$

Sketch of an algorithm

It remains to prove that in the ith step with $R_i = L_i D_i^{-1} L_i^H$,

$$R_i - F_i R_i F_i^H = G_i J G_i^H \tag{88}$$

holds true.

Sketch of an algorithm

It remains to prove that in the ith step with $R_i = L_i D_i^{-1} L_i^H$,

$$R_i - F_i R_i F_i^H = G_i J G_i^H \tag{88}$$

holds true.

It follows immediately from equating second block rows,

$$\begin{aligned}\begin{pmatrix}\tilde{L}_i & F_i L_i & G_i\end{pmatrix}\left(\hat{D}_i^{-1} \oplus D_i^{-1} \oplus J\right)\begin{pmatrix}\tilde{L}_i & F_i L_i & G_i\end{pmatrix}^H &= \\ \begin{pmatrix}\tilde{L}_i & L_i & O\end{pmatrix}\left(\hat{D}_i^{-1} \oplus D_i^{-1} \oplus J\right)\begin{pmatrix}\tilde{L}_i & L_i & O\end{pmatrix}^H&,\end{aligned} \tag{89}$$

Sketch of an algorithm

It remains to prove that in the ith step with $R_i = L_i D_i^{-1} L_i^H$,

$$R_i - F_i R_i F_i^H = G_i J G_i^H \tag{88}$$

holds true.

It follows immediately from equating second block rows,

$$\begin{aligned} &\begin{pmatrix} \tilde{L}_i & F_i L_i & G_i \end{pmatrix} \left(\hat{D}_i^{-1} \oplus D_i^{-1} \oplus J\right) \begin{pmatrix} \tilde{L}_i & F_i L_i & G_i \end{pmatrix}^H = \\ &\qquad \begin{pmatrix} \tilde{L}_i & L_i & O \end{pmatrix} \left(\hat{D}_i^{-1} \oplus D_i^{-1} \oplus J\right) \begin{pmatrix} \tilde{L}_i & L_i & O \end{pmatrix}^H, \end{aligned} \tag{89}$$

that

$$F_i (L_i D_i^{-1} L_i^H) F_i^H + G_i J G_i^H = (L_i D_i^{-1} L_i^H). \tag{90}$$

Sketch of an algorithm

It remains to prove that in the ith step with $R_i = L_i D_i^{-1} L_i^H$,

$$R_i - F_i R_i F_i^H = G_i J G_i^H \tag{88}$$

holds true.

It follows immediately from equating second block rows,

$$\begin{aligned} \begin{pmatrix} \tilde{L}_i & F_i L_i & G_i \end{pmatrix} \left(\hat{D}_i^{-1} \oplus D_i^{-1} \oplus J\right) \begin{pmatrix} \tilde{L}_i & F_i L_i & G_i \end{pmatrix}^H &= \\ \begin{pmatrix} \tilde{L}_i & L_i & O \end{pmatrix} \left(\hat{D}_i^{-1} \oplus D_i^{-1} \oplus J\right) \begin{pmatrix} \tilde{L}_i & L_i & O \end{pmatrix}^H&, \end{aligned} \tag{89}$$

that

$$F_i (L_i D_i^{-1} L_i^H) F_i^H + G_i J G_i^H = (L_i D_i^{-1} L_i^H). \tag{90}$$

Thus, the displacement structure of the Schur complements has been verified.

Sketch of an algorithm

Now we have everything in place to sketch an algorithm.

Sketch of an algorithm

Now we have everything in place to sketch an algorithm.

The Schur complements R_i have displacement structure with respect to F_i,

$$R_i - F_i R_i F_i^H = G_i J G_i^H. \tag{91}$$

Sketch of an algorithm

Now we have everything in place to sketch an algorithm.

The Schur complements R_i have displacement structure with respect to F_i,

$$R_i - F_i R_i F_i^H = G_i J G_i^H. \tag{91}$$

The next column of L is (Jacobi-style scaling) given by $l_i = R_i e_1$ and thus we have to "solve" the linear system

$$(I_{n-i} - \bar{f}_i F_i) R_i e_1 = (I_{n-i} - \bar{f}_i F_i) l_i = G_i J G_i^H e_1. \tag{92}$$

Sketch of an algorithm

Now we have everything in place to sketch an algorithm.

The Schur complements R_i have displacement structure with respect to F_i,

$$R_i - F_i R_i F_i^H = G_i J G_i^H. \tag{91}$$

The next column of L is (Jacobi-style scaling) given by $l_i = R_i e_1$ and thus we have to "solve" the linear system

$$(I_{n-i} - \bar{f}_i F_i) R_i e_1 = (I_{n-i} - \bar{f}_i F_i) l_i = G_i J G_i^H e_1. \tag{92}$$

The next diagonal element is given by $d_i = e_1^T l_i$.

Sketch of an algorithm

Now we have everything in place to sketch an algorithm.

The Schur complements R_i have displacement structure with respect to F_i,

$$R_i - F_i R_i F_i^H = G_i J G_i^H. \tag{91}$$

The next column of L is (Jacobi-style scaling) given by $l_i = R_i e_1$ and thus we have to “solve” the linear system

$$(I_{n-i} - \overline{f}_i F_i) R_i e_1 = (I_{n-i} - \overline{f}_i F_i) l_i = G_i J G_i^H e_1. \tag{92}$$

The next diagonal element is given by $d_i = e_1^T l_i$.

Afterwards we compute the new generator with a $d_i^{-1} \oplus J$-unitary transformation Ω_i (the leading columns remain the same),

$$\begin{pmatrix} l_i & o^T \\ & G_{i+1} \end{pmatrix} = \begin{pmatrix} F_i l_i & G_i \end{pmatrix} \Omega_i. \tag{93}$$

Sketch of an algorithm

Not surprisingly it turns out the d_i^{-1} part of the $d_i^{-1} \oplus J$-unitary transformation Ω_i,

$$\begin{pmatrix} l_i & o^T \\ & G_{i+1} \end{pmatrix} = \begin{pmatrix} F_i l_i & G_i \end{pmatrix} \Omega_i, \tag{94}$$

can be computed explicitely.

Sketch of an algorithm

Not surprisingly it turns out the d_i^{-1} part of the $d_i^{-1} \oplus J$-unitary transformation Ω_i,

$$\begin{pmatrix} l_i & o^T \\ & G_{i+1} \end{pmatrix} = \begin{pmatrix} F_i l_i & G_i \end{pmatrix} \Omega_i, \tag{94}$$

can be computed explicitely.

The transformation takes the form

$$\begin{pmatrix} o^T \\ G_{i+1} \end{pmatrix} = \left(G_i + (\Phi_i - I_{n-i}) G_i \frac{J g_i^H g_i}{g_i J g_i^H} \right) \Theta_i, \tag{95}$$

Sketch of an algorithm

Not surprisingly it turns out the d_i^{-1} part of the $d_i^{-1} \oplus J$-unitary transformation Ω_i,

$$\begin{pmatrix} l_i & o^T \\ & G_{i+1} \end{pmatrix} = \begin{pmatrix} F_i l_i & G_i \end{pmatrix} \Omega_i, \tag{94}$$

can be computed explicitely.

The transformation takes the form

$$\begin{pmatrix} o^T \\ G_{i+1} \end{pmatrix} = \left(G_i + (\Phi_i - I_{n-i}) G_i \frac{J g_i^H g_i}{g_i J g_i^H} \right) \Theta_i, \tag{95}$$

where

$$\Phi_i = (F_i - f_i I_{n-i})(I_{n-i} - \bar{f}_i F_i)^{-1} \tag{96}$$

is the so-called Blaschke-Potapov matrix

Sketch of an algorithm

Not surprisingly it turns out the d_i^{-1} part of the $d_i^{-1} \oplus J$-unitary transformation Ω_i,

$$\begin{pmatrix} l_i & o^T \\ & G_{i+1} \end{pmatrix} = \begin{pmatrix} F_i l_i & G_i \end{pmatrix} \Omega_i, \tag{94}$$

can be computed explicitely.

The transformation takes the form

$$\begin{pmatrix} o^T \\ G_{i+1} \end{pmatrix} = \left(G_i + (\Phi_i - I_{n-i}) G_i \frac{J g_i^H g_i}{g_i J g_i^H} \right) \Theta_i, \tag{95}$$

where

$$\Phi_i = (F_i - f_i I_{n-i})(I_{n-i} - \bar{f}_i F_i)^{-1} \tag{96}$$

is the so-called Blaschke-Potapov matrix, $g_i = e_1^T G_i$ is the leading row of G_i

Sketch of an algorithm

Not surprisingly it turns out the d_i^{-1} part of the $d_i^{-1} \oplus J$-unitary transformation Ω_i,

$$\begin{pmatrix} l_i & o^T \\ & G_{i+1} \end{pmatrix} = \begin{pmatrix} F_i l_i & G_i \end{pmatrix} \Omega_i, \tag{94}$$

can be computed explicitely.

The transformation takes the form

$$\begin{pmatrix} o^T \\ G_{i+1} \end{pmatrix} = \left(G_i + (\Phi_i - I_{n-i}) G_i \frac{J g_i^H g_i}{g_i J g_i^H} \right) \Theta_i, \tag{95}$$

where

$$\Phi_i = (F_i - f_i I_{n-i})(I_{n-i} - \bar{f}_i F_i)^{-1} \tag{96}$$

is the so-called Blaschke-Potapov matrix, $g_i = e_1^T G_i$ is the leading row of G_i and Θ_i is a J-unitary matrix chosen to introduce leading zeros.

An algorithm

Set $G_0 = G$, $i = 0$ and iterate the following: Set $g_i = e_1^T G_i$.

An algorithm

Set $G_0 = G$, $i = 0$ and iterate the following: Set $g_i = e_1^T G_i$.

Solve

$$(I_{n-i} - \bar{f}_i F_i) l_i = G_i J g_i^H. \tag{97}$$

An algorithm

Set $G_0 = G$, $i = 0$ and iterate the following: Set $g_i = e_1^T G_i$.

Solve

$$(I_{n-i} - \bar{f}_i F_i) l_i = G_i J g_i^H. \tag{97}$$

Compute

$$d_i = \frac{g_i J g_i^H}{1 - f_i \bar{f}_i} = e_1^T l_i. \tag{98}$$

An algorithm

Set $G_0 = G$, $i = 0$ and iterate the following: Set $g_i = e_1^T G_i$.

Solve

$$(I_{n-i} - \overline{f}_i F_i) l_i = G_i J g_i^H. \tag{97}$$

Compute

$$d_i = \frac{g_i J g_i^H}{1 - f_i \overline{f}_i} = e_1^T l_i. \tag{98}$$

Compute a new G_{i+1} using a J-unitary matrix Θ_i,

$$\begin{pmatrix} o^T \\ G_{i+1} \end{pmatrix} = \left(G_i + (\Phi_i - I_{n-i}) G_i \frac{J g_i^H g_i}{g_i J g_i^H} \right) \Theta_i, \tag{99}$$

where $\Phi_i = (F_i - f_i I_{n-i})(I_{n-i} - \overline{f}_i F_i)^{-1}$.

C. G. J. Jacobi
Über eine elementare Transformation eines in Bezug auf jedes von zwei Variablen-Systemen linearen und homogenen Ausdrucks.
Journal für die reine und angewandte Mathematik, 53(1), p. 265–270, 1857.

Otto Toeplitz
Die Jacobische Transformation der quadratischen Formen von unendlichvielen Veränderlichen.
Nachrichten von der Gesellschaft der Wissenschaften zu Göttingen, Mathematisch-Physikalische Klasse, 1907.

Otto Toeplitz
Zur Theorie der quadratischen und bilinearen Formen von unendlichvielen Veränderlichen. 1. Teil: Theorie der L-Formen.
Mathematische Annalen, 70(3), p. 351–376, 1911.

J. Schur
Über Potenzreihen, die im Innern des Einheitskreises beschränkt sind.
Journal für die reine und angewandte Mathematik, 147(4), p. 205–232, 1917 und 148(3/4), p. 122–145, 1918.

Thomas Kailath
A theorem of I. Schur and its impact on modern signal processing.
In OT 18, I. Gohberg (ed.), p. 9–30, 1986.

Hanoch Lev-Ari and Thomas Kailath
Triangular factorization of structured Hermitian matrices.
In OT 18, I. Gohberg (ed.), p. 301–324, 1986.

I. Gohberg (ed.)
I. Schur Methods in Operator Theory and Signal Processing.
Operator Theory: Advances and Applications, Vol. 18, Birkhäuser, 1986.

S. Chandrasekaran and Ali H. Sayed
Stabilizing the generalized Schur algorithm.
SIAM J. Matrix Anal. Appl. 17(4), p. 950–983, 1996.

T. Kailath and A. H. Sayed (eds.)
Fast Reliable Algorithms for Matrices with Structure.
SIAM, Philadelphia, 1999.