The Lanczos algorithms and their relations to formal orthogonal polynomials, Padé approximation, continued fractions, and the qd algorithm

With a brief outlook onto:

A completed theory of the unsym. Lanczos and related algorithms

Martin H. Gutknecht
IPS*, ETH Zurich
ETH-Zentrum
CH-8092 Zurich

* Interdisciplinary Project Center for Supercomputing
AIM OF THIS PAPER

* Exposition of basic theoretical facts about — and relations among —
  - Lanczos tridiagonalization: $B_0$ alg., "BC alg" = $B_0(A)$
  - Lanczos bidiagonalization: "BOBC alg."
  - LR transf. for tridiag. matrices: qd alg.
  - Application to solving linear systems
    $B_0 \rightarrow \text{BIORES} \stackrel{\text{normalized}}{\rightarrow} \text{LANCEOS/ORTHORES}$
    $BOBC \rightarrow \text{BIONIN} \stackrel{\text{norm.}}{\rightarrow} \text{BCG} = \text{LANCEOS/ORTHONIN}$
    $BC \rightarrow \text{BIDIR} \stackrel{\text{unnorm.}}{=} \text{LANCEOS/ORTHODIR}$
  - Breakdown conditions for these algorithms
  - Formal orthogonal polynomials and Padé approximation
  - Continued fractions
  - (Bi)conjugate squared algorithms:
    $B_0 \rightarrow \text{BIORES}^2 \stackrel{\text{normalized unnormalized}}{\rightarrow} \text{BCG}$
    $BOBC \rightarrow \text{BIONIN}^2 = \text{CGS}$
    $BC \rightarrow \text{BIDIR}^2$ (2 versions)

* Emphasis is on theoretical relationships, not numerical properties or implementation.

* Preparation for "A completed theory of the unsymmetric Lanczos process and related algorithms", which treats the nongeneric situations and allows us, e.g., to overcome a "curable serious breakdown" of the $B_0$ algorithm: nongeneric $B_0$ alg., ..., nongeneric qd, ...

Others involved in nongeneric case: Gragg '74, Draux '85, Parlett/Taylor/Liu '85, Golub/G. '89, Joubert, Parlett, Boley.
1. The Lanczos biorthogonalization algorithm (BO algorithm)

Given: \( A \in \mathbb{C}^{n \times n}, \ x_0, y_0 \in \mathbb{C}^n \) s.t. \( y_0^H x_0 \neq 0 \).

[or: Hilbert space, \( x_0, y_0 \in H, A \in L(H) \)]

Aim: Generate sequences \( \{x_n\}_n \), \( \{y_n\}_n \) with

\[(1) \quad \begin{align*}
&x_n \in K_n := \text{span} \left( x_0, Ax_0, \ldots, A^n x_0 \right), \\
y_m \in L_m := \text{span} \left( y_0, A^H y_0, \ldots, (A^H)^{m-1} y_0 \right),
\end{align*}
\]

\[(2) \quad y_n^H x_n = \begin{cases} 0 & \text{if } m \neq n \\ d_n & \text{if } m = n \end{cases} \]

and \( v \) as large as possible.

(1) and (2) imply:

\[(3) \quad x_n \in K_n \setminus K_{n-1}, \quad y_n \in L_n \setminus L_{n-1} \]

\[(4) \quad K_n = \text{span} \left( x_0, x_1, \ldots, x_n \right), \quad L_n = \text{span} \left( y_0, y_1, \ldots, y_n \right) \]

\[(5) \quad x_n \perp L_{n-1}, \quad y_n \perp K_{n-1} \]

\[(6) \quad \begin{align*}
&\begin{pmatrix} A x_n = x_{n+1} \tau_{n+1,n} + x_n \tau_{n,n} + x_{n-1} \tau_{n-1,n} + \ldots + x_0 \tau_{0,0} \\ A^H y_n = y_{n+1} \tau_{n+1,n} + y_n \tau_{n,n} + y_{n-1} \tau_{n-1,n} + \ldots + y_0 \tau_{0,0} \end{pmatrix} \begin{pmatrix} y_k^* \\ x_k^* \end{pmatrix} \quad & (K \leq n) \\
&\Rightarrow \tau_{k,n} = 0, \quad \tau_{k,n}' = 0, \quad k = 0, \ldots, n-2 \\
&\tau_{n,n} = \frac{d_n}{\tau_{n-1,n}} \quad \frac{\tau_{n,n-1}}{\tau_{n-1,n-1}} \\
\end{align*}
\]
(7) can be satisfied by choosing

(8) \( \tau_{n-1,n} = \bar{\tau}_{n-1,n}, \quad \tau_{n,n-1} = \bar{\tau}_{n,n-1}. \quad T' = \bar{T} \)

Another frequent choice is

(8') \( \tau_{n-1,n} = \bar{\tau}_{n,n-1}, \quad \tau_{n,n-1} = \bar{\tau}_{n-1,n}. \quad T' = T^H \)

Set \( \alpha_n := \tau_{n,n}, \quad \beta_n := \tau_{n-1,n}, \quad \gamma_n := \tau_{n+1,n}, \)

then (6) becomes

\[
A x_n = x_{n+1} \gamma_n + x_n \alpha_n + x_{n-1} \beta_n, \\
A^H y_n = y_{n+1} \bar{\gamma}_n + y_n \bar{\alpha}_n + y_{n-1} \bar{\beta}_n,
\]

\( n = 0, 1, \ldots, N-2, \)

with \( \beta_0 := 0, \quad x_1 := y_1 := 0. \)

Constructive process for generating \( \{x_n\}, \{y_n\}, \{\alpha_n\}, \{\beta_n\}, \{\gamma_n\} \):

**BO algorithm** [Lanczos 1950]:

**ALG. 1**

For \( n = 0, 1, \ldots \) do:

\[
\alpha_n := y_n^H A x_n / \delta_n, \\
\beta_n := y_{n-1}^H A x_n / \delta_{n-1} = y_{n-1} \delta_n / \delta_{n-1} \quad (\text{if } n > 0), \\
\tilde{x}_{n+1} := A x_n - x_n \alpha_n - x_{n-1} \beta_n, \\
\tilde{y}_{n+1} := A^H y_n - y_n \bar{\alpha}_n - y_{n-1} \bar{\beta}_n, \\
\tilde{\delta}_{n+1} := \tilde{y}_{n+1}^H \tilde{x}_{n+1},
\]

if \( \delta_{n+1} = 0 \), set \( \nu := n + 1 \), stop, else choose \( y_n, \delta_{n+1} \) such that \( y_n^2 \delta_{n+1} = \tilde{\delta}_{n+1} \),

\[
x_{n+1} := \tilde{x}_{n+1} / \tilde{y}_n, \quad y_{n+1} := \tilde{y}_{n+1} / \tilde{y}_n.
\]
Typical choices for $y_n$ and $s_{n+1}$:

(i) $s_{n+1} := 1 \Rightarrow \{x_m\} \{y_m\}$ biorthonormal, $\beta_n = y_{n-1}$.
   However, $y_n$ may be complex even if $A \in \mathbb{R}^{N \times N}$ unless $A$ is Hermitian and $x_0 = y_0$ \(\Rightarrow x_n = y_n\ (tn)\)

(ii) $y_n := 1$ often inappropriate for numerical computations, O.K. for complex theory.

(iii) $y_n := \sqrt{s_{n+1}} \Rightarrow |d_{n+1}| = 1; \Rightarrow y_n, s_{n+1}, x_n, \beta_n \in \mathbb{R}$ if $A \in \mathbb{R}^{N \times N}, x_0, y_0 \in \mathbb{R}^N$.

(iv) $y_n := -\alpha_n - \beta_n$ for solving linear systems of equations. (BIORES)

Types of terminations/breakdowns:

(i) $\tilde{y}_n = 0$ or $\tilde{y}_n = 0$:
   $\Rightarrow K_{n-1}$ or $L_{n-1}$, resp., is invariant subspace of $A$ or $A^T$, resp. [\(\Rightarrow n \leq N\)]

(ii) $\tilde{y}_n \neq 0$, $\tilde{y}_n \neq 0$, but $\tilde{y}_n^H \tilde{x}_n = 0$:
   Serious breakdown. Cannot happen if $A$ is Hermitian and $x_0 = y_0$. Until 1988 the theory was not able to cope with this serious breakdown. Seems to occur very rarely in practice. But may occur approximately and have a devastating effect ($\|x_{n+1}\|$ very large if $s_{n+1} = 1$). There always exists $x_0, y_0$ so that no serious breakdown occurs [Rutishauser, Householder]. In fact, this is w.r.t. $x_0, y_0 \in \mathbb{C}^N$ the generic case [Joubert '90, Parlett '90].
Matrix interpretation

Let

\[ X := (x_0, x_1, \ldots, x_{n-1}), \quad Y := (y_0, y_1, \ldots, y_{n-1}) \]

\(N \times n\) matrices

Then, by (2) and (9), resp.,

\[
Y^T X = D
\]

\[
A^T Y = Y^T + y_0 \overline{y_{n-1}} e_0^T + \sum_{i=1}^{n-1} y_i \overline{y_{n-i}} e_i^T
\]

\(0\) or rank-one

Where

\[
T := \begin{pmatrix}
\alpha_0 & \beta_1 \\
\gamma_0 & \alpha_1 & \beta_2 \\
& \gamma_1 & \alpha_2 & \ddots \\
& & & \ddots & \beta_{n-1} \\
& & & \gamma_{n-2} & \alpha_{n-1}
\end{pmatrix}
\]

\(x_n = 0 \Rightarrow X\) is invariant subspace of \(A\)

\(T\) represents \(A\) in \(X\) w.r.t. basis \(\{x_i\}\)

\(\Rightarrow\) Evauls of \(T\) are Evauls of \(A\)

\(y_n = 0\) : analogous

\(\Rightarrow\) Eval computations : compute Evauls of \(T\)

(numerically very tricky; some eigenvalues tend to appear several times)


N.B.: (1a), (1b) imply

\[ Y^T A X = D T \]
Solution of linear system $A\mathbf{z} = \mathbf{b}$

Choose in BO algorithm $y_n := -\alpha_n - \beta_n$
and $x_0 := b - A z_0$, where $z_0$ is initial approximation of solution. (⇒ Breakdown if $y_n = 0$ !)

Add to the BO algorithm the instruction

$z_{n+1} := - \left( x_n + \alpha_n z_n + \beta_n z_{n-1} \right) / y_n$

Then $x_{n+1} = b - A z_{n+1}$ (residual)

**Proof:**

\[
b - A z_{n+1} = b + \left( A x_n + \alpha_n A z_n + \beta_n A z_{n-1} \right) / y_n
\]

\[
= b + \left( A x_n + \alpha_n (b - x_n) + \beta_n (b - x_{n-1}) \right) / y_n
\]

Induction

\[
x_{n+1}
\]

In the case of a Hermitian system the BO algorithm with (12) becomes the 3-term recurrence version of the conjugate gradient method (CG algorithm) often called **ORTHORES** today or, briefly, **ORES**.

Matrix interpretation of (12): Let $\mathbf{Z} := (z_0, z_1, \ldots)$, then

\[
(12') \quad \mathbf{X} = - \mathbf{Z}^T
\]
The consistency condition $\eta = -\alpha_n - \beta_n$ of BIORES may cause a breakdown due to $\eta_n = 0$ at $n \leq N$.

Let $\dot{\nu}$ denote the breakdown index of BIORES, i.e.,
\[ \dot{\nu} = 0 \text{ or } \eta_{\dot{\nu}} = 0 \]

Clearly, $\dot{\nu} \leq \nu$ (breakdown index of BO alg.)

Can one avoid the breakdown due to $\eta_n = 0$?

**Unnormalized BIORES [G. '90 COPPER]**

Same alg. as (normalized) BIORES, except that $\eta_n \neq 0$ is chosen arbitrarily.

\[ \Rightarrow \text{Breakdown index } \nu \text{ (same as BO alg.)} \]

However, what is then the meaning of approximants $z_n$?

**Lemma** Let $\dot{\eta}_0 = 1$ and
\[ \dot{\eta}_{n+1} := - (\alpha_n \eta_n + \beta_n \dot{\eta}_{n-1}) / \eta_n, \quad n = 0, 1, \ldots, \nu \]

Then $x_n$ and $z_n$ are in unnormalized BIORES related by
\[ x_n = b \dot{\eta}_n - A \dot{z}_n \]

**Proof.**
\[ (b \dot{\eta}_{n+1} - A \dot{z}_{n+1}) \eta_n = -b(\alpha_n \dot{\eta}_n + \beta_n \dot{\eta}_{n-1}) + A x_n + A \dot{z}_{n-1} \eta_n + A \dot{z}_{n-1} \eta_n \]
\[ = A x_n - \dot{\eta}_n \dot{\eta}_n x_n - \dot{\eta}_{n-1} \beta_n = x_{n+1} \eta_n \]

\[ \square \]
2. The Lanczos biconjugate gradient algorithm (BCG algorithm)

Just another way to generate the same sequences \( \{x_n\}, \{y_n\} \) as in \( \S 1 \).

Lanczos [1952] called it "the complete algorithm for minimized iterations."

Main application: Solution of linear equations. Popularized by Fletcher [1976].

**BOBC algorithm** (≈ BCG, but not adapted to linear systems)

Given \( A \in \mathbb{C}^{N \times N} \), \( x_0 \in \mathbb{C}^N \), \( y_0 \in \mathbb{C}^N \) s.t. \( y_0^* x_0 \neq 0 \),
set \( u_0 := x_0 \), \( v_0 := y_0 \).
For \( n = 0, 1, \ldots \) do:

\[
\psi_n := y_n^* A u_n / \delta_n = v_n^* A u_n / \delta_n,
\]

if \( \psi_n = 0 \), set \( i := n \) and **stop**; otherwise
choose \( \gamma_n \neq 0 \),

\[
x_{n+1} := (A u_n - x_n \psi_n) / \gamma_n,
\]
\[
y_{n+1} := (A^H v_n - y_n \overline{\psi_n}) / \overline{\gamma_n},
\]
\[
\delta_{n+1} := y_{n+1}^* x_{n+1}, \quad \psi_{n+1} := \delta_{n+1} \gamma_n / (\delta_n \overline{\gamma_n})
\]

if \( \delta_{n+1} = 0 \) (\( \Rightarrow \psi_{n+1} = 0 \)), set \( i := n+1 \), **stop**; otherwise

\[
u_{n+1} := x_{n+1} - u_n \psi_{n+1},
\]
\[
v_{n+1} := y_{n+1} - v_n \overline{\psi_{n+1}}.
\]

N.B.: Two different possible causes for breakdown
**Theorem 1:** Assume \( \delta_n \neq 0, \gamma_n \neq 0 \) \((n = 0, \ldots, n-1)\).

Then, for \( m, n = 0, \ldots, n-1 \):

\[
y_m^H x_n = \begin{cases} 
0 & \text{if } m \neq n \\
\delta_n & \text{if } m = n
\end{cases}
\]

\( y_m^H A_n u_n = \begin{cases} 
0 & \text{if } m \neq n \\
\delta_n \gamma_n & \text{if } m = n
\end{cases} \)

**Matrix interpretation:**

For simplicity, assume again \((n = N)\). Set \( X, Y \) as before, \( U := (u_0, \ldots, u_{n-1}), V := (v_0, \ldots, v_{n-1}) \).

Then

\[
AU = XL^+, \quad A^HV = YL^+, \quad X = UR, \quad Y = VR
\]

Where

\[
L := 
\begin{pmatrix}
\gamma_0 & \gamma_1 & \cdots & \gamma_{n-1} \\
0 & \gamma_1 & \cdots & \gamma_{n-2} \\
& 0 & \ddots & \vdots \\
& & 0 & \gamma_1 \\
& & & 0
\end{pmatrix}, \quad R := 
\begin{pmatrix}
1 & \gamma_1 & \gamma_2 & \cdots & \gamma_{n-1} \\
1 & 1 & \gamma_2 & \cdots & \gamma_{n-2} \\
& 1 & 1 & \ddots & \vdots \\
& & 1 & 1 & \gamma_1 \\
& & & 1 & 1
\end{pmatrix}
\]

By eliminating \( X \) and \( Y \) or \( U \) and \( V \), respectively:

\[
AU = UT_1, \quad A^HV = V T_1, \quad AX = XT, \quad A^HY = YT
\]

Where

\[
T_1 := RL, \quad T := LR.
\]

\( \Rightarrow \) BOBC method makes implicit use of the \( LL \) decomposition of \( T \). (Hence, it may break down earlier.)
Solution of linear system $Az = b$

Choose in BOBC algorithm $y_n := y_n$ [consistency cond.] and $x_0 := b - Az_0$, where $z_0$ is initial approximation.
Add to the BOBC algorithm the instruction

$$z_{n+1} := z_n + u_n/Y_n, \quad n = 0, 1, \ldots$$

Then $x_{n+1} = b - A\bar{z}_{n+1}$ (residual)

**Proof:** $b - A\bar{z}_{n+1} = b - A\bar{z}_n - A u_n / y_n = x_n - A u_n / y_n = x_{n+1}$

In the case of a Hermitian system the BCG alg. becomes the (two-term recurrence version of) the CG algorithm: ORTHOMIN or OMN.

**Relation to LR and qd algorithms**

$$T = LR \implies T_1 = RL$$

is one step of the LR algorithm applied to the tridiagonal matrix $T$ (=progressive qd algorithm)

$$\implies T_1 = RTR^{-1}, \text{ i.e. } T_1 \text{ is similar to } T$$

Typically, the sequence

$$T \rightarrow T_1 \rightarrow T_2 \rightarrow T_3 \rightarrow \ldots$$

converges to diagonal matrix (containing the eigenvalues of $T$) [Hadamard, de Montessus de Ballore].
BO ↔ BCG parameter relations; qd formulas

\[
T = \begin{pmatrix}
\alpha_0 & \beta_1 \\
\gamma_0 & \alpha_1 & \beta_2 \\
\gamma_1 & \alpha_2 & \ldots
\end{pmatrix}, \quad T_1 = \begin{pmatrix}
\alpha'_0 & \beta'_1 \\
\gamma'_0 & \alpha'_1 & \beta'_2 \\
\gamma'_1 & \alpha'_2 & \ldots
\end{pmatrix}
\]

\[
L_1 = \begin{pmatrix}
\gamma'_0 \\
\gamma'_0 & \gamma'_1 \\
\gamma'_1 & \gamma'_2 & \ldots
\end{pmatrix}, \quad R_1 = \begin{pmatrix}
1 & \gamma'_1 \\
1 & \gamma'_2 & \ldots
\end{pmatrix}
\]

\[
T = LR \quad \sim \quad T_1 = RLT_1
\]

\[
\alpha_0 = \gamma_0 \\
\beta_1 = \gamma_0 \gamma_1
\]

\[
\alpha_n = \gamma_n + \gamma_{n-1} \gamma_n \\
\beta_{n+1} = \gamma_n \gamma_{n+1}
\]

\[
n = 0, 1, \ldots \quad (\gamma_0 = 0)
\]

\[
\gamma_0 = \alpha_0 \\
\gamma_1 = \beta_1 / \gamma_0
\]

\[
\gamma_n = \alpha_n - \gamma_{n-1} \gamma_n \\
\gamma_{n+1} = \beta_{n+1} / \gamma_n
\]

\[
n = 0, 1, \ldots
\]

\[
\text{rhombus rules of qd alg.}
\]

\[
\gamma_n + \gamma_n \gamma_{n+1} = \alpha'_n = \gamma'_0 \\
\gamma_2 \gamma_2 = \beta'_1 = \gamma'_0 \gamma'_1
\]

\[
\gamma'_n = \alpha'_n + \alpha'_n \gamma_{n+1} \\
\gamma'_{n+1} = \beta'_{n+1} = \gamma'_{n} \gamma'_{n+1}
\]

\[
(n = 0, 1, \ldots)
\]

(usually \( j_n = 1 \) (th) in qd alg.)
BIOMIN has the same breakdown index \( \hat{v} \) as the BOBC algorithm and, as can be shown, normalized BIORES, while for unnormalized BIORES and the BO algorithm the breakdown is at \( \nu \geq \hat{v} \).

Does it help to define an unnormalized version of BIOMIN?

**Unnormalized BIOMIN**

Choose in BOBC algorithm \( y_0 \neq 0 \) arbitrarily, and set \( x_0 = b - Az_0, \hat{y}_0 = 1, \)

\[
\begin{align*}
\tilde{x}_{n+1} &= -\tilde{x}_n \frac{y_n}{\hat{y}_n} - \tilde{u}_n / \hat{y}_n \\
\tilde{y}_{n+1} &= -\tilde{y}_n \frac{y_n}{\hat{y}_n}
\end{align*}
\]

Then, again, \( x_n = b \tilde{y}_n - A \tilde{x}_n, \ n = 0, 1, \ldots, \hat{v} \). But the BOBC algorithm, and hence also unnormalized BIOMIN still break down at \( \hat{v} \).

However, there is another possibility to reduce the chance of breakdown.

Recall:

\[
V_m^\dagger A u_n = \begin{cases} 0 & \text{if } m \neq n \\ \delta_n y_n & \text{if } m = n \end{cases} \Rightarrow \{v_m\}\{u_n\} \text{ are bi-conjugate}
\]

\[
\{v_m\}\{u_n\} \text{ are } A^{-1}\text{-orthogonal}
\]

Hence, the two sequences can be generated by the BO algorithm using the "A-norm."

(\( \Rightarrow \) new parameters \( \tilde{x}_n, \beta_n, \tilde{y}_n' \))

Need additional formulas for \( x_{n+1} \) and \( \tilde{u}_{n+1} \).
"BC algorithm" (= BO algorithm with "A-norm")

Given $u_0, v_0 \in \mathbb{C}^n$ with $v_0^* A u_0 \neq 0$, let $\beta_0 := 0$
and, for $n = 0, 1, 2, \ldots$

$$
x_n' := (A^* v_n')^* A u_n' / \delta_n'
$$

$$
\beta_n' := v_n^* A u_n \cdot v_{n-1} / \delta_{n-1}' = \beta_{n-1}' \delta_n' / \delta_{n-1}' \quad (\text{if } n > 0)
$$

$$
\tilde{u}_{n+1} := A u_n' - u_n' x_n' - u_{n-1}' \beta_n'
$$

$$
\tilde{v}_{n+1} := A^* v_n' - v_n' \tilde{u}_n' - v_{n-1}' \beta_n'
$$

$$
\delta_{n+1}' := \tilde{v}_{n+1}^* A \tilde{u}_{n+1}
$$

If $\delta_{n+1}' = 0$, set $n' := n+1$ and stop; otherwise, choose $y_n'$ and $\delta_{n+1}'$ such that $(y_n')^2 \delta_{n+1}' = \delta_{n+1}'$, and set

$$
u_{n+1}' := \tilde{u}_{n+1} / y_n', \quad \tilde{v}_{n+1}' := \tilde{v}_{n+1} / y_n'.
$$

**Biodir** (="BC alg. for solving linear systems")

Choose initial approx. $z_0$ and set $u_0' := x_0 := b - A z_0$.

Apply BC algorithm, computing additionally

$$
\omega_n' := v_n'^* x_n' / \delta_n'
$$

$$
z_{n+1} := z_n + u_n' \omega_n'
$$

$$
x_{n+1} := x_n - A u_n' \omega_n'
$$

Breakdown only when $\delta_{n+1}' = 0$, i.e., when BC alg. breaks down. Breakdown index $y'$

$\omega_n' = 0$ is no reason for breakdown, although then $z_{n+1} = z_n$ and $x_{n+1} = x_n$. But in general $u_{n+1}$ will be nonzero and will help span a larger Krylov space, so that $z_{n+2} \neq z_n$, $x_{n+2} \neq x_n$. 

The following result on the breakdown conditions of the five methods for solving a linear system can be proven:

**THM.** Assume that the same initial approximation \( z_0 \) and the same initial vectors \( u_0 := u_0' := x_0 := b - A z_0 \) and \( v_0 := v_0' := y_0 \) are chosen in each algorithm.

Then \( \nu = \min \{ \nu, \nu' \} \)

The following four conditions are equivalent:

(i) \( \omega_m \neq 0 \) (m<n), \( \nu' = 0 \) in BIODIR

(ii) \( \delta_n = 0 \) in BLOMIN

(iii) \( \delta_n = 0 \) in BLORES and \( \nu = \nu' \)

(iv) \( n = \nu = \nu' \)

Likewise, the following four conditions are equivalent:

(iii) \( \delta_n' = 0 \) in BIODIR and \( \nu = \nu' \)

(ii) \( \mu_n = 0 \) in BLOMIN

(iii) \( \alpha_n + \beta_n = 0 \) in BLORES

(iv) \( n = \nu = \nu' \)

Except (i) for BIODIR and (iii)" for unnormalized BLORES, all these conditions are breakdown conditions for the respective algorithms.
3. The relation to formal orthogonal polynomials and Padé approximation

From definitions of BO and BOBC algorithms one sees immediately that

\[ x_n = s_n(A) x_0, \quad y_n = \overline{s}_n(A^H) y_0, \]
\[ u_n = \sigma_n(A) u_0, \quad v_n = \overline{\sigma}_n(A^H) v_0. \]

where \( s_n \) and \( \overline{s}_n \) are polynomials satisfying the recurrences

\[ s_0(\xi) := 1, \quad s_{n+1}(\xi) := \left[ \xi s_n(\xi) - s_n(\xi) y_n \right] / y_n, \]
\[ \sigma_0(\xi) := 1, \quad \sigma_{n+1}(\xi) := s_{n+1}(\xi) - \sigma_n(\xi) v_{n+1} \]

"polynomial form of the BOBC algorithm (BCG)"

and

\[ s_0(\xi) := 1, \quad s_2(\xi) := (\xi - \alpha_0) s_0(\xi) / y_0, \]
\[ s_{n+2}(\xi) := [ (\xi - \alpha_n) s_n(\xi) - \beta_n \cdot s_{n-1}(\xi) ] / y_n. \]

"polynomial form of the BO algorithm"

\( s_n \) and \( \sigma_n \) have both exact degree \( n \) and leading coefficient \( (\gamma_0 \gamma_2 \cdots \gamma_{2n})^{-1} \).

When used for solving linear systems both algorithms are "polynomial acceleration methods" (semiliterative methods). Necessarily \( s_n(0) = 1 \) (th). \( \Rightarrow \) Need \( s_n + \alpha_n + \beta_n = 0, \quad y_n = -\beta_n \) there.
Thm. 1 turns into:

\( y_0^H \varphi_m(A) \varphi_n(A) x_0 = \delta_{mn} \delta_n, \) \textit{biorthogonality}

\( y_0^H \tau_m(A) A \tau_n(A) x_0 = \delta_{mn} \delta_n \gamma_n. \) \textit{biconjugacy}

Define linear functionals \( \Phi \) and \( \Phi_1 \) on \( \mathcal{P} \) by

\[
\Phi(\varsigma^k) := \mu_k, \quad \Phi_1(\varsigma^k) := \mu_{k+1}, \quad k \in \mathbb{N}
\]

where

\[
\mu_k := y_0^H A^k x_0 \quad \text{Schwarz constants,}
\]

\[
\{\mu_k\}: \text{impulse response}
\]

Then Thm. 1 turns into

\[
\Phi(\varphi_m \varphi_n) = \delta_{mn} \delta_n, \quad (0 \leq m < n),
\]

\[
\Phi_1(\tau_m \tau_n) = \delta_{mn} \delta_n \gamma_n, \quad (0 \leq n < n),
\]

which, by the linearity of \( \Phi \) and \( \Phi_1 \) is equivalent to

\[
\Phi(\varsigma^m \varphi_n) = \begin{cases}
0 & \text{if } 0 \leq m < n < n' \\
y_0 \cdots y_m \delta_n & \text{if } m = n < n'
\end{cases}
\]

\[
\Phi_1(\varsigma^m \tau_n) = \begin{cases}
0 & \text{if } 0 \leq m < n < n' \\
y_0 \cdots y_m \delta_n \gamma_n & \text{if } m = n < n'
\end{cases}
\]

Hence, \( \{\varphi_n\} \) and \( \{\tau_n\} \) are sequences of formal orthogonal polynomials.

However, \( (\varphi, \sigma) \in \mathcal{P} \times \mathcal{P} \rightarrow \Phi(\varphi \sigma) \in \mathbb{C} \)

does not define an inner product unless

\[
F(\varsigma) := \sum_{k=0}^{\infty} \mu_k \varsigma^{-k+1}
\]

is a Stieltjes function

\( \Leftrightarrow \) \text{A Hermitian pos. def. and } x_0 = y_0. \)
Let
\[ g_n(\xi) = \sum_{k=0}^{n} \xi^k, \quad \sigma_n(\xi) = \sum_{k=0}^{n} \xi^k \]

Then (5) implies
\[
\begin{pmatrix}
\mu_0 & \mu_1 & \cdots & \mu_n \\
\mu_1 & \mu_2 & \cdots & \mu_{n+1} \\
\vdots & \vdots & \ddots & \vdots \\
\mu_n & \mu_{n+1} & \cdots & \mu_{2n-1}
\end{pmatrix}
\begin{pmatrix}
g_0(n) \\
g_1(n) \\
\vdots \\
g_n(n)
\end{pmatrix}
= \begin{pmatrix}
0 \\
0 \\
0 \\
0
\end{pmatrix},
\begin{pmatrix}
\mu_1 & \mu_2 & \cdots & \mu_{n+1} \\
\mu_2 & \mu_3 & \cdots & \mu_{n+2} \\
\vdots & \vdots & \ddots & \vdots \\
\mu_n & \mu_{n+1} & \cdots & \mu_{2n}
\end{pmatrix}
\begin{pmatrix}
\sigma_0(n) \\
\sigma_1(n) \\
\vdots \\
\sigma_n(n)
\end{pmatrix}
= \begin{pmatrix}
0 \\
0 \\
0 \\
0
\end{pmatrix}
\]

These are \( n \times (n+1) \) systems of linear equations, hence nontrivial solutions must exist. (We know that anyway.)

Under our assumption of orthogonality of \( \{g_n\}_{n=0}^{\infty}, \{\sigma_n\}_{n=0}^{\infty} \), it is known that the solution is unique up to a scalar factor and that \( g_0(n) \neq 0, \sigma_0(n) \neq 0 \).

Next, let
\[ F(\xi) := g_0(\xi)(5I-A)^{-1}x_0 = \sum_{k=0}^{\infty} \frac{\mu_k}{\xi^{k+1}} \]

Rational Ft., order \( \leq \) degree of minimal pol.

Then, by (6), there exist \( \tau_n \in \mathbb{P}_{n-1}, \pi_n \in \mathbb{P}_n \) s.t.
\[
F(\xi)g_n(\xi) - \pi_n(\xi) = O(\xi^{-n-1}), \quad \xi F(\xi)\sigma_n(\xi) - \tau_n(\xi) = O(\xi^{-n-1}),
\]
i.e.,
\[
\frac{\pi_n}{g_n} \quad \text{and} \quad \frac{\tau_n}{\sigma_n}
\]
are the \((n,n)\) and \((n+1,n)\) Pade' approximant of \( F \) at \( \xi = 0 \). They lie on two adjacent diagonals of the Pade' table.
The Padé table of $F$ and breakdowns:

![Diagram showing breakdowns]

**What can we say about $F$?**

Consider the Eigenvector decomposition of $A$ (or transform to Jordan form):

$$A W = W \mathbf{D}_2, \quad W = [\omega, \omega_2, ..., \omega_N]$$

**Jordan matrix**

$$A^H W^{-H} = W^{-H} \mathbf{D}_2^H, \quad W^{-H} = [\omega, \omega_2, ..., \omega_N]$$

$$x_0 = \sum_{j=1}^{N} \omega_j \tilde{x}_j, \quad y_0 = \sum_{j=1}^{N} \omega_j \tilde{y}_j$$

$D_2$ diag:

$$M_k = y_0^H A^k x_0 = \sum_{j=1}^{N} \omega_j \tilde{x}_j \tilde{y}_j$$

$$F(\xi) = y_0^H (S^T - A)^{-1} x_0 = \sum_{j=1}^{N} \tilde{x}_j \tilde{y}_j \frac{\xi_j^2}{\xi - \xi_j}$$

$D_2$ non-diag:

$$F(\xi) = \sum_{j=1}^{N} \sum_{i=0}^{\tau(j)} \tilde{x}_j \tilde{y}_j \frac{\xi_j^2}{(\xi - \xi_j)^{i+1}}$$

$\tau(j)$: $j$: index of last column in $D_2$ that belongs to same Jordan subblock as column $j$. 
The Stieltjes procedure

Recall: with $g_n := 1$, then,

\[(S1)\]
\[
\begin{align*}
  g_0 (s) &= 1, \\
  g_1 (s) &= (s - \alpha_0) g_0 \\
  g_{n+1} (s) &= (s - \alpha_n) g_n (s) - \beta_n g_{n-1} (s)
\end{align*}
\]

In Lanczos setting:

\[\Phi (s) := \int g_n^H s (A) x_0 = \sum_{j=1}^{N} \phi (\beta_j) \varphi_j \varphi_j^H \]

if \(A\) diagonalizable

In orthogonal polynomial setting

\[\Phi (s) := \int p (s) \, \text{d}w (s)\]

pos. measure
support \( \subseteq \mathbb{R} \)

If \(A = A^*\), \(g_0 = x_0\):

\[\text{d}w (s) = \sum_{j=1}^{N} 1 \delta (s - \alpha_j)\]

Positiv measure

Dirac \(\delta\) fct.

From (10):

\[
\begin{align*}
  \alpha_n &= \frac{g_n^H A x_n}{g_n^H x_n} = \frac{\Phi (s_n^2)}{\Phi (s_n)} \\
  \beta_n &= \frac{g_n^H x_n}{g_{n-1}^H x_{n-1}} = \frac{\Phi (s_n^2)}{\Phi (s_n^2)}
\end{align*}
\]

Hence,

\[(S2)\]
\[
\begin{align*}
  \alpha_n &= \frac{\int s_n^2 (s) \, \text{d}w}{\int s_n^2 (s) \, \text{d}w} \\
  \beta_n &= \frac{\int s_n^2 (s) \, \text{d}w}{\int s_{n-1}^2 (s) \, \text{d}w}
\end{align*}
\]

\((S1) + (S2) = \text{Stieltjes procedure} [1884] = \text{Lanczos for polyns.}\)
Factorization of the moment matrix

\[ \exists \quad P = \begin{pmatrix} * & * & \cdots & * \\ * & * & \cdots & * \\ \vdots & \vdots & \ddots & \vdots \\ * & * & \cdots & * \end{pmatrix} \quad \text{s.t.} \quad \begin{pmatrix} \phi_0(t) & \phi_1(t) & \cdots & \phi_{n-1}(t) \end{pmatrix} = \begin{pmatrix} 1 & \cdots & 1 \end{pmatrix} P \]

or,

\[ X = (x_0, x_1, \ldots, x_{n-1}) = (x_0, Ax_0, \ldots, A^{n-1}x_0) P \]
\[ Y = (y_0, y_1, \ldots, y_{n-1}) = (y_0, A^H y_0, \ldots, (A^H)^{n-1} y_0) \overline{P} \]

Biorthogonality, i.e. \( Y^H X = D \), yields

\[ P^{-T} D P^{-1} = P^{-T} Y^H X P^{-1} = \begin{pmatrix} y_0^H \\ y_0^H A \\ \vdots \\ y_0^H (A^H)^{n-1} \end{pmatrix} \begin{pmatrix} x_0 \\ Ax_0 \\ \vdots \\ A^{n-1}x_0 \end{pmatrix} = \begin{pmatrix} \mu_0 & \mu_1 & \cdots & \mu_{n-1} \\ \mu_1 & \mu_1 & \cdots & \mu_{n-2} \\ \vdots & \vdots & \ddots & \vdots \\ \mu_{n-1} & \mu_{n-2} & \cdots & \mu_{2n-1} \end{pmatrix} \]

\( n \times n \) moment matrix \( M_0 \) (Hankel)

\[ \Rightarrow \quad P^T M_0 P = D \quad \text{or} \quad M_0^{-1} = P D^{-1} P^T \]

"inverse" LDU dec. \quad UDL dec. of \( M_0^{-1} \)

Basis of some fast \( O(n^2) \) Hankel solvers
4. The relation to continued fractions

Continued fraction:

\[ b_0 + \cfrac{a_1}{b_1 + \cfrac{a_2}{b_2 + \cfrac{a_3}{b_3 + \cdots}} = \cfrac{a_j}{b_j} \quad j = 0 \to K \]

(1)

The *n*th convergent \( W_n \) (or approximant) := "*n*th partial sum"

In our applications, \( a_n \) and \( b_n \) are complex numbers or polynomials. By induction one can prove easily:

**Lemma [Wallin, Euler]**

Define the *n*th numerator \( u_n \) and the *n*th denominator \( v_n \) of (1) by the recurrence

(2)

\[
\begin{align*}
    u_{-1} &= 1, \\
    u_0 &= b_0, \\
    u_n &= b_n u_{n-1} + a_n u_{n-2}, \quad n \geq 1, \\
    v_{-1} &= 0, \\
    v_0 &= 1, \\
    v_n &= b_n v_{n-1} + a_n v_{n-2}, \quad n \geq 1.
\end{align*}
\]

Then

(3)

\[
W_n = b_0 + \sum_{j=0}^{K} \frac{a_j}{b_j} = \frac{u_n}{v_n}
\]

Elimination of all odd [even] indexed \( u_n, v_n \) from (2) gives a new continued fraction, called the even [odd] part of (1). [Cyclic reduction]
Recall:

\[(4) \quad F(5) := y_0^H (5I - A)^{-1} x_0 = \sum_{k=0}^{\infty} \frac{\mu_k}{5^{k+1}}, \quad \mu_k := y_0^H A^k x_0\]

\(F(5)\) can be expanded into a continued fraction in several ways. For example,

\[
F(5) = 0 + \frac{y_0^H 5}{F_1(5)}, \quad \text{where} \quad F_1(5) := (1 + \mu_0 5^{-1} + \mu_1 5^{-2} + \ldots)
\]

\[
= 0 + \frac{\mu_0 5}{1 + \frac{\mu_1 5}{F_2(5)}} = \frac{\mu_0}{5} (F(5) - 0)^{-1}
\]

etc.

The following can be shown by comparing the recurrence formulas:

**THM.** Assume \(\nu = N\).

\[
(i) \quad F(5) = \frac{\mu_0}{5} \left[ \frac{y_0}{1} - \frac{y_1}{5} - \frac{y_2}{1} - \frac{\mu_2}{5} - \frac{\mu_3}{1} - \frac{\mu_4}{5} - \ldots \right]
\]

where \(y_0, y_1, y_2\) are the parameters of the biconjugate gradient method. For this continued fraction there holds, with \(\Gamma_n := y_0 \ldots y_{n-1}\),

\[
(ii) \quad \Gamma_{2n} = \Gamma_n T_{2n}, \quad \Gamma_{2n+1} = \Gamma_n S_{2n+1},
\]

\[
\Gamma_{2n+1} = \Gamma_n T_{2n+1}, \quad \Gamma_{2n+2} = \Gamma_n S_{2n+2},
\]

and \(\Gamma_{2n}\) and \(\Gamma_{2n+1}\) are the \((n-1, n)\) and \((n, n)\) Padé approximants of \(F\) at \(\alpha\), respectively.
where \( \alpha_n, \beta_n, \gamma_n \) are the parameters of the \( \mathcal{B}_0 \) algorithm. For this continued fraction there holds:

\[
\begin{align*}
\upsilon_n &= \Gamma_n \tau_n, \\
\nu_n &= \Gamma_n \eta_n,
\end{align*}
\]

and \( \upsilon_n \) is the \((n-1,n)\) Padé approximant of \( F \) at \( s = \infty \). Hence, the continued fraction in (7) is the even part of the one in (5).

\[
F(s) = \mu_0 + \frac{\mu_1}{5 - \alpha_0} - \frac{\beta_1 \gamma_0}{5 - \alpha_1} - \frac{\beta_2 \gamma_1}{5 - \alpha_2} - \ldots
\]

where

\[
\alpha_n' = \gamma_n + \gamma_n \gamma_{n+1}, \quad \beta_n' = \gamma_{n-1} \gamma_{n+1} \quad (n \geq 0)
\]

and \( \gamma_n \) are the elements of the matrix \( T_1 = RT_1 R^T \).

For this continued fraction there holds

\[
\begin{align*}
\upsilon_n &= \Gamma_n \tau_n, \\
\nu_n &= \Gamma_n 5 \upsilon_n,
\end{align*}
\]

and \( \upsilon_n \) is the \((n,n)\) Padé approximant of \( F \) at \( s = \infty \). Hence, (9) is the odd part of (5).

\[
F(s) = \mu_0 + \frac{\mu_1}{5} - \frac{\gamma_0}{5} - \frac{\gamma_1}{1} - \frac{\gamma_2}{5} - \frac{\gamma_3}{1} - \frac{\gamma_4}{5} - \ldots
\]

where \( \gamma_n \) and \( \gamma_n' \) satisfy (with \( \gamma_1 = 0 \))

\[
\begin{align*}
\alpha_n' &= \gamma_n + \gamma_{n-1} \gamma_n, \\
\beta_n' &= \gamma_n' \gamma_{n+1} \quad (n \geq 0)
\end{align*}
\]
and are the elements of the bidiagonal factors of the \( LL \) decomposition \( T_1 = L_1 R_1 \).
(The diagonal elements of \( L_1 \) are \( \psi_n \), those of \( R \) are 1.) For this continued fraction \( \psi_n \)
and \( \psi_{2n+1} \) are the \((n, n+1)\) and \((n+1, n+1)\) Padé-approximant of \( F \), respectively.

(i) By equating (10) and (13) one has, if \( \psi_n = 1 \) (th):

\[
\psi_n' + \psi_n' = \psi_n + \psi_{n+1}, \quad \psi_n' \psi_{n+1}' = \psi_{n+1} \psi_n' \tag{n \geq 0}
\]

These are the rhombus rules of the qd algorithm.

\[
\text{qd table}
\]

Rubischannais:

\[
\psi_{n-1} \to \psi_n
\]

\[
\psi_n \to \psi_{n+1}
\]
Outlook: Theory including nongeneric situations

\[ F(S) = y'' (S I - A)^{-1} x_0 = \sum_{k=0}^{n} \frac{K_k}{S^{k+1}} \]
linear functional \( \Phi \), \( \Phi (S) = \mu_k \)

- Formal orthogonal polynomials w.r.t. \( \Phi \)
  - definition, properties, conn. w/ Pade'
  - recurrence relation

- Matrix interpretation of recurrence
  - \( AX = XT \), \( T \) block tridiag.
    \[ \alpha_n = \begin{pmatrix} 0 & 1 & 0 & \cdots \\ 1 & \ddots & \ddots & \cdots \\ \vdots & \ddots & \ddots & 1 \\ 0 & \cdots & 1 & 0 \end{pmatrix}, \beta_n = \begin{pmatrix} 0 & * \\ \vdots & \ddots \\ 0 & * \\ 0 & \cdots & 0 & * \end{pmatrix}, \gamma_n = \begin{pmatrix} 0 & * \\ \vdots & \ddots \\ 0 & * \\ 0 & \cdots & 0 & * \end{pmatrix} \]

- Continued fractions
  - "block diagonal": \( P \)-fractions, Chebyshev, Arne Magnus '62
  - "block staircase": staircase \( P \)-fractions, new
  - contraction of staircase \( P \)-fractions, new (generalization of odd/even part)

- qd algorithm (incl. deg. cases), new
  - NGBO algorithm, NGBCG algorithm, NGBCS algorithm, news

- Matrix interpretation of relations between
  - block diagonal - block staircase \( \Rightarrow \) block LU decomp.
  - NGBO - NGBCO \( \Rightarrow \) " UL "
  - diagonals of qd parameters \( \Rightarrow \) " LR algorithm "