Fast algorithms for polynomial-Vandermonde matrices related to quasiseparable matrices

Tom Bella
Department of Mathematics
University of Connecticut
Storrs, CT 06269 USA

Joint work with Yuli Eidelman, Israel Gohberg, Israel Koltracht, & Vadim Olshevsky.

Outline

Fast algorithms for Vandermonde matrices

- Classical Björck-Pereyra & Traub algorithms
- Polynomial-Vandermonde matrices and previous work

A Björck-Pereyra algorithm for Szegö-Vandermonde matrices

- Extension to Szegö-Vandermonde matrices
- Computational speedup
- Numerical illustrations

Fast algorithms for quasiseparable-Vandermonde matrices

- Quasiseparable matrices & special cases
- Björck-Pereyra-like algorithm for quasiseparable matrices
- Recurrence relations
- Traub-like algorithm for quasiseparable matrices

Classical Vandermonde matrices

- **Definition.** For a set of nodes, a **Vandermonde matrix** is defined by

$$x = \{x_1, x_2, \ldots, x_n\}$$
$$\Downarrow$$
$$V(x) = \begin{bmatrix} 1 & x_1 & {x_1}^2 & \cdots & {x_1}^{n-1} \\ 1 & x_2 & {x_2}^2 & \cdots & {x_2}^{n-1} \\ 1 & x_3 & {x_3}^2 & \cdots & {x_3}^{n-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & {x_n}^2 & \cdots & {x_n}^{n-1} \end{bmatrix}$$

- **Note.** The n^2 entries of $V(x)$ are defined by only n parameters. This allows the design of **fast algorithms**.

Fast Björck-Pereyra algorithm

➠ The **Björck-Pereyra algorithm** (1970) is based on the formula

$$V(x)^{-1} = U_1^{-1} \dots U_{n-1}^{-1} L_{n-1}^{-1} \dots L_1^{-1},$$

with

$$U_k^{-1} = \left[\begin{array}{c|ccccc} I_{k-1} & & & & & \\ \hline & 1 & -x_k & & & \\ & & 1 & \ddots & & \\ & & & \ddots & -x_k & \\ & & & & 1 & \end{array}\right]$$

$$L_k^{-1} = \left[\begin{array}{c|cccc} I_{k-1} & & & & \\ \hline & 1 & & & \\ & & \frac{1}{x_{k+1}-x_k} & & \\ & & & \ddots & \\ & & & & \frac{1}{x_n - x_k} \end{array}\right] \left[\begin{array}{c|cccc} I_{k-1} & & & & \\ \hline & 1 & & & \\ & -1 & 1 & & \\ & \vdots & & \ddots & \\ & -1 & & & 1 \end{array}\right]$$

Fast Björck-Pereyra algorithm

- The solution a of the linear system

$$V(x)a = f$$

is computed by Björck-Pereyra as

$$a = V(x)^{-1} f = U_1^{-1} \ldots U_{n-1}^{-1} L_{n-1}^{-1} \ldots L_1^{-1} f$$

- Each matrix in the factorization is **sparse**, and so each matrix-vector product can be computed in $\mathcal{O}(n)$ operations.
- **Björck-Pereyra** requires only $\mathcal{O}(n^2)$ arithmetic operations vs $\mathcal{O}(n^3)$ of Gaussian elimination.

Fast Traub algorithm

- The **Traub algorithm** (1966) is based on the formula

$$V(x)^{-1} = \tilde{I} \cdot \begin{bmatrix} r_0(x_1) & r_1(x_1) & r_2(x_1) & \cdots & r_{n-1}(x_1) \\ r_0(x_2) & r_1(x_2) & r_2(x_2) & \cdots & r_{n-1}(x_2) \\ r_0(x_3) & r_1(x_3) & r_2(x_3) & \cdots & r_{n-1}(x_3) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ r_0(x_n) & r_1(x_n) & r_2(x_n) & \cdots & r_{n-1}(x_n) \end{bmatrix}^T \cdot \mathrm{diag}(c_1, c_2, \ldots, c_n)$$

where $\tilde{I}$ is the antidiagonal matrix, $c_k = \prod_{\substack{j=1 \\ j \neq k}}^{n} (x_j - x_k)^{-1}$

- The polynomials $\{r_0(x), \ldots, r_{n-1}(x)\}$ are the **Horner (associated) polynomials**, and satisfy **two-term recurrence relations**

$$r_0(x) = \beta_n, \qquad r_k(x) = x r_{k-1}(x) + \beta_{n-k}$$

- **Fast:** requires only $\mathcal{O}(n^2)$ arithmetic operations vs $\mathcal{O}(n^3)$ of Gaussian elimination.

Conditioning of Vandermonde matrices

- The condition numbers of Vandermonde matrices **grow exponentially with their size** (Tyrtyshnikov (1994).
- Björck-Pereyra (1970) : "*... some problems, connected with Vandermonde systems, which traditionally have been considered to be too ill-conditioned to be attacked, actually can be solved with good precision*".

Some Numerical Experiments

- **Björck-Pereyra algorithm.** (**Higham's example**)
  1. Nodes chosen randomly in $(0, 1)$.
  2. RHS alternating signs, $\begin{bmatrix} 1 & -1 & 1 & \cdots \end{bmatrix}^T$
  3. Forward error measured by

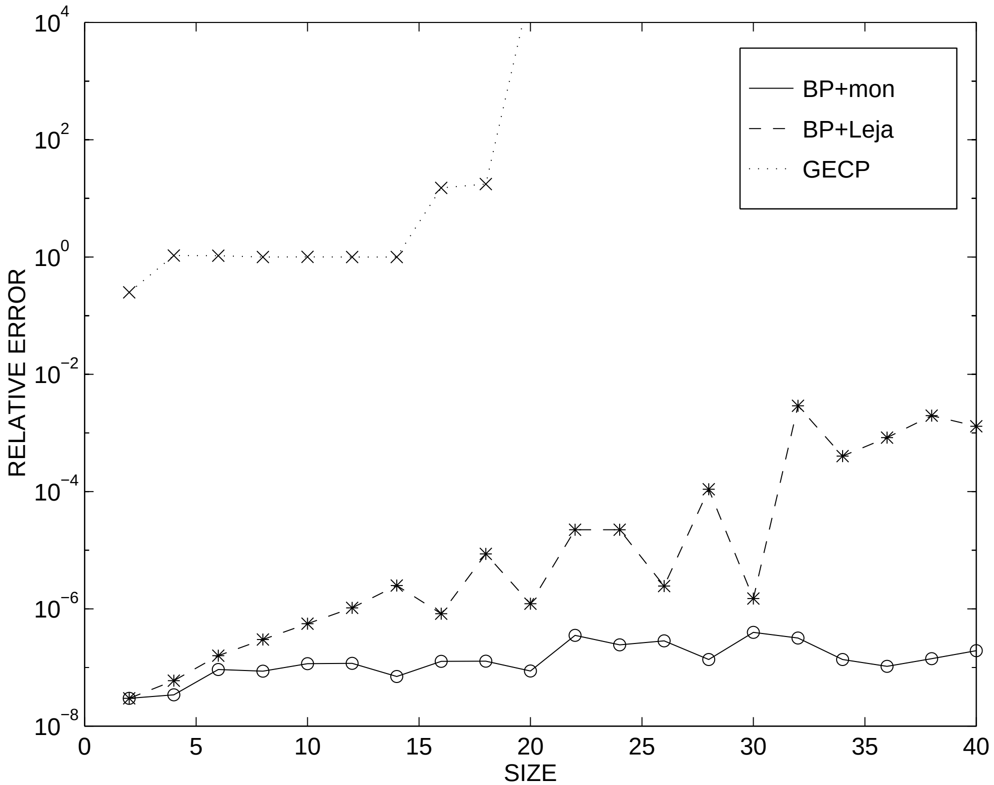
$$e = \frac{\|x - \hat{x}\|_2}{\|x\|_2}$$

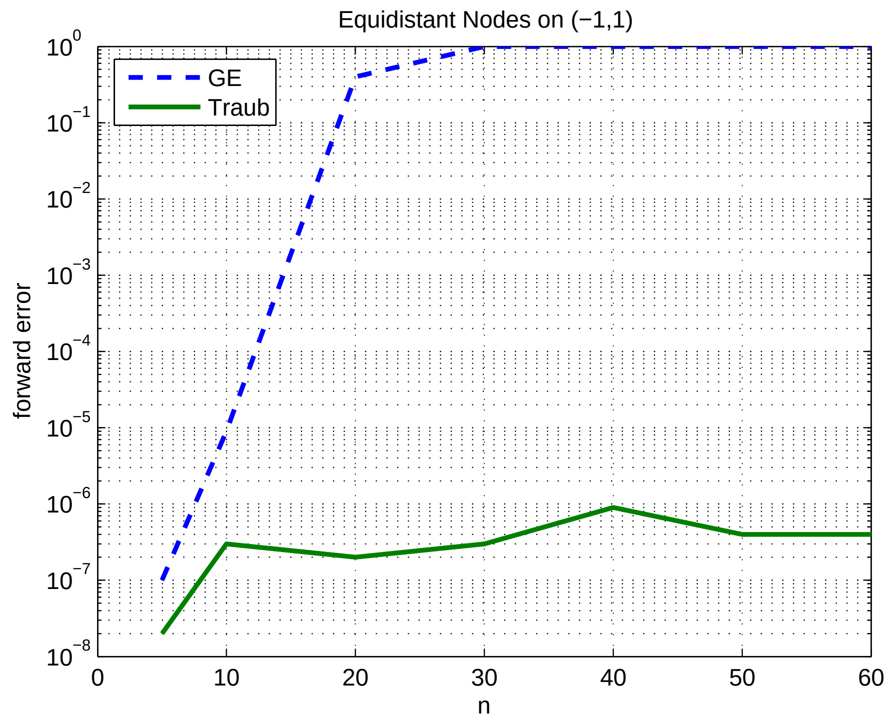
- **Traub algorithm.**
  1. Nodes chosen randomly in $(0, 1)$.
  2. Forward error measured by

$$e = \frac{\|A^{-1} - \widehat{A^{-1}}\|_2}{\|A^{-1}\|_2}$$

Numerical Experiments

Classical Björck-Pereyra & Traub Algorithms





Error Analysis

Björck-Pereyra Algorithm

- **Higham** (1990) showed that under some conditions (sign-oscillating right-hand-side and monotonic ordering of the nodes, all of which are positive), there is a **provably excellent forward error bound:**

$$\frac{|a-\widehat{a}|}{|a|} \leq 5nu + \mathcal{O}(u^2)$$

- **Conclusion:** Very good forward error is to be expected under some conditions.

Polynomial-Vandermonde matrices

- **Definition.** For sets of polynomials and nodes, define a **polynomial-Vandermonde matrix**:

$$x = \{x_1, x_2, \ldots, x_n\}$$
$$R = \{r_0(x), r_1(x), \ldots, r_{n-1}(x)\}$$
$$\Downarrow$$
$$V_R(x) = \begin{bmatrix} r_0(x_1) & r_1(x_1) & r_2(x_1) & \cdots & r_{n-1}(x_1) \\ r_0(x_2) & r_1(x_2) & r_2(x_2) & \cdots & r_{n-1}(x_2) \\ r_0(x_3) & r_1(x_3) & r_2(x_3) & \cdots & r_{n-1}(x_3) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ r_0(x_n) & r_1(x_n) & r_2(x_n) & \cdots & r_{n-1}(x_n) \end{bmatrix}$$

- **Note.** If the polynomial system R satisfies nice recurrence relations, then the n^2 entries of $V_R(x)$ can be defined by only $\mathcal{O}(n)$ parameters.

Fast algorithms for polynomial-Vandermonde matrices

Previous work

polynomial-Vandermonde matrix	Björck-Pereyra-type	Traub-type
Vandermonde matrices	Björck-Pereyra (1970)	Traub (1966)
Vandermonde-like matrices	Kailath-Olshevsky (1996)	Gohberg-Olshevsky (1996)
block Vandermonde matrices	Tang-Golub (1981)	
Chebyshev-Vandermonde matrices	Reichel-Opfer (1991)	Gohberg-Olshevsky (1994)
three-term Vandermonde matrices	Higham (1988,90)	Calvetti-Reichel (1993)
Szegö-Vandermonde matrices	BEGKO (2006)	Olshevsky (2001)

A Björck-Pereyra algorithm for Szegö-Vandermonde matrices

Joint work with Y.Eidelman, I.Gohberg, I.Koltracht, V.Olshevsky

Szegö-Vandermonde matrices

- If the polynomials are Szëgo polynomials, the resulting matrix is a Szëgo-Vandermonde matrix.

$$V_\Phi = \begin{bmatrix} \phi_0(x_1) & \phi_1(x_1) & \phi_2(x_1) & \cdots & \phi_{n-1}(x_1) \\ \phi_0(x_2) & \phi_1(x_2) & \phi_2(x_2) & \cdots & \phi_{n-1}(x_2) \\ \phi_0(x_3) & \phi_1(x_3) & \phi_2(x_3) & \cdots & \phi_{n-1}(x_3) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \phi_0(x_n) & \phi_1(x_n) & \phi_2(x_n) & \cdots & \phi_{n-1}(x_n) \end{bmatrix}$$

Björck-Pereyra-like Formula

- based on the formula

$$V_\Phi^{-1} = U_1^{-1} \dots U_{n-1}^{-1} \, L_{n-1}^{-1} \dots L_1^{-1} \, ,$$

with

$$U_k^{-1} = \text{diag}\{I_{k-1}, \left[\begin{array}{c|ccc} \frac{1}{\alpha_0} & & & \\ 0 & & H - x_k I & \\ \vdots & & & \\ 0 & & & \\ \hline 0 & 0 & \cdots \; 0 & \frac{1}{\alpha_{n-k}} \end{array}\right]\}$$

$$L_k^{-1} = \left[\begin{array}{c|cccc} I_{k-1} & & & & \\ \hline & 1 & & & \\ & & \frac{1}{x_{k+1}-x_k} & & \\ & & & \ddots & \\ & & & & \frac{1}{x_n - x_k} \end{array}\right] \left[\begin{array}{c|cccc} I_{k-1} & & & & \\ \hline & 1 & & & \\ & -1 & 1 & & \\ & \vdots & & \ddots & \\ & -1 & & & 1 \end{array}\right]$$

Unitary Hessenberg Matrices

$$H = \begin{bmatrix} -\rho_0^*\rho_1 & -\rho_0^*\mu_1\rho_2 & -\rho_0^*\mu_1\mu_2\rho_3 & -\rho_0^*\mu_1\mu_2\mu_3\rho_4 & -\rho_0^*\mu_1\mu_2\mu_3\mu_4\rho_5 \\ \mu_1 & -\rho_1^*\rho_2 & -\rho_1^*\mu_2\rho_3 & -\rho_1^*\mu_2\mu_3\rho_4 & -\rho_1^*\mu_2\mu_3\mu_4\rho_5 \\ 0 & \mu_2 & -\rho_2^*\rho_3 & -\rho_2^*\mu_3\rho_4 & -\rho_2^*\mu_3\mu_4\rho_5 \\ 0 & 0 & \mu_3 & -\rho_3^*\rho_4 & -\rho_3^*\mu_4\rho_5 \\ 0 & 0 & 0 & \mu_4 & -\rho_4^*\rho_5 \end{bmatrix}$$

- For monomials (classical BP), the algorithm is **fast** because the H's are **bidiagonal**.
- For the Szegö case, the algorithm is **fast** using a **factorization** of H into **Givens rotations**.

Decomposition of H into Givens (plane) Rotations

Schur parameterization

$$H = H(\rho_1) \times H(\rho_2) \times \cdots \times H(\rho_{n-1}) \times \tilde{H}(\rho_n)$$

$$H = \begin{bmatrix} \rho_1 & \mu_1 & & & \\ \mu_1 & -\rho_1^* & & & \\ & & 1 & & \\ & & & \ddots & \\ & & & & 1 \end{bmatrix} \begin{bmatrix} 1 & & & & \\ & \rho_2 & \mu_2 & & \\ & \mu_2 & -\rho_2^* & & \\ & & & \ddots & \\ & & & & 1 \end{bmatrix} \cdots \begin{bmatrix} 1 & & & & \\ & 1 & & & \\ & & \ddots & & \\ & & & 1 & \\ & & & & \rho_n \end{bmatrix}$$

- Multiplication by H can be implemented in $\mathcal{O}(n)$ operations.
- The cost of the Björck-Pereyra-like algorithm is $\mathcal{O}(n^2)$.

Numerical Illustrations

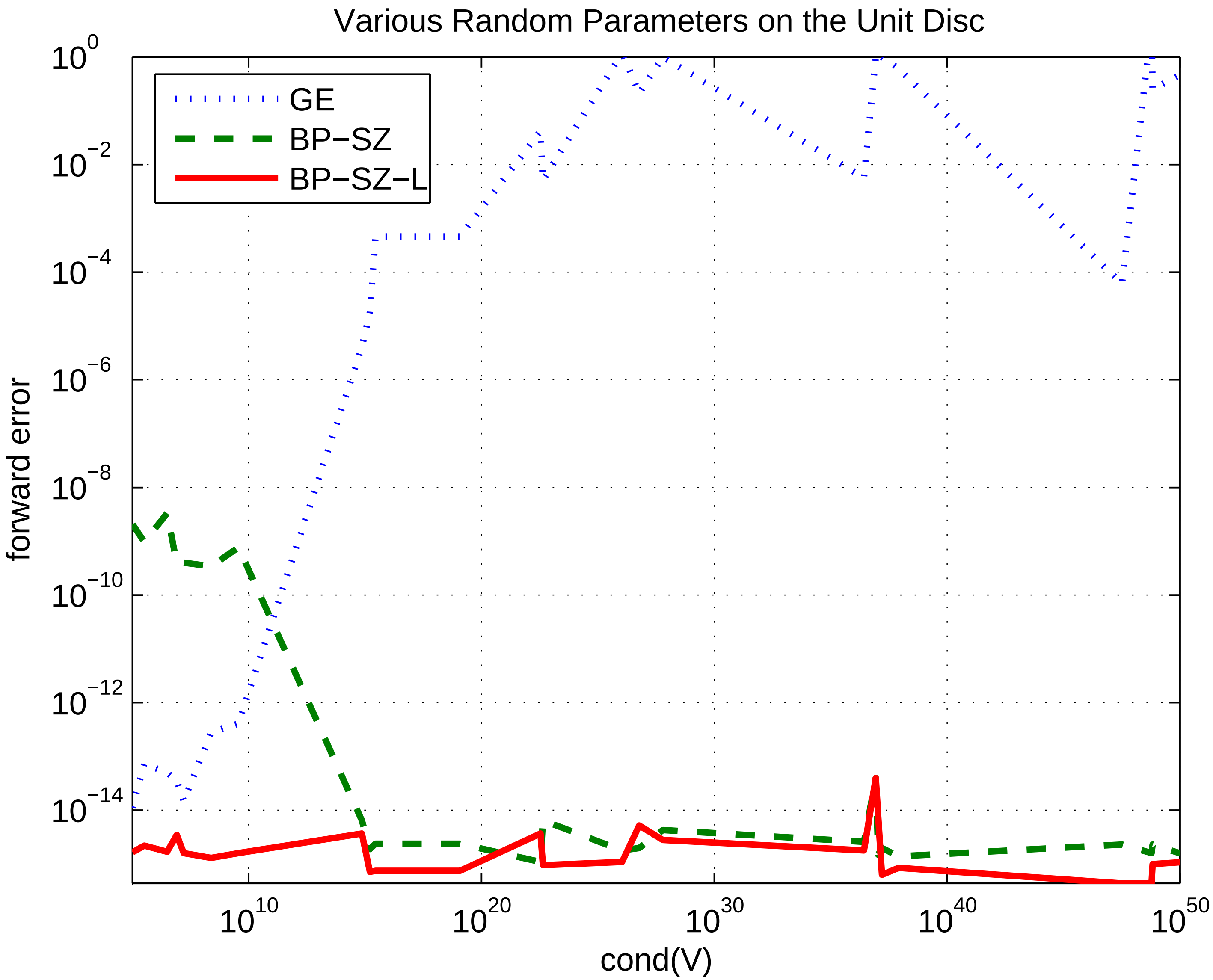
- We compare the **forward error** of the solutions $\hat{x}$ from MATLAB in double precision via

$$e = \frac{\|x - \hat{x}\|_2}{\|x\|_2},$$

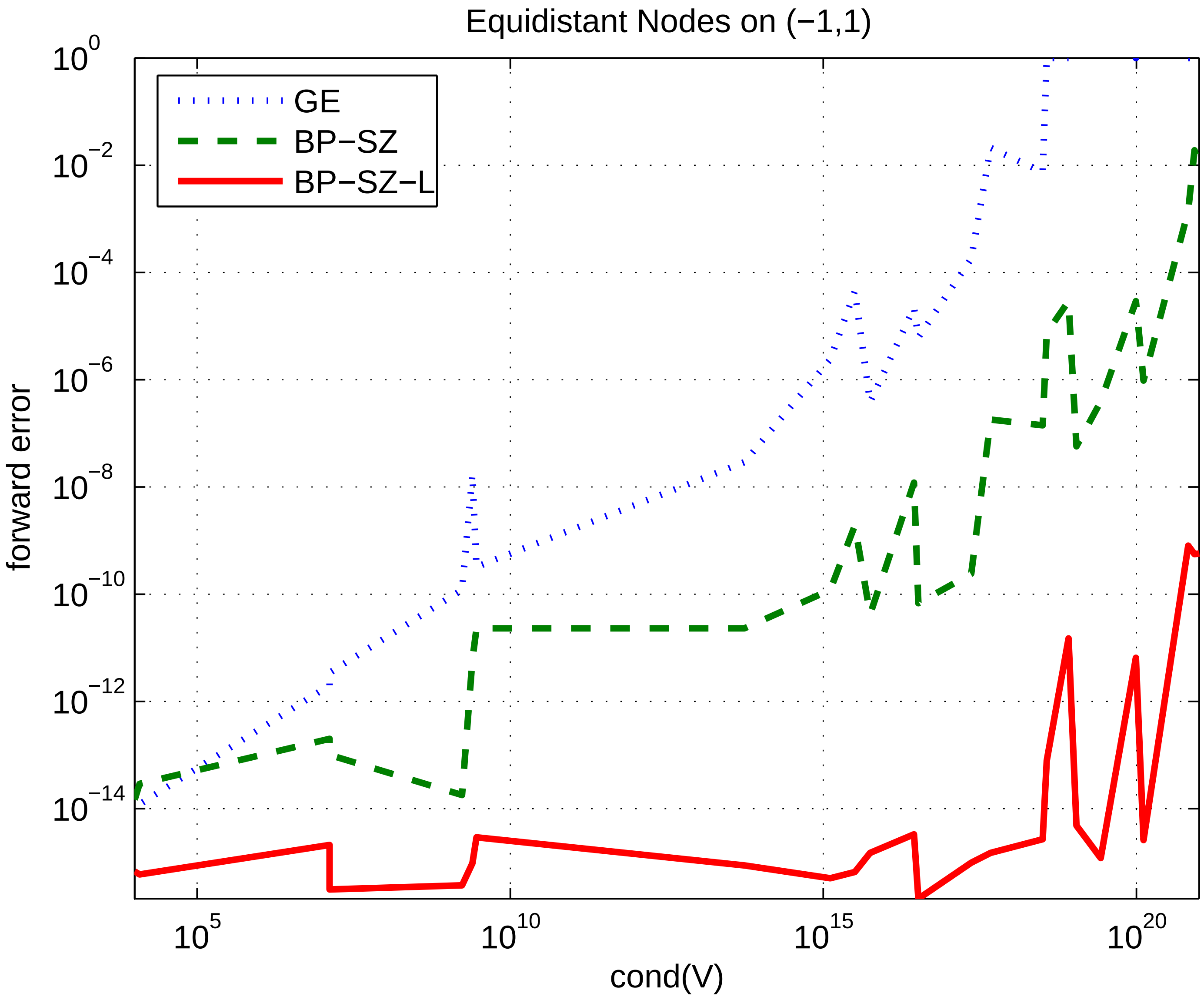
with x, the “exact” solution using MATLAB’s `vpa()` command for software-implemented arbitrary digit arithmetic.

- **GE** - Gaussian elimination via MATLAB’s backslash command.
- **BP-SZ** - Björck-Pereyra-like algorithm.
- **BP-SZ-L** - Björck-Pereyra-like algorithm with nodes ordered via the **Leja ordering**. (Reichel, Higham)
- Use of the **Leja ordering** exploits the Vandermonde structure to produce a partial pivoting ordering of the rows of V. This is at a cost of $\mathcal{O}(n^2)$.

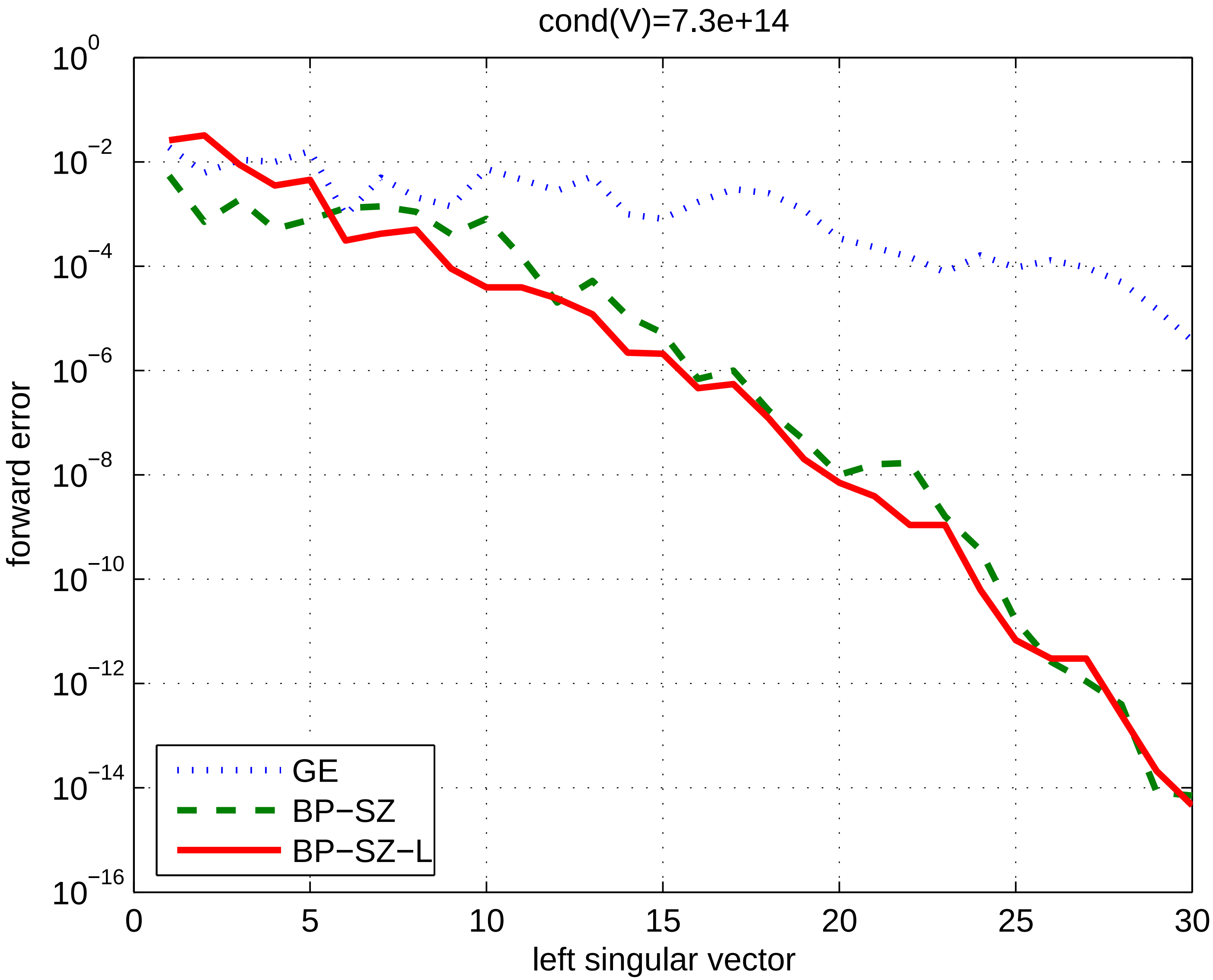
Numerical Illustrations - Experiment 1



Numerical Illustrations - Experiment 2



Numerical Illustrations - Experiment 3



Fast algorithms for quasiseparable-Vandermonde matrices

Joint work with Y.Eidelman, I.Gohberg, I.Koltracht, V.Olshevsky

Fast algorithms for polynomial-Vandermonde matrices

Previous work

polynomial-Vandermonde matrix	Björck-Pereyra-type	Traub-type
Vandermonde matrices	Björck-Pereyra(1970)	Traub (1966)
Vandermonde-like matrices	Kailath-Olshevsky (1996)	Gohberg-Olshevsky (1996)
block Vandermonde matrices	Tang-Golub (1981)	
Chebyshev-Vandermonde matrices	Reichel-Opfer (1991)	Gohberg-Olshevsky (1994)
three-term Vandermonde matrices	Higham (1988,90)	Calvetti-Reichel (1993)
Szegö-Vandermonde matrices	BEGKO (2006)	Olshevsky (2001)
Quasiseparable-Vandermonde	????????????????	????????????????

Quasiseparable Matrices

- **Definition.** A matrix C is **quasiseparable of order one** if

$$\max \text{Rank} C_{12} = \max \text{Rank} C_{21} = 1$$

where the maxima are taken over all symmetric partitions of the form

$$C = \left[\begin{array}{c|c} * & C_{12} \\ \hline C_{21} & * \end{array}\right]$$

- **Previous Work.** Gohberg-Kaashoek-Lerer, Dewilde, Gohberg-Eidelman, Van Barel et al, Tyrtyshnikov et al, Bini et al, Gu-Chandrasekaran et al.

- Henceforth we consider only **Hessenberg** matrices C.

Important Special Cases

$$C = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

- The system of polynomials $r_k(x) = \det(xI - C_{k\times k})$ associated with C is the **monomials** with recurrence relations

$$r_k(x) = x r_{k-1}(x)$$

- The matrix C is **quasiseparable**.

Important Special Cases

$$C = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

Important Special Cases

$$C = \left[\begin{array}{cc|ccc} 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ \hline 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{array}\right]$$

Important Special Cases

$$C = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

Important Special Cases

$$C = \left[\begin{array}{cccc|c} 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ \hline 0 & 0 & 0 & 1 & 0 \end{array}\right]$$

Important Special Cases

Tridiagonal

$$C = \begin{bmatrix} d_1 & g_1 & 0 & 0 & 0 \\ q_1 & d_2 & g_2 & 0 & 0 \\ 0 & q_2 & d_3 & g_3 & 0 \\ 0 & 0 & q_3 & d_4 & g_4 \\ 0 & 0 & 0 & q_4 & d_5 \end{bmatrix}$$

- The system of polynomials $r_k(x) = \det(xI - C_{k\times k})$ associated with C are **real orthogonal polynomials** with recurrence relations

$$r_k(x) = \frac{1}{q_k}(x - d_k)r_{k-1}(x) - \frac{g_{k-1}}{q_k}r_{k-2}(x)$$

- The matrix C is **quasiseparable**.

Important Special Cases

Tridiagonal

$$C = \begin{bmatrix} d_1 & g_1 & 0 & 0 & 0 \\ q_1 & d_2 & g_2 & 0 & 0 \\ 0 & q_2 & d_3 & g_3 & 0 \\ 0 & 0 & q_3 & d_4 & g_4 \\ 0 & 0 & 0 & q_4 & d_5 \end{bmatrix}$$

Important Special Cases

Tridiagonal

$$C = \begin{bmatrix} d_1 & g_1 & 0 & 0 & 0 \\ q_1 & d_2 & g_2 & 0 & 0 \\ 0 & q_2 & d_3 & g_3 & 0 \\ 0 & 0 & q_3 & d_4 & g_4 \\ 0 & 0 & 0 & q_4 & d_5 \end{bmatrix}$$

Important Special Cases

Tridiagonal

$$C = \begin{bmatrix} d_1 & g_1 & 0 & 0 & 0 \\ q_1 & d_2 & g_2 & 0 & 0 \\ 0 & q_2 & d_3 & g_3 & 0 \\ 0 & 0 & q_3 & d_4 & g_4 \\ 0 & 0 & 0 & q_4 & d_5 \end{bmatrix}$$

Important Special Cases

Tridiagonal

$$C = \begin{bmatrix} d_1 & g_1 & 0 & 0 & 0 \\ q_1 & d_2 & g_2 & 0 & 0 \\ 0 & q_2 & d_3 & g_3 & 0 \\ 0 & 0 & q_3 & d_4 & g_4 \\ 0 & 0 & 0 & q_4 & d_5 \end{bmatrix}$$

Important Special Cases

Unitary Hessenberg

$$C = \begin{bmatrix} -\rho_0^*\rho_1 & -\rho_0^*\mu_1\rho_2 & -\rho_0^*\mu_1\mu_2\rho_3 & -\rho_0^*\mu_1\mu_2\mu_3\rho_4 & -\rho_0^*\mu_1\mu_2\mu_3\mu_4\rho_5 \\ \mu_1 & -\rho_1^*\rho_2 & -\rho_1^*\mu_2\rho_3 & -\rho_1^*\mu_2\mu_3\rho_4 & -\rho_1^*\mu_2\mu_3\mu_4\rho_5 \\ 0 & \mu_2 & -\rho_2^*\rho_3 & -\rho_2^*\mu_3\rho_4 & -\rho_2^*\mu_3\mu_4\rho_5 \\ 0 & 0 & \mu_3 & -\rho_3^*\rho_4 & -\rho_3^*\mu_4\rho_5 \\ 0 & 0 & 0 & \mu_4 & -\rho_4^*\rho_5 \end{bmatrix}$$

- The system of polynomials $r_k(x) = \det(xI - C_{k\times k})$ associated with C are the **Szegö polynomials** with recurrence relations

$$\begin{bmatrix} G_k(x) \\ r_k(x) \end{bmatrix} = \frac{1}{\mu_k} \begin{bmatrix} 1 & -\rho_k^* \\ -\rho_k & 1 \end{bmatrix} \begin{bmatrix} G_{k-1}(x) \\ x r_{k-1}(x) \end{bmatrix}$$

- The matrix C is **quasiseparable**.

Important Special Cases

Unitary Hessenberg

$$C = \begin{bmatrix} -\rho_0^*\rho_1 & -\rho_0^*\mu_1\rho_2 & -\rho_0^*\mu_1\mu_2\rho_3 & -\rho_0^*\mu_1\mu_2\mu_3\rho_4 & -\rho_0^*\mu_1\mu_2\mu_3\mu_4\rho_5 \\ \mu_1 & -\rho_1^*\rho_2 & -\rho_1^*\mu_2\rho_3 & -\rho_1^*\mu_2\mu_3\rho_4 & -\rho_1^*\mu_2\mu_3\mu_4\rho_5 \\ 0 & \mu_2 & -\rho_2^*\rho_3 & -\rho_2^*\mu_3\rho_4 & -\rho_2^*\mu_3\mu_4\rho_5 \\ 0 & 0 & \mu_3 & -\rho_3^*\rho_4 & -\rho_3^*\mu_4\rho_5 \\ 0 & 0 & 0 & \mu_4 & -\rho_4^*\rho_5 \end{bmatrix}$$

Important Special Cases

Unitary Hessenberg

$$C = \begin{bmatrix} -\rho_0^*\rho_1 & -\rho_0^*\mu_1\rho_2 & -\rho_0^*\mu_1\mu_2\rho_3 & -\rho_0^*\mu_1\mu_2\mu_3\rho_4 & -\rho_0^*\mu_1\mu_2\mu_3\mu_4\rho_5 \\ \mu_1 & -\rho_1^*\rho_2 & -\rho_1^*\mu_2\rho_3 & -\rho_1^*\mu_2\mu_3\rho_4 & -\rho_1^*\mu_2\mu_3\mu_4\rho_5 \\ 0 & \mu_2 & -\rho_2^*\rho_3 & -\rho_2^*\mu_3\rho_4 & -\rho_2^*\mu_3\mu_4\rho_5 \\ 0 & 0 & \mu_3 & -\rho_3^*\rho_4 & -\rho_3^*\mu_4\rho_5 \\ 0 & 0 & 0 & \mu_4 & -\rho_4^*\rho_5 \end{bmatrix}$$

Important Special Cases

Unitary Hessenberg

$$C = \begin{bmatrix} -\rho_0^*\rho_1 & -\rho_0^*\mu_1\rho_2 & -\rho_0^*\mu_1\mu_2\rho_3 & -\rho_0^*\mu_1\mu_2\mu_3\rho_4 & -\rho_0^*\mu_1\mu_2\mu_3\mu_4\rho_5 \\ \mu_1 & -\rho_1^*\rho_2 & -\rho_1^*\mu_2\rho_3 & -\rho_1^*\mu_2\mu_3\rho_4 & -\rho_1^*\mu_2\mu_3\mu_4\rho_5 \\ 0 & \mu_2 & -\rho_2^*\rho_3 & -\rho_2^*\mu_3\rho_4 & -\rho_2^*\mu_3\mu_4\rho_5 \\ 0 & 0 & \mu_3 & -\rho_3^*\rho_4 & -\rho_3^*\mu_4\rho_5 \\ 0 & 0 & 0 & \mu_4 & -\rho_4^*\rho_5 \end{bmatrix}$$

Important Special Cases

Unitary Hessenberg

$$C = \begin{bmatrix} -\rho_0^*\rho_1 & -\rho_0^*\mu_1\rho_2 & -\rho_0^*\mu_1\mu_2\rho_3 & -\rho_0^*\mu_1\mu_2\mu_3\rho_4 & -\rho_0^*\mu_1\mu_2\mu_3\mu_4\rho_5 \\ \mu_1 & -\rho_1^*\rho_2 & -\rho_1^*\mu_2\rho_3 & -\rho_1^*\mu_2\mu_3\rho_4 & -\rho_1^*\mu_2\mu_3\mu_4\rho_5 \\ 0 & \mu_2 & -\rho_2^*\rho_3 & -\rho_2^*\mu_3\rho_4 & -\rho_2^*\mu_3\mu_4\rho_5 \\ 0 & 0 & \mu_3 & -\rho_3^*\rho_4 & -\rho_3^*\mu_4\rho_5 \\ 0 & 0 & 0 & \mu_4 & -\rho_4^*\rho_5 \end{bmatrix}$$

Quasiseparable-Vandermonde matrices

- **Definition.** A **Quasiseparable-Vandermonde matrix** is of the form

$$V_R = \begin{bmatrix} r_0(x_1) & r_1(x_1) & r_2(x_1) & \cdots & r_{n-1}(x_1) \\ r_0(x_2) & r_1(x_2) & r_2(x_2) & \cdots & r_{n-1}(x_2) \\ r_0(x_3) & r_1(x_3) & r_2(x_3) & \cdots & r_{n-1}(x_3) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ r_0(x_n) & r_1(x_n) & r_2(x_n) & \cdots & r_{n-1}(x_n) \end{bmatrix}$$

where the polynomials $r_k(x) = \det(xI - C_{k\times k})$ correspond to an **order-one quasi-separable Hessenberg** matrix C.

Björck-Pereyra-like algorithm for quasiseparable-Vandermonde matrices

- based on the formula

$$V_R^{-1} = U_1^{-1} \ldots U_{n-1}^{-1} \, L_{n-1}^{-1} \ldots L_1^{-1} \, ,$$

with

$$U_k^{-1} = \text{diag}\{I_{k-1}, \left[\begin{array}{c|c} \begin{matrix} \frac{1}{\alpha_0} \\ 0 \\ \vdots \\ 0 \end{matrix} & C - x_k I \\ \hline 0 & \begin{matrix} 0 & \cdots & 0 & \frac{1}{\alpha_{n-k}} \end{matrix} \end{array}\right]\}$$

$$L_k^{-1} = \left[\begin{array}{c|c} I_{k-1} & \\ \hline & \begin{matrix} 1 & & & \\ & \frac{1}{x_{k+1}-x_k} & & \\ & & \ddots & \\ & & & \frac{1}{x_n - x_k} \end{matrix} \end{array}\right] \left[\begin{array}{c|c} I_{k-1} & \\ \hline & \begin{matrix} 1 & & & \\ -1 & 1 & & \\ \vdots & & \ddots & \\ -1 & & & 1 \end{matrix} \end{array}\right]$$

Complexity of the Björck-Pereyra-like algorithm

- To design a fast algorithm, we need **fast multiplication of C by a vector**.

  In the **monomial** case, C is **bidiagonal**.

  In the **Szegö** case, C is **unitary Hessenberg**, and hence admits a convenient **factorization**.

- In the **quasiseparable** case, C can be multiplied by a vector in $\mathcal{O}(n)$ operations using an algorithm of **Eidelman and Gohberg** (1999).

- Thus the cost of the algorithm is $\mathcal{O}(n^2)$ arithmetic operations.

Special cases of the Björck-Pereyra-like algorithm

- **Tridiagonal** case: This algorithm reduces to the **Higham algorithm** (1990).
  - **Stage I** corresponds to computing $L_{n-1}^{-1} \ldots L_1^{-1} f = x$.
  - **Stage II** corresponds to computing $U_1^{-1} \ldots U_{n-1}^{-1} x$.
- **Unitary Hessenberg** case: This reduces to the **BEGKO algorithm** (2006).

Numerical Illustrations - Björck-Pereyra

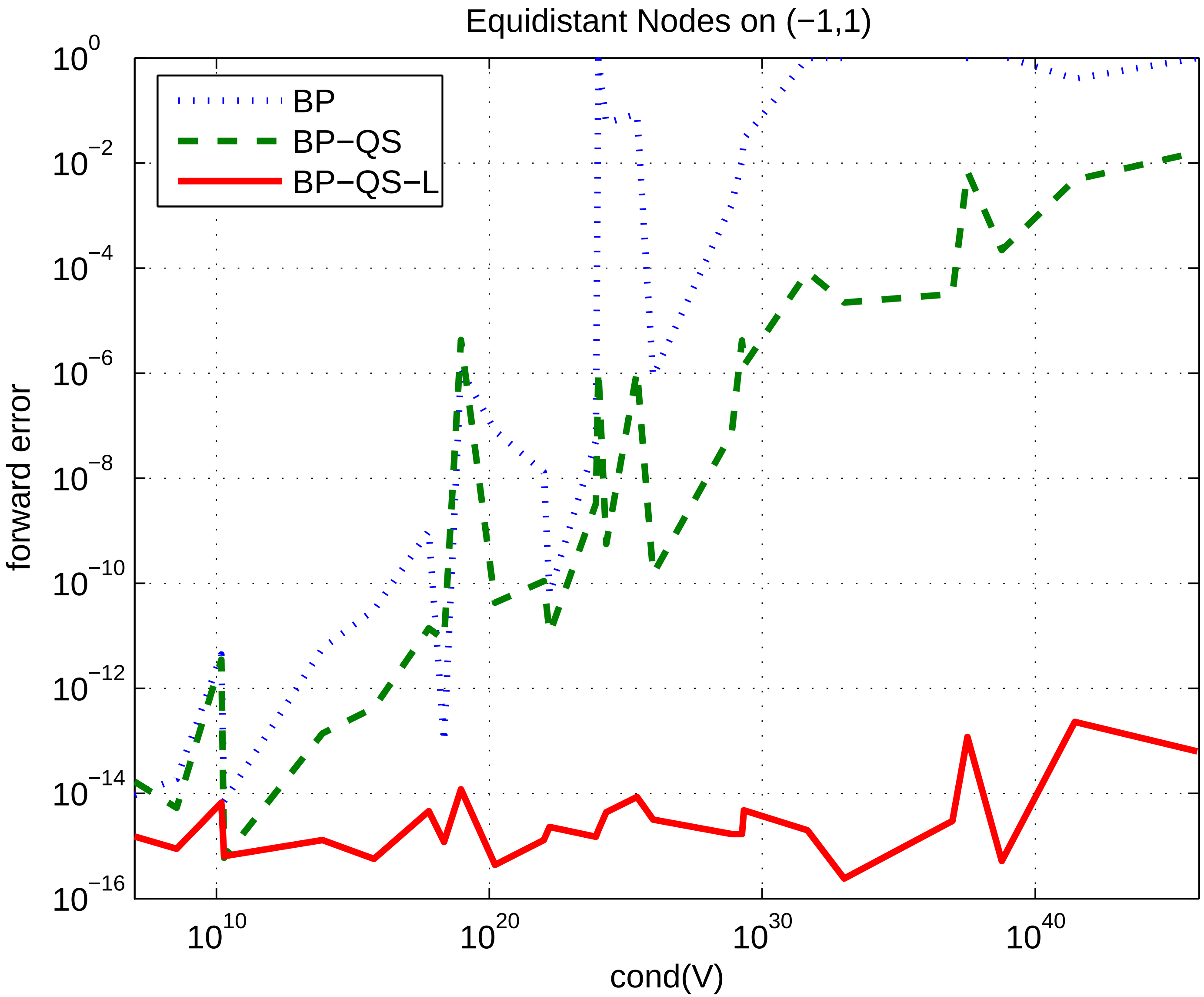
- We compare the **forward error** of the solutions $\hat{x}$ from MATLAB in double precision via

$$e = \frac{\|x - \hat{x}\|_2}{\|x\|_2},$$

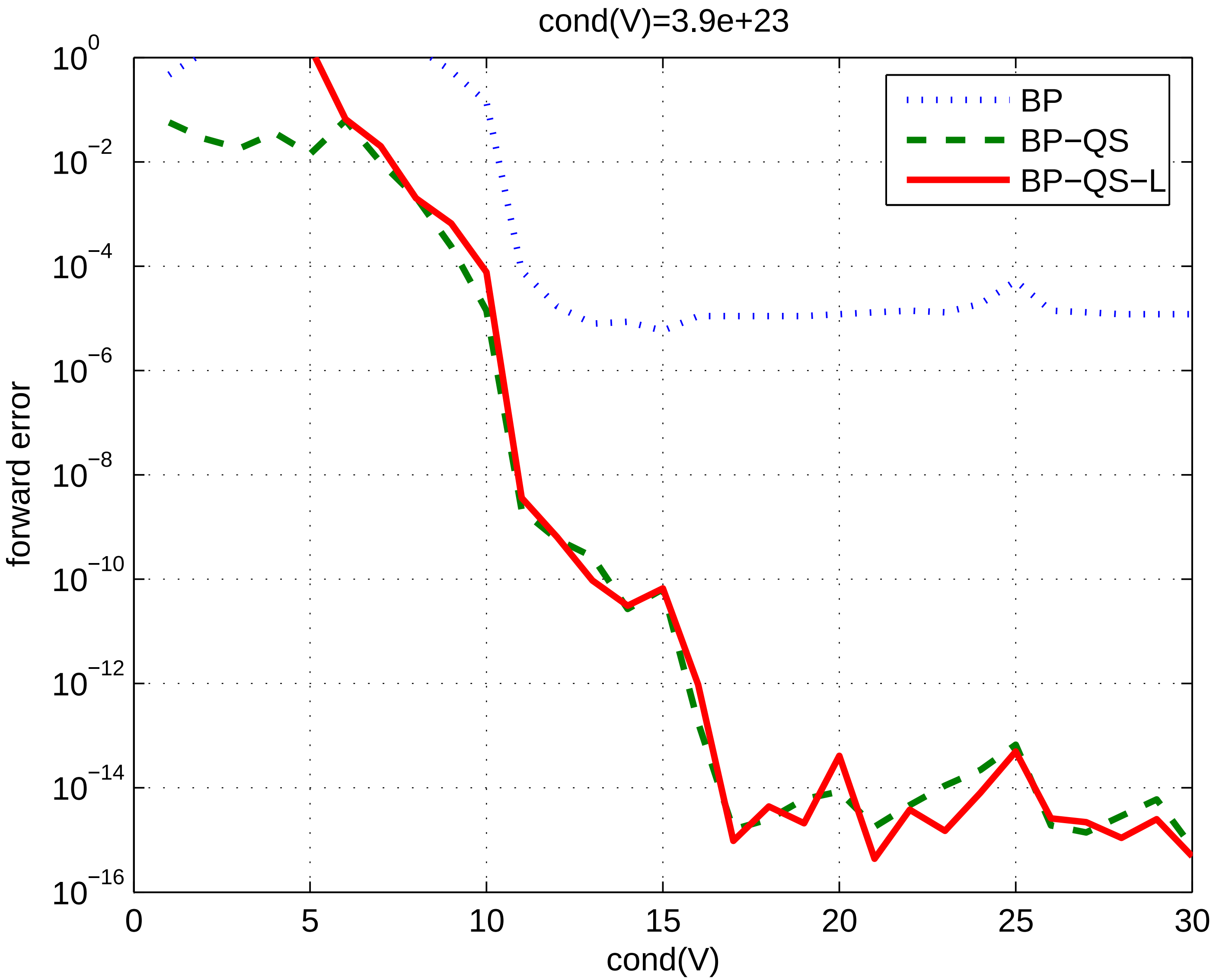
with x, the "exact" solution using MATLAB's `vpa()` command for software-implemented arbitrary digit arithmetic.

- **GE** - Gaussian elimination via MATLAB's backslash command.
- **BP-QS** - Björck-Pereyra-like algorithm.
- **BP-QS-L** - Björck-Pereyra-like algorithm with nodes ordered via the **Leja ordering**. (Reichel, Higham)

Numerical Illustrations - BP Experiment 1



Numerical Illustrations - BP Experiment 2



Fast algorithms for polynomial-Vandermonde matrices

Previous work

polynomial-Vandermonde matrix	Björck-Pereyra-type	Traub-type
Vandermonde matrices	Björck-Pereyra(1970)	Traub (1966)
Vandermonde-like matrices	Kailath-Olshevsky (1996)	Gohberg-Olshevsky (1996)
block Vandermonde matrices	Tang-Golub (1981)	
Chebyshev-Vandermonde matrices	Reichel-Opfer (1991)	Gohberg-Olshevsky (1994)
three-term Vandermonde matrices	Higham (1988,90)	Calvetti-Reichel (1993)
Szegö-Vandermonde matrices	BEGKO (2006)	Olshevsky (2001)
Quasiseparable-Vandermonde	BEGKO (2006)	????????????????

A generator representation

- The following matrix is **Hessenberg** and **order-one quasiseparable**:

$$\begin{bmatrix} d_1 & g_1h_2 & g_1b_2h_3 & g_1b_2b_3h_4 & g_1b_2b_3b_4h_5 \\ p_2q_1 & d_2 & g_2h_3 & g_2b_3h_4 & g_2b_3b_4h_5 \\ 0 & p_3q_2 & d_3 & g_3h_4 & g_3b_4h_5 \\ 0 & 0 & p_4q_3 & d_4 & g_4h_5 \\ 0 & 0 & 0 & p_5q_4 & d_5 \end{bmatrix}$$

A generator representation

➠ The following matrix is **Hessenberg** and **order-one quasiseparable**:

$$\begin{bmatrix} d_1 & g_1h_2 & g_1b_2h_3 & g_1b_2b_3h_4 & g_1b_2b_3b_4h_5 \\ p_2q_1 & d_2 & g_2h_3 & g_2b_3h_4 & g_2b_3b_4h_5 \\ 0 & p_3q_2 & d_3 & g_3h_4 & g_3b_4h_5 \\ 0 & 0 & p_4q_3 & d_4 & g_4h_5 \\ 0 & 0 & 0 & p_5q_4 & d_5 \end{bmatrix}$$

A generator representation

➠ The following matrix is **Hessenberg** and **order-one quasiseparable**:

$$\begin{bmatrix} d_1 & g_1h_2 & g_1b_2h_3 & g_1b_2b_3h_4 & g_1b_2b_3b_4h_5 \\ p_2q_1 & d_2 & g_2h_3 & g_2b_3h_4 & g_2b_3b_4h_5 \\ 0 & p_3q_2 & d_3 & g_3h_4 & g_3b_4h_5 \\ 0 & 0 & p_4q_3 & d_4 & g_4h_5 \\ 0 & 0 & 0 & p_5q_4 & d_5 \end{bmatrix}$$

A generator representation

➠ The following matrix is **Hessenberg** and **order-one quasiseparable**:

$$\left[\begin{array}{ccc|cc} d_1 & g_1h_2 & g_1b_2h_3 & g_1b_2b_3h_4 & g_1b_2b_3b_4h_5 \\ p_2q_1 & d_2 & g_2h_3 & g_2b_3h_4 & g_2b_3b_4h_5 \\ 0 & p_3q_2 & d_3 & g_3h_4 & g_3b_4h_5 \\ \hline 0 & 0 & p_4q_3 & d_4 & g_4h_5 \\ 0 & 0 & 0 & p_5q_4 & d_5 \end{array}\right]$$

A generator representation

- The following matrix is **Hessenberg** and **order-one quasiseparable**:

$$\begin{bmatrix} d_1 & g_1h_2 & g_1b_2h_3 & g_1b_2b_3h_4 & g_1b_2b_3b_4h_5 \\ p_2q_1 & d_2 & g_2h_3 & g_2b_3h_4 & g_2b_3b_4h_5 \\ 0 & p_3q_2 & d_3 & g_3h_4 & g_3b_4h_5 \\ 0 & 0 & p_4q_3 & d_4 & g_4h_5 \\ 0 & 0 & 0 & p_5q_4 & d_5 \end{bmatrix}$$

A generator representation

- The following matrix is **Hessenberg** and **order-one quasiseparable**:

$$\begin{bmatrix} d_1 & g_1h_2 & g_1b_2h_3 & g_1b_2b_3h_4 & g_1b_2b_3b_4h_5 \\ p_2q_1 & d_2 & g_2h_3 & g_2b_3h_4 & g_2b_3b_4h_5 \\ 0 & p_3q_2 & d_3 & g_3h_4 & g_3b_4h_5 \\ 0 & 0 & p_4q_3 & d_4 & g_4h_5 \\ 0 & 0 & 0 & p_5q_4 & d_5 \end{bmatrix}$$

- This **generator representation** exists for any **order-one quasiseparable** matrix.

Order-one quasiseparable-Hessenberg $\Leftrightarrow$ Generator representation

Recurrence relations satisfied by polynomials related to quasiseparable matrices

- **Theorem.** A matrix C is **order-one quasiseparable-Hessenberg** if and only if the system of polynomials $r_k(x) = \det(xI - C_{k \times k})$ satisfy the recurrence relations

Two-term recurrence relations

$$\begin{bmatrix} G_k(x) \\ r_k(x) \end{bmatrix} = \frac{1}{p_{k+1}q_k} \begin{bmatrix} \alpha_k & -g_{k+1} \\ h_k/b_k & 1 \end{bmatrix} \begin{bmatrix} G_{k-1}(x) \\ u_k(x) r_{k-1}(x) \end{bmatrix}$$

Three-term recurrence relations

$$r_k(x) = -\frac{1}{p_{k+1}q_k}\psi_k(x) r_{k-1}(x) - \frac{1}{p_{k+1}p_k q_{k-1} q_k}\varphi_k(x) r_{k-2}(x)$$

Special cases of these recurrence relations

Matrix	Polynomial system
Shift matrix	monomials
Tridiagonal matrix	real orthogonal polynomials
Unitary Hessenberg matrix	Szegö polynomials

Special cases of these recurrence relations

Matrix	Polynomial system
Shift matrix	monomials
Tridiagonal matrix	real orthogonal polynomials
Unitary Hessenberg matrix	Szegö polynomials

Special cases of these recurrence relations

Monomial case

$$r_k(x) = -\frac{1}{p_{k+1}q_k}\psi_k(x)r_{k-1}(x) - \frac{1}{p_{k+1}p_kq_{k-1}q_k}\varphi_k(x)r_{k-2}(x)$$

$$\Downarrow$$

$$p_k = 1, \ \ q_k = 1, \ \ d_k = 0, \ \ g_k = 0, \ \ b_k = 0, \ \ h_k = 1$$

$$\Downarrow$$

$$r_k(x) = x \cdot r_{k-1}(x)$$

Special cases of these recurrence relations

Matrix	Polynomial system
Shift matrix	monomials
Tridiagonal matrix	real orthogonal polynomials
Unitary Hessenberg matrix	Szegö polynomials

Special cases of these recurrence relations

Real orthogonal polynomial case

$$r_k(x) = -\frac{1}{p_{k+1}q_k}\psi_k(x)r_{k-1}(x) - \frac{1}{p_{k+1}p_k q_{k-1}q_k}\varphi_k(x)r_{k-2}(x)$$

$$\Downarrow$$

$$p_k = 1, \ \ b_k = 0, \ \ h_k = 1$$

$$\Downarrow$$

$$r_k(x) = \frac{1}{q_k}(x - d_k)r_{k-1}(x) - \frac{g_{k-1}}{q_k}r_{k-2}(x)$$

Special cases of these recurrence relations

Matrix	Polynomial system
Shift matrix	monomials
Tridiagonal matrix	real orthogonal polynomials
Unitary Hessenberg matrix	Szegö polynomials

Special cases of these recurrence relations

Szegö polynomial case

$$\begin{bmatrix} G_k(x) \\ r_k(x) \end{bmatrix} = \frac{1}{p_{k+1}q_k} \begin{bmatrix} \alpha_k & -g_{k+1} \\ h_k/b_k & 1 \end{bmatrix} \begin{bmatrix} G_{k-1}(x) \\ u_k(x)r_{k-1}(x) \end{bmatrix}$$

$$\Downarrow$$

$$p_k = 1, \;\; q_k = \mu_k, \;\; d_k = -\rho^*_{k-1}\rho_k, \;\; g_k = \rho^*_{k-1}, \;\; b_k = \mu_{k-1}, \;\; h_k = -\mu_{k-1}\rho_k$$

$$\Downarrow$$

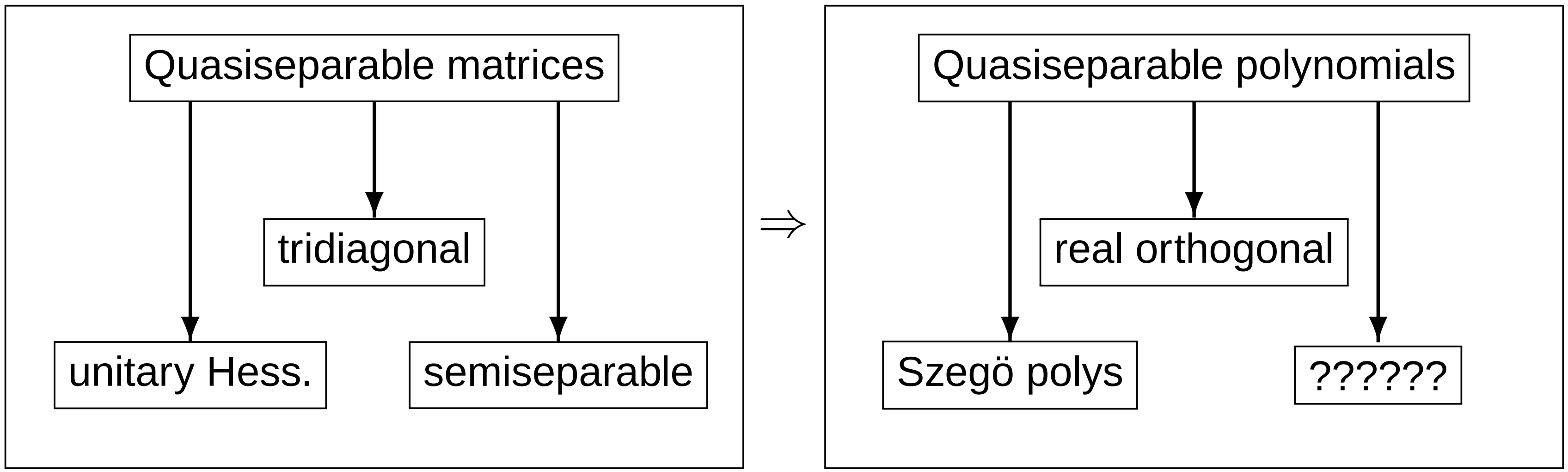
$$\begin{bmatrix} G_k(x) \\ r_k(x) \end{bmatrix} = \frac{1}{\mu_k} \begin{bmatrix} 1 & -\rho^*_k \\ -\rho_k & 1 \end{bmatrix} \begin{bmatrix} G_{k-1}(x) \\ x r_{k-1}(x) \end{bmatrix}$$

A Class of Polynomials

➠ Systems of polynomials associated with a **quasiseparable matrix**.

(**Quasiseparable polynomials**)?

Tridiagonal matrices, unitary Hessenberg matrices, semiseparable matrices are special cases of quasiseparable matrices	$\Rightarrow$	Polynomials orthogonal on (i) a real interval, and (ii) on the unit circle are special cases of quasiseparable polynomials



Traub-like algorithm for quasiseparable-Vandermonde matrices

- **Traub-like algorithm**. Based on the formula

$$V_R{}^{-1} = \begin{bmatrix} \hat{r}_0(x_1) & \hat{r}_1(x_1) & \cdots & \hat{r}_{n-1}(x_1) \\ \hat{r}_0(x_2) & \hat{r}_1(x_2) & \cdots & \hat{r}_{n-1}(x_2) \\ \vdots & \vdots & \ddots & \vdots \\ \hat{r}_0(x_n) & \hat{r}_1(x_n) & \cdots & \hat{r}_{n-1}(x_n) \end{bmatrix}^{-1} = \tilde{I} \cdot V_{\hat{R}}{}^T \cdot \mathrm{diag}(c_1, c_2, \ldots, c_n)$$

where $\tilde{I}$ is the antidiagonal matrix, $c_k = \prod_{\substack{j=1 \\ j \neq k}}^{n} (x_j - x_k)^{-1}$

- $\hat{R}$ is the system of **Horner-like** polynomials corresponding to the polynomial system R. (When P is the monomial basis, this is the classical Traub (1966))
- **How do we evaluate the polynomials $\hat{r}_k$ at the nodes?**

Three-Term Recurrence Relations

Original polynomials forming V_R

$$r_k(x) = -\frac{1}{p_{k+1}q_k}\psi_k(x)r_{k-1}(x) - \frac{1}{p_{k+1}p_kq_{k-1}q_k}\varphi_k(x)r_{k-2}(x)$$

Horner-like polynomials forming V_R^{-1}

$$\hat{r}_k(x) = -\frac{1}{\hat{p}_{k+1}\hat{q}_k}\hat{\psi}_k(x)\hat{r}_{k-1}(x) - \frac{1}{\hat{p}_{k+1}\hat{p}_k\hat{q}_{k-1}\hat{q}_k}\hat{\varphi}_k(x)\hat{r}_{k-2}(x) - \hat{z}_k$$

$$\psi_k(x) = (d_k - x) - \frac{p_kh_kq_{k-1}b_{k-1}}{h_{k-1}} \qquad \varphi_k(x) = \frac{p_kh_kq_{k-1}}{h_{k-1}}((d_k - x)b_k - h_kg_k)$$

$$z_k = \frac{h_kb_{k-1}}{p_{k+1}q_kh_{k-1}}\beta_{n-k+1} - \frac{1}{p_{k+1}q_k}\beta_{n-k}$$

Two-Term Recurrence Relations

Original polynomials forming V_R

$$\begin{bmatrix} G_k(x) \\ r_k(x) \end{bmatrix} = \frac{1}{p_{k+1}q_k} \begin{bmatrix} \alpha_k & -g_{k+1} \\ h_k/b_k & 1 \end{bmatrix} \begin{bmatrix} G_{k-1}(x) \\ u_k(x) r_{k-1}(x) \end{bmatrix}$$

Horner-like polynomials forming V_R^{-1}

$$\begin{bmatrix} \hat{G}_k(x) \\ \hat{r}_k(x) \end{bmatrix} = \frac{1}{\hat{p}_{k+1}\hat{q}_k} \begin{bmatrix} \hat{\alpha}_k & -\hat{g}_{k+1} \\ \hat{h}_k/\hat{b}_k & 1 \end{bmatrix} \begin{bmatrix} \hat{G}_{k-1}(x) \\ \hat{u}_k(x) \hat{r}_{k-1}(x) + \beta_{n-k} \end{bmatrix}$$

$$\alpha_k = p_{k+1} b_{k+1} q_k - \frac{g_{k+1} h_k}{b_k} \qquad u_k(x) = x - d_k + \frac{g_k h_k}{b_k}$$

What are the coefficients β_k?

- The difference between the recurrence relations for the original polynomials and those for the Horner-like polynomials is the presence of the coefficients β_k.
- β_k is the coefficient of $r_k(x)$ in the decomposition of the **master polynomial** $\beta(x) = \prod_{i=1}^{n}(x - x_i)$ into the $\{r_k\}$ basis:

$$\prod_{i=1}^{n}(x - x_i) = \beta_0 r_0(x) + \beta_1 r_1(x) + \cdots + \beta_n r_n(x)$$

- These coefficients can be computed in $\mathcal{O}(n^2)$ arithmetic operations.

Complexity of the Traub-like algorithm

- Each Horner-like polynomial can be evaluated at all of the nodes in $\mathcal{O}(n)$ operations.
- The coefficients β_k can be computed in $\mathcal{O}(n^2)$ operations.
- The total cost of the algorithm is $\mathcal{O}(n^2)$ operations. Comparing this to the complexity of Gaussian elimination, $\mathcal{O}(n^3)$, we have the algorithm is **FAST!**

Special cases of the Traub-like algorithm

- **Tridiagonal** case: This reduces to the algorithm of **Calvetti-Reichel** (1993).
- **Unitary Hessenberg** case: This reduces to the algorithm of **Olshevsky** (2001).

Fast algorithms for polynomial-Vandermonde matrices related to quasiseparable matrices

Tom Bella
Department of Mathematics
University of Connecticut
Storrs, CT 06269 USA

Joint work with Yuli Eidelman, Israel Gohberg, Israel Koltracht, & Vadim Olshevsky.

Supplemental Slides

Confederate Matrices

- **Definition** For polynomials $R = \{r_0(x), r_1(x), \ldots, r_n(x)\}$ satisfying n-term recurrence relations

$$r_k(x) = \alpha_k \cdot x r_{k-1}(x) - a_{k-1,k} \cdot r_{k-1}(x) - a_{k-2,k} \cdot r_{k-2}(x) - \cdots - a_{0,k} \cdot r_0(x).$$

and the polynomial

$$b(x) = b_0 \cdot r_0(x) + b_1 \cdot r_1(x) + \cdots + b_{n-1} \cdot r_{n-1}(x) + b_n \cdot r_n(x)$$

define the **confederate matrix** of b with respect to R by

$$C_R(b) = \begin{bmatrix} \frac{a_{01}}{\alpha_1} & \frac{a_{02}}{\alpha_2} & \frac{a_{03}}{\alpha_3} & \cdots & \cdots & \frac{a_{0,n}}{\alpha_n} - \frac{1}{\alpha_n} \cdot \frac{b_0}{b_n} \\ \frac{1}{\alpha_1} & \frac{a_{12}}{\alpha_2} & \frac{a_{13}}{\alpha_3} & \cdots & \cdots & \frac{a_{1,n}}{\alpha_n} - \frac{1}{\alpha_n} \cdot \frac{b_1}{b_n} \\ 0 & \frac{1}{\alpha_2} & \frac{a_{23}}{\alpha_3} & \cdots & \cdots & \frac{a_{2,n}}{\alpha_n} - \frac{1}{\alpha_n} \cdot \frac{b_2}{b_n} \\ 0 & 0 & \frac{1}{\alpha_3} & \ddots & & \vdots \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & \frac{1}{\alpha_{n-1}} & \frac{a_{n-1,n}}{\alpha_n} - \frac{1}{\alpha_n} \cdot \frac{b_{n-1}}{b_n} \end{bmatrix}$$

Confederate Matrices

$$
\begin{aligned}
r_1(x) &= \alpha_1 \cdot x r_0(x) - a_{0,1} \cdot r_0(x) \\
r_2(x) &= \alpha_2 \cdot x r_1(x) - a_{1,2} \cdot r_1(x) - a_{0,2} \cdot r_0(x) \\
r_3(x) &= \alpha_3 \cdot x r_2(x) - a_{2,3} \cdot r_2(x) - a_{1,3} \cdot r_1(x) - a_{0,3} \cdot r_0(x) \\
&\vdots
\end{aligned}
$$

$$
\begin{bmatrix}
\frac{a_{01}}{\alpha_1} & \frac{a_{02}}{\alpha_2} & \frac{a_{03}}{\alpha_3} & \cdots & \cdots & \frac{a_{0,n}}{\alpha_n} - \frac{1}{\alpha_n} \cdot \frac{b_0}{b_n} \\
\frac{1}{\alpha_1} & \frac{a_{12}}{\alpha_2} & \frac{a_{13}}{\alpha_3} & \cdots & \cdots & \frac{a_{1,n}}{\alpha_n} - \frac{1}{\alpha_n} \cdot \frac{b_1}{b_n} \\
0 & \frac{1}{\alpha_2} & \frac{a_{23}}{\alpha_3} & \cdots & \cdots & \frac{a_{2,n}}{\alpha_n} - \frac{1}{\alpha_n} \cdot \frac{b_2}{b_n} \\
0 & 0 & \frac{1}{\alpha_3} & \ddots & & \vdots \\
\vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\
0 & 0 & \cdots & 0 & \frac{1}{\alpha_{n-1}} & \frac{a_{n-1,n}}{\alpha_n} - \frac{1}{\alpha_n} \cdot \frac{b_{n-1}}{b_n}
\end{bmatrix}
$$

Confederate Matrices

$$
\begin{aligned}
r_1(x) &= \alpha_1 \cdot x r_0(x) - a_{0,1} \cdot r_0(x) \\
r_2(x) &= \alpha_2 \cdot x r_1(x) - a_{1,2} \cdot r_1(x) - a_{0,2} \cdot r_0(x) \\
r_3(x) &= \alpha_3 \cdot x r_2(x) - a_{2,3} \cdot r_2(x) - a_{1,3} \cdot r_1(x) - a_{0,3} \cdot r_0(x) \\
&\vdots
\end{aligned}
$$

$$
\begin{bmatrix}
\frac{a_{01}}{\alpha_1} & \frac{a_{02}}{\alpha_2} & \frac{a_{03}}{\alpha_3} & \cdots & \cdots & \frac{a_{0,n}}{\alpha_n} - \frac{1}{\alpha_n} \cdot \frac{b_0}{b_n} \\
\frac{1}{\alpha_1} & \frac{a_{12}}{\alpha_2} & \frac{a_{13}}{\alpha_3} & \cdots & \cdots & \frac{a_{1,n}}{\alpha_n} - \frac{1}{\alpha_n} \cdot \frac{b_1}{b_n} \\
0 & \frac{1}{\alpha_2} & \frac{a_{23}}{\alpha_3} & \cdots & \cdots & \frac{a_{2,n}}{\alpha_n} - \frac{1}{\alpha_n} \cdot \frac{b_2}{b_n} \\
0 & 0 & \frac{1}{\alpha_3} & \ddots & & \vdots \\
\vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\
0 & 0 & \cdots & 0 & \frac{1}{\alpha_{n-1}} & \frac{a_{n-1,n}}{\alpha_n} - \frac{1}{\alpha_n} \cdot \frac{b_{n-1}}{b_n}
\end{bmatrix}
$$

Confederate Matrices

$$
\begin{aligned}
r_1(x) &= \alpha_1 \cdot x r_0(x) - a_{0,1} \cdot r_0(x) \\
r_2(x) &= \alpha_2 \cdot x r_1(x) - a_{1,2} \cdot r_1(x) - a_{0,2} \cdot r_0(x) \\
r_3(x) &= \alpha_3 \cdot x r_2(x) - a_{2,3} \cdot r_2(x) - a_{1,3} \cdot r_1(x) - a_{0,3} \cdot r_0(x) \\
&\vdots
\end{aligned}
$$

$$
\begin{bmatrix}
\frac{a_{01}}{\alpha_1} & \frac{a_{02}}{\alpha_2} & \frac{a_{03}}{\alpha_3} & \cdots & \cdots & \frac{a_{0,n}}{\alpha_n} - \frac{1}{\alpha_n} \cdot \frac{b_0}{b_n} \\
\frac{1}{\alpha_1} & \frac{a_{12}}{\alpha_2} & \frac{a_{13}}{\alpha_3} & \cdots & \cdots & \frac{a_{1,n}}{\alpha_n} - \frac{1}{\alpha_n} \cdot \frac{b_1}{b_n} \\
0 & \frac{1}{\alpha_2} & \frac{a_{23}}{\alpha_3} & \cdots & \cdots & \frac{a_{2,n}}{\alpha_n} - \frac{1}{\alpha_n} \cdot \frac{b_2}{b_n} \\
0 & 0 & \frac{1}{\alpha_3} & \ddots & & \vdots \\
\vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\
0 & 0 & \cdots & 0 & \frac{1}{\alpha_{n-1}} & \frac{a_{n-1,n}}{\alpha_n} - \frac{1}{\alpha_n} \cdot \frac{b_{n-1}}{b_n}
\end{bmatrix}
$$

Motivation for Horner-like Polynomials

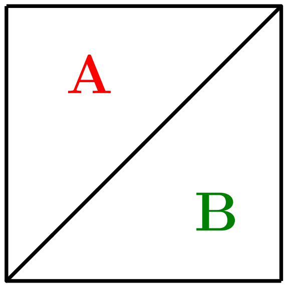
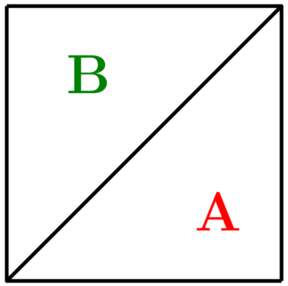
- The confederate matrix $C(b)$ for a polynomial $b(x) = b_0 + b_1 x + \cdots + b_n x_n$ in the **monomial basis** reduces to the companion matrix, and the confederate matrix $C_R(\hat{p}_n)$ for the **Horner polynomials** is:

$$\mathbf{C}(\mathbf{b}) = \begin{bmatrix} 0 & 0 & \cdots & 0 & -b_0 \\ 1 & 0 & \cdots & 0 & -b_1 \\ 0 & 1 & \ddots & \vdots & \vdots \\ \vdots & & \ddots & 0 & \vdots \\ 0 & \cdots & 0 & 1 & -b_{n-1} \end{bmatrix} \qquad \mathbf{C_R}(\mathbf{\hat{p}_n}) = \begin{bmatrix} -b_{n-1} & -b_{n-2} & \cdots & -b_1 & -b_0 \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \ddots & \vdots & 0 \\ \vdots & & \ddots & 0 & \vdots \\ 0 & \cdots & 0 & 1 & 0 \end{bmatrix}$$

- **Observation**: $\mathbf{C_R}(\mathbf{\hat{p}_n}) = \tilde{I} \cdot \mathbf{C}(\mathbf{b})^T \cdot \tilde{I}$

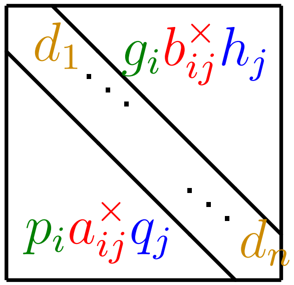
Horner-like Polynomials

- A relation between the matrix corresponding to a system of polynomials R and the matrix corresponding to the **Horner-like** polynomials $\hat{R}$.

		
C	$\Rightarrow$ **pertransposition**	$\tilde{I} \cdot C^T \cdot \tilde{I}$
$\{r_0(x), r_1(x), \ldots, r_{n-1}(x)\}$		$\{\hat{r}_0(x), \hat{r}_1(x), \ldots, \hat{r}_{n-1}(x)\}$
Polynomial System		Horner-like polynomials

Horner-like Polynomials

➠ **Fact.** A pertransposed quasiseparable matrix is again a quasiseparable matrix.

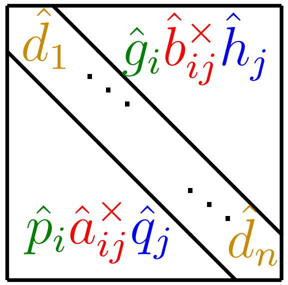


Q/S-Hessenberg Matrix

$$\{r_0(x), r_1(x), \ldots, r_{n-1}(x)\}$$

Polynomial System

$\Rightarrow$ **pertransposition**



Q/S-Hessenberg Matrix

$$\{\hat{r}_0(x), \hat{r}_1(x), \ldots, \hat{r}_{n-1}(x)\}$$

Horner-like polynomials

Generators of a Quasiseparable matrix

- Can be represented in terms of their **generators**:

Diagonal entries

$$d_k \quad k = 1, \ldots, n$$

Lower Generators		Upper Generators	
p_k	$k = 2, \ldots, n$	g_k	$k = 1, \ldots, n-1$
a_k	$k = 2, \ldots, n-1$	b_k	$k = 2, \ldots, n-1$
q_k	$k = 1, \ldots, n-1$	h_k	$k = 2, \ldots, n$

- **Example.** In terms of generators, with $n = 5$,

$$\begin{bmatrix} d_1 & g_1 h_2 & g_1 b_2 h_3 & g_1 b_2 b_3 h_4 & g_1 b_2 b_3 b_4 h_5 \\ p_2 q_1 & d_2 & g_2 h_3 & g_2 b_3 h_4 & g_2 b_3 b_4 h_5 \\ p_3 a_2 q_1 & p_3 q_2 & d_3 & g_3 h_4 & g_3 b_4 h_5 \\ p_4 a_3 a_2 q_1 & p_4 a_3 q_2 & p_4 q_3 & d_4 & g_4 h_5 \\ p_5 a_4 a_3 a_2 q_1 & p_5 a_4 a_3 q_2 & p_5 a_4 q_3 & p_5 q_4 & d_5 \end{bmatrix}$$

Recurrence relations that generalize Szegö recurrence relations

Three-Term Recurrence Relations

$$r_0(x) = 1 \qquad r_1(x) = \frac{1}{p_2 q_1}(x - d_1)$$

$$r_k(x) = -\frac{1}{p_{k+1}q_k}\psi_k(x) r_{k-1}(x) - \frac{1}{p_{k+1}p_k q_{k-1} q_k}\varphi_k(x) r_{k-2}(x)$$

$$\Downarrow$$

$$p_k = 1, \;\; q_k = \mu_k, \;\; d_k = -\rho_{k-1}^* \rho_k, \;\; g_k = \rho_{k-1}^*, \;\; b_k = \mu_{k-1}, \;\; h_k = -\mu_{k-1}\rho_k$$

$$\Downarrow$$

$$r_0(x) = 1 \qquad r_1(x) = \frac{1}{\mu_1}(x + \rho_0^* \rho_1)$$

$$r_k(x) = -\frac{1}{\mu_k}\left(x + \frac{\rho_k}{\rho_{k-1}}\right) r_{k-1}(x) - \left(\frac{\rho_k \mu_{k-1}}{\rho_{k-1}\mu_k}\right) x \cdot r_{k-2}(x)$$

Recurrence relations that generalize Szegö recurrence relations

Perturbed Two-Term Recurrence Relations

$$\begin{bmatrix} G_0(x) \\ r_0(x) \end{bmatrix} = \frac{1}{p_2 q_1} \begin{bmatrix} -g_1 \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} G_k(x) \\ r_k(x) \end{bmatrix} = \frac{1}{p_{k+1} q_k} \begin{bmatrix} \alpha_k & -g_{k+1} \\ h_k/b_k & 1 \end{bmatrix} \begin{bmatrix} G_{k-1}(x) \\ u_k(x) r_{k-1}(x) \end{bmatrix}$$

$$\Downarrow$$

$$p_k = 1, \;\; q_k = \mu_k, \;\; d_k = -\rho^*_{k-1}\rho_k, \;\; g_k = \rho^*_{k-1}, \;\; b_k = \mu_{k-1}, \;\; h_k = -\mu_{k-1}\rho_k$$

$$\Downarrow$$

$$\begin{bmatrix} G_0(x) \\ r_0(x) \end{bmatrix} = \frac{1}{\mu_1} \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} G_k(x) \\ r_k(x) \end{bmatrix} = \frac{1}{\mu_k} \begin{bmatrix} 1 & -\rho^*_k \\ -\rho_k & 1 \end{bmatrix} \begin{bmatrix} G_{k-1}(x) \\ x r_{k-1}(x) \end{bmatrix}$$

Numerical Illustrations - Traub

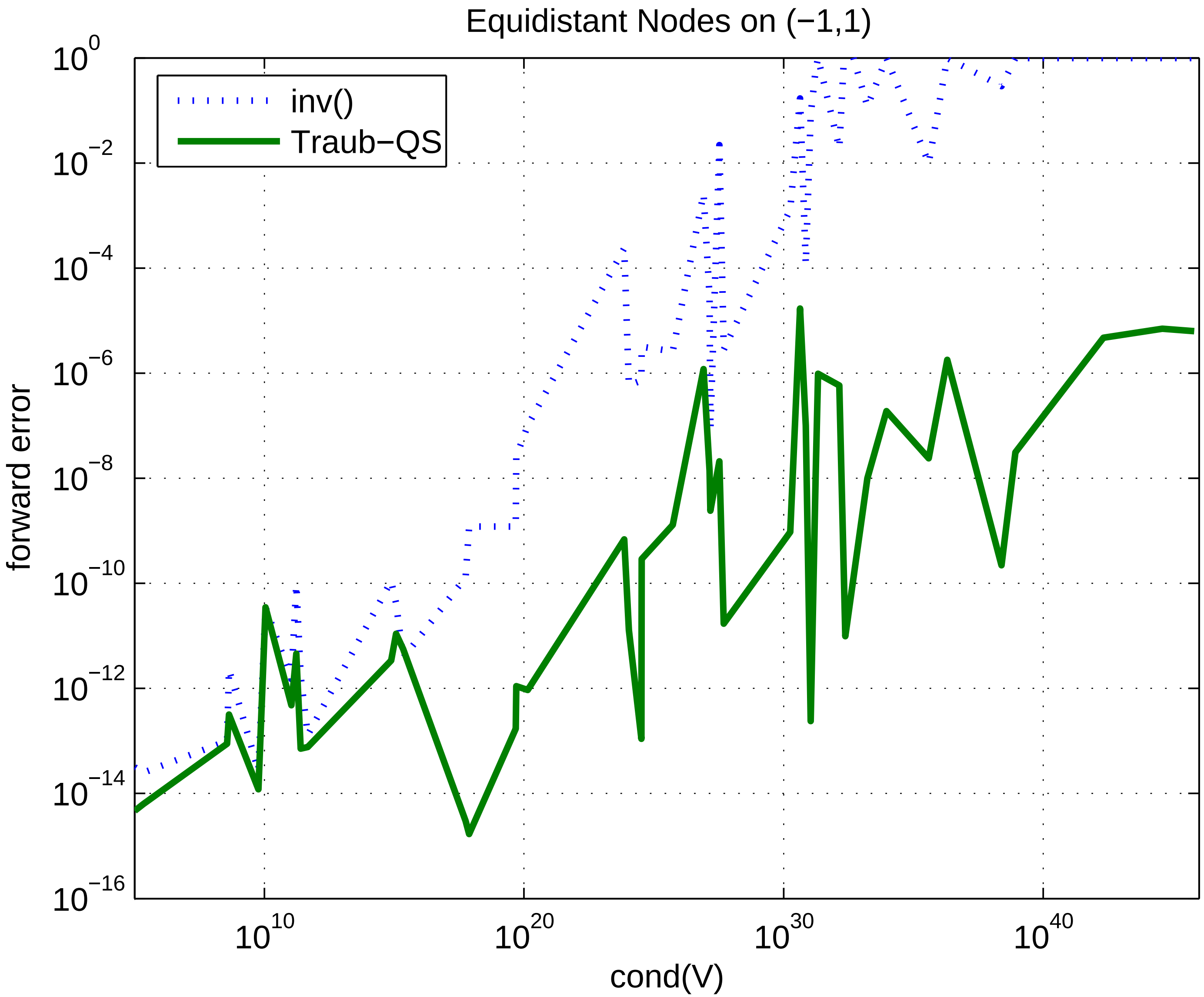
- We compare the **forward error** of the inverse $\widehat{A^{-1}}$ from MATLAB in double precision via

$$e = \frac{\|A^{-1} - \widehat{A^{-1}}\|_2}{\|A^{-1}\|_2},$$

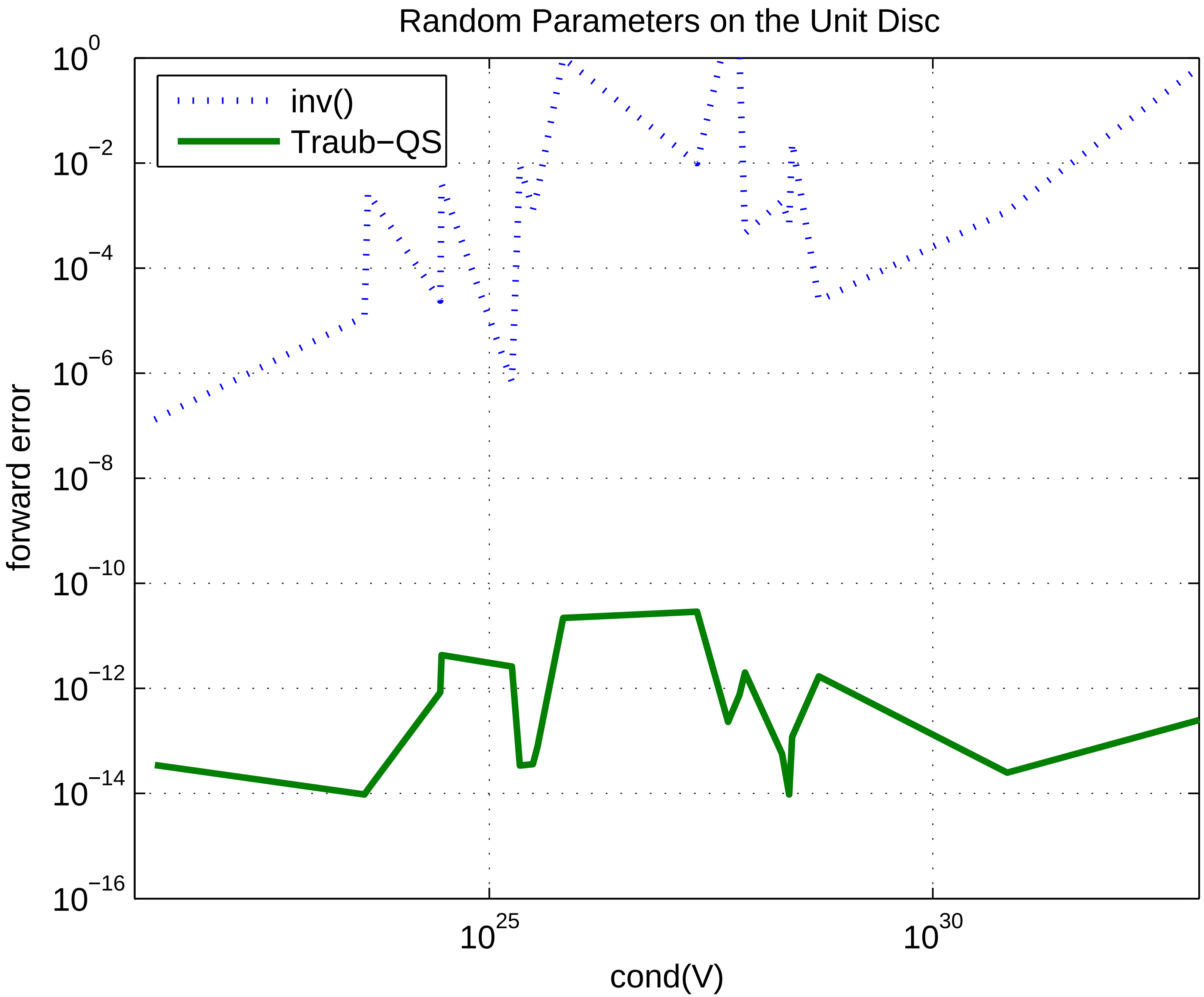
with A^{-1}, the "exact" solution using MATLAB's `vpa()` command for software-implemented arbitrary digit arithmetic.

- **GE** - Gaussian elimination via MATLAB's `inv()` command.
- **Traub-QS** - Traub-like algorithm with nodes ordered via the **Leja ordering**. (Reichel, Higham)

Numerical Illustrations - Traub Experiment 1



Numerical Illustrations - Traub Experiment 2



Fast algorithms for polynomial-Vandermonde matrices related to quasiseparable matrices

Tom Bella

Department of Mathematics

University of Connecticut

Storrs, CT 06269 USA

Based on joint work with Yuli Eidelman, Israel Gohberg, & Vadim Olshevsky.

Fast algorithms for polynomial-Vandermonde matrices related to quasiseparable matrices

Tom Bella
Department of Mathematics
University of Connecticut
Storrs, CT 06269 USA

Based on joint work with Yuli Eidelman, Israel Gohberg, & Vadim Olshevsky.