The generalized Björck-Pereyra algorithm for Szegö-Vandermonde matrices based on properties of unitary Hessenberg matrices

Tom Bella

Department of Mathematics

University of Connecticut

Storrs, CT 06269 USA

Generalized Björck-Pereyra alg. for Szegö-Vandermonde matrices

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

Vandermonde Systems

- **Vandermonde** - Linear systems $Va = f$ with V of the form

$$V = \begin{bmatrix} 1 & x_1 & x_1^2 & \cdots & x_1^{n-1} \\ 1 & x_2 & x_2^2 & \cdots & x_2^{n-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & \cdots & x_n^{n-1} \end{bmatrix}$$

- **Polynomial-Vandermonde** - Linear systems $V_P a = f$ with V_P of the form

$$V_P = \begin{bmatrix} p_0(x_1) & p_1(x_1) & p_2(x_1) & \cdots & p_{n-1}(x_1) \\ p_0(x_2) & p_1(x_2) & p_2(x_2) & \cdots & p_{n-1}(x_2) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ p_0(x_n) & p_1(x_n) & p_2(x_n) & \cdots & p_{n-1}(x_n) \end{bmatrix}$$

for a system of polynomials $P = \{p_0(x), p_1(x), \ldots, p_{n-1}(x)\}$.

The Björck-Pereyra Algorithm

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

$$V^{-1} = \begin{bmatrix} 1 & x_1 & x_1^2 & \cdots & x_1^{n-1} \\ 1 & x_2 & x_2^2 & \cdots & x_2^{n-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & \cdots & x_n^{n-1} \end{bmatrix}^{-1} = U_1^{-1} \ldots U_{n-1}^{-1} L_{n-1}^{-1} \ldots L_1^{-1},$$

with

$$U_k^{-1} = \begin{bmatrix} 1 & -\gamma_1^{(k)} & & \\ & 1 & \ddots & \\ & & \ddots & -\gamma_n^{(k)} \\ & & & 1 \end{bmatrix}, \qquad L_k^{-1} = \begin{bmatrix} \delta_0^{(k)} & & & \\ & \delta_1^{(k)} & & \\ & & \ddots & \\ & & & \delta_n^{(k)} \end{bmatrix} \begin{bmatrix} 1 & & & \\ -1 & 1 & & \\ & \ddots & \ddots & \\ & & -1 & 1 \end{bmatrix},$$

➠ **Fast:** requires only $O(n^2)$ arithmetic operations vs $O(n^3)$ of Gaussian elimination.

Björck-Pereyra-like Algorithms

Tang-Golub (1981)	block Vandermonde matrices
Reichel-Opfer (1991)	Chebyshev-Vandermonde matrices
Higham (1988,90)	three-term Vandermonde matrices

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

What about the Szegö polynomials?

The Szegö Polynomials

- **Orthogonal on the unit circle** with respect to some weight function.

$$< p(x), q(x) >= \frac{1}{2\pi}\int_{-\pi}^{\pi} p(e^{i\theta}) \cdot [q(e^{i\theta})]^* w^2(\theta) d\theta.$$

- Satisfy **two-term recurrence relations**

$$\begin{bmatrix} \phi_0(x) \\ \phi_0^{\#}(x) \end{bmatrix} = \frac{1}{\mu_0}\begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad \begin{bmatrix} \phi_{i+1}(x) \\ \phi_{i+1}^{\#}(x) \end{bmatrix} = \frac{1}{\mu_{i+1}}\begin{bmatrix} 1 & -\rho_{i+1} \\ -\rho_{i+1}^* & 1 \end{bmatrix}\begin{bmatrix} x & 0 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} \phi_i(x) \\ \phi_i^{\#} \end{bmatrix},$$

where ρ_k are the **reflection coefficients**, and

$$\mu_k = \begin{cases} 1 & |\rho_k| = 1 \\ \sqrt{1 - |\rho_k|^2} & \text{otherwise} \end{cases}$$

are the **complementary parameters**

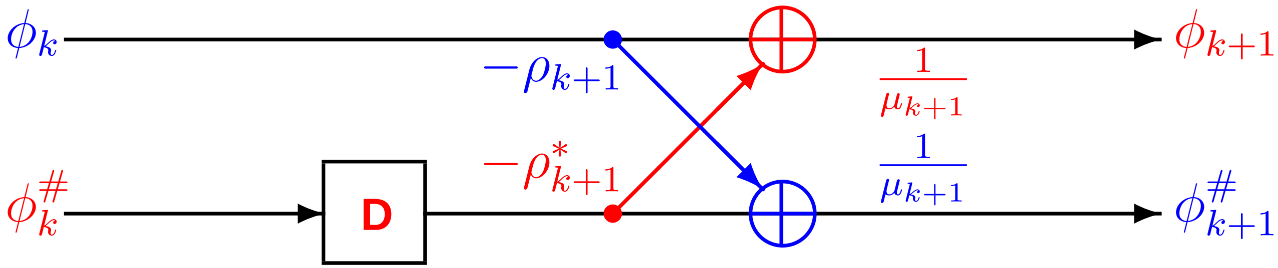
Applications of the Szegö Polynomials

- **Numerical Linear Algebra** - Related to Gaussian quadrature on the unit circle, direct and inverse unitary eigenvalue problems, etc.

  W.Gragg(82,86,97), G.Ammar, W.Gragg, L.Reichel(86,87)

- **Signal Processing** - Discrete transmission lines, and corresponding realizations are used in signal modeling, speech processing, adaptive filtering, etc. Remez algorithm.

  V.Olshevsky(98), M.Morf(74), E.Remez(57)



- **Operator Theory** - Associated with Naimark dilation.

  M.Bakonyi, T.Constantinescu (92), C.Foias, A.Frazho(89)

New Björck-Pereyra-type Algorithm for the Szegö polynomials

- based on the formula

$$V^{-1} = \begin{bmatrix} \phi_0(x_1) & \phi_1(x_1) & \phi_2(x_1) & \cdots & \phi_{n-1}(x_1) \\ \phi_0(x_2) & \phi_1(x_2) & \phi_2(x_2) & \cdots & \phi_{n-1}(x_2) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \phi_0(x_n) & \phi_1(x_n) & \phi_2(x_n) & \cdots & \phi_{n-1}(x_n) \end{bmatrix}^{-1} = U_1^{-1} \dots U_{n-1}^{-1} L_{n-1}^{-1} \dots L_1^{-1},$$

with

$$U_k = \left[\begin{array}{c|ccc} \frac{1}{\alpha_0} & & & \\ 0 & & & \\ \vdots & & \mathbf{H - x_k I} & \\ 0 & & & \\ \hline 0 & 0 & \cdots & 0 \quad \frac{1}{\alpha_{n-k}} \end{array}\right], \qquad L_k^{-1} = \begin{bmatrix} \delta_0^{(k)} & & & \\ & \delta_1^{(k)} & & \\ & & \ddots & \\ & & & \delta_n^{(k)} \end{bmatrix} \begin{bmatrix} 1 & & & \\ -1 & 1 & & \\ & \ddots & \ddots & \\ & & -1 & 1 \end{bmatrix},$$

- H is **Unitary Hessenberg**.

Unitary Hessenberg Matrices

$$H = \begin{bmatrix} * & * & * & * & * \\ * & * & * & * & * \\ 0 & * & * & * & * \\ 0 & 0 & * & * & * \\ 0 & 0 & 0 & * & * \end{bmatrix}$$

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

Factorizations of the Unitary Hessenberg Matrix H

- Decomposition of H into the product of Plane Rotations.

$$\begin{bmatrix} * & * & * & * & * \\ * & * & * & * & * \\ 0 & * & * & * & * \\ 0 & 0 & * & * & * \\ 0 & 0 & 0 & * & * \end{bmatrix} = \begin{bmatrix} * & * & & & \\ * & * & & & \\ & & 1 & & \\ & & & 1 & \\ & & & & 1 \end{bmatrix} \begin{bmatrix} 1 & & & & \\ & * & * & & \\ & * & * & & \\ & & & 1 & \\ & & & & 1 \end{bmatrix} \cdots$$

- Factorization of $H - x_k I$ capturing the shifts.

$$\begin{bmatrix} * & * & * & * & * \\ * & * & * & * & * \\ 0 & * & * & * & * \\ 0 & 0 & * & * & * \\ 0 & 0 & 0 & * & * \end{bmatrix} = \begin{bmatrix} * & 0 & 0 & & \\ * & 0 & * & & \\ * & 1 & * & & \\ & & & 1 & \\ & & & & 1 \end{bmatrix} \begin{bmatrix} 1 & & & & \\ & * & 0 & 0 & \\ & * & 0 & * & \\ & * & 1 & * & \\ & & & & 1 \end{bmatrix} \cdots$$

Numerical Illustrations - 30×30 Matrices

- We compare the **forward accuracy** of $\hat{x}$ from MATLAB in double precision by

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

  x is "exact" solution from Maple v7 using software-implemented 40-digit arithmetic.

- $\log e$ gives estimate of number of correct significant figures out of 16.

- The algorithms
  - **GBP**/**GBP-IS** - Generalized BP alg. using plane rotations and implicit shifts, resp.
  - **GE2t**/**GE3t** - Gaussian elimination with matrix derived from 2-term and 3-term recurrence relations, resp.

- The parameters
  - ρ_k - Reflection coefficients
  - x_k - Nodes
  - b_k - Entries in right-hand-side vector.

Numerical Illustrations - Experiment 1

Ill-Conditioned Matrices

#	cond(V)	GBP	GBP-IS	GE2t	GE3t
1	7e+14	5e-15	9e-15	8e-06	1e-05
2	1e+15	2e-15	2e-15	5e-05	8e-05
3	3e+15	3e-15	4e-15	6e-04	2e-04
4	1e+18	2e-15	2e-15	1e-01	7e-01
5	2e+15	4e-15	1e-15	4e-04	4e-04
6	5e+17	1e-14	1e-14	5e-02	3e-02
7	1e+16	4e-15	2e-15	2e-05	4e-05
8	1e+18	1e-15	1e-15	2e-02	1e-02
9	1e+18	2e-15	1e-15	1e-00	1e-00
10	9e+18	6e-16	8e-16	5e-01	7e-01

#	cond(V)	GBP	GBP-IS	GE2t	GE3t
1	8e+49	3e-14	3e-14	1e-00	1e-00
2	9e+51	1e-14	1e-14	9e-01	1e-00
3	9e+48	8e-15	8e-15	1e-01	3e-01
4	8e+49	7e-15	8e-15	2e-02	2e-01
5	1e+52	2e-15	2e-15	1e-00	3e-01
6	6e+50	1e-13	1e-13	8e-01	1e-00
7	2e+51	2e-14	2e-14	3e-01	3e-01
8	6e+51	1e-15	1e-15	2e-03	2e-02
9	2e+51	1e-15	9e-16	1e-00	1e-00
10	3e+50	2e-15	2e-15	1e-00	1e-00

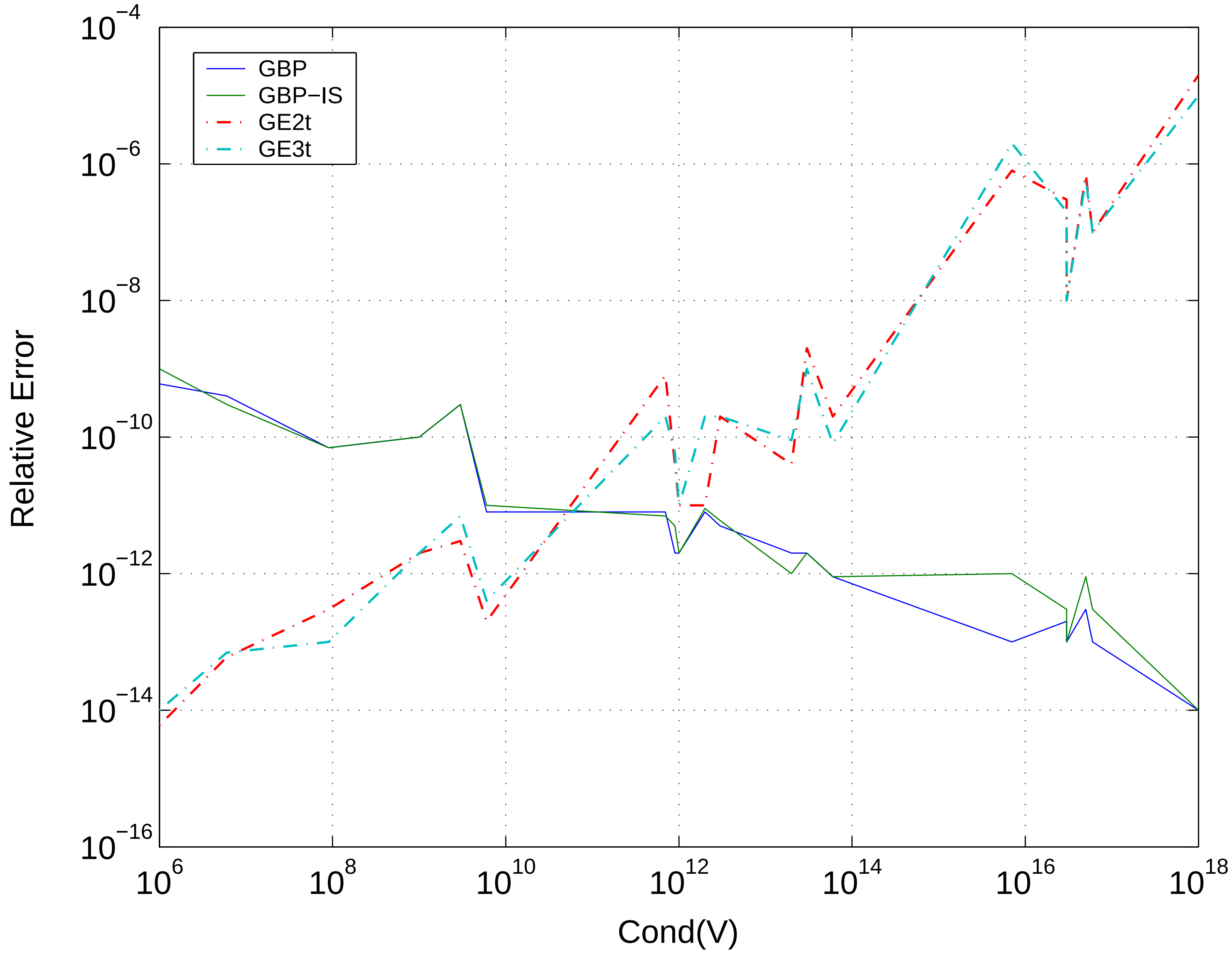
Numerical Illustrations - Experiment 2

Better Conditioned Matrices

#	cond(V)	GBP	GBP-IS	GE2t	GE3t
1	4e+08	3e-11	4e-11	1e-12	3e-12
2	2e+07	8e-11	1e-10	2e-14	5e-14
3	9e+05	1e-09	1e-09	9e-15	3e-14
4	3e+08	3e-11	3e-11	2e-13	2e-13
5	6e+09	2e-09	2e-09	1e-13	3e-13
6	1e+09	3e-11	1e-10	3e-14	8e-15
7	5e+05	3e-11	4e-11	1e-14	3e-14
8	3e+06	7e-11	9e-11	1e-13	7e-14
9	7e+05	1e-08	1e-08	2e-14	6e-14
10	2e+06	1e-10	3e-10	1e-13	1e-13

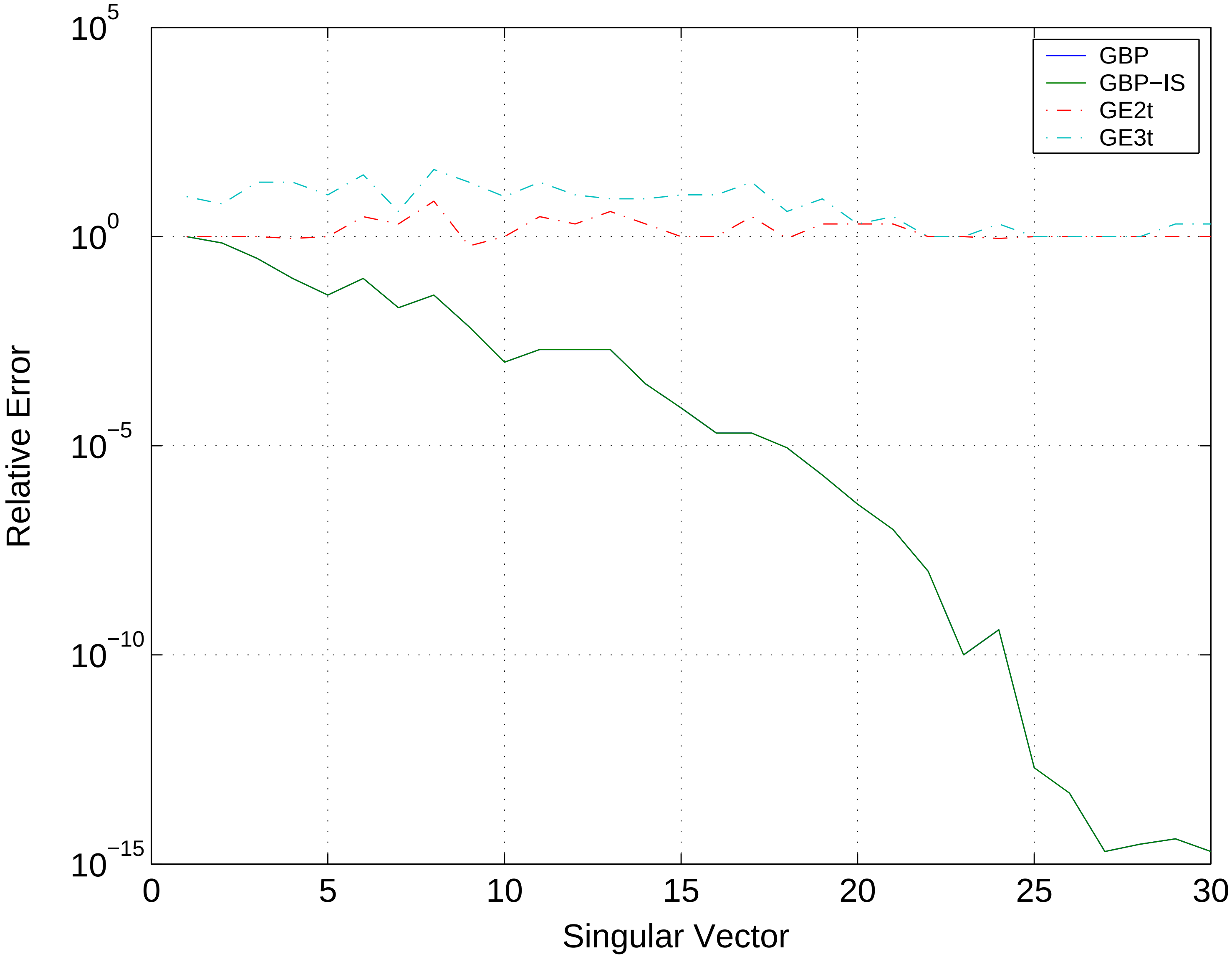
Numerical Illustrations - Experiment 3

Effects of Conditioning



Numerical Illustrations - Experiment 4

Dependence on Direction



Numerical Illustrations - Experiment 5

Iterative Refinement

		no refinement		iterative refinement			
#	cond(V)	GBP	GBP-IS	GBP	GBP-IS	GE2t	GE3t
1	1e+07	2e-09	4e-09	1e-14	9e-15	4e-13	6e-13
2	3e+07	2e-10	2e-10	4e-14	7e-15	4e-14	8e-14
3	1e+08	2e-10	2e-10	4e-14	2e-14	9e-13	2e-13
4	6e+08	6e-11	6e-11	3e-14	1e-14	4e-13	4e-13
5	1e+07	3e-10	5e-10	4e-15	4e-15	1e-13	2e-13
6	5e+05	5e-10	3e-10	1e-15	9e-16	4e-14	7e-14
7	3e+06	3e-10	1e-10	1e-15	1e-15	2e-14	2e-13
8	1e+08	7e-10	6e-10	3e-15	4e-15	4e-14	3e-14
9	2e+07	8e-10	5e-10	2e-15	3e-15	1e-13	9e-14
10	3e+07	5e-10	5e-10	3e-14	3e-14	8e-14	1e-13

Numerical Illustrations - Summary

- **Experiment 1** - **GBP** & **GBP-IS** outperform **GE** for ill-conditioned and **very** ill-conditioned systems.
- **Experiment 2** - **GBP** & **GBP-IS** do not necessarily perform better than **GE** for well-conditioned systems.
- **Experiment 3** - As the nodes x_k are perturbed from the roots of $\phi_n(x)$, **GBP** & **GBP-IS** perform worse than **GE** until the condition number $\approx 10^{12}$, then outperform **GE**.
- **Experiment 4** - Unlike **GE** which is insensitive to the direction of the RHS vector, **GBP** & **GBP-IS** are affected by the direction of the RHS vector.
- **Experiment 5** - With one step of **iterative refinement**, the **GBP** & **GBP-IS** both perform very well in the previously noted cases where performance was poor.

Future Directions

- Additional experimentation
- Error Analysis to confirm observations

Generalized Björck-Pereyra alg. for Szegö-Vandermonde matrices

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

Part II. Equivalence of Hadamard and Pseudo-Noise Matrices

Joint work with V.Olshevsky and L.Sakhnovich

Hadamard Matrices

Hadamard matrices of size n x n, are $(-1, 1)$ matrices such that

$$H_n^T H_n = nI_n$$

A special case: **Hadamard-Sylvester matrices**

$$H_1 = [1], \qquad H_{2n} = \begin{bmatrix} H_n & H_n \\ H_n & -H_n \end{bmatrix}$$

For example,

$$H_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \qquad H_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}$$

Linear Recurring Sequences. A Matrix Interpretation

- Linear **m-term** recurrence relation:

$$\mathbf{a_i} = \mathbf{a_{i-1}}\mathbf{h_{m-1}} + \mathbf{a_{i-2}}\mathbf{h_{m-2}} + \cdots + \mathbf{a_{i-m+1}}\mathbf{h_1} + \mathbf{a_{i-m}}\mathbf{h_0} \quad \text{for } i \geq m$$

- $m \times m$ Matrix formulation

$$\begin{bmatrix} \mathbf{a_{i-(m-1)}} \\ \mathbf{a_{i-(m-2)}} \\ \mathbf{a_{i-(m-3)}} \\ \vdots \\ \mathbf{a_{i-1}} \\ \mathbf{a_i} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & \dots & 0 \\ 0 & 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 0 & 1 & \dots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \ddots & 0 \\ 0 & 0 & 0 & \dots & 0 & 1 \\ \mathbf{h_0} & \mathbf{h_1} & \mathbf{h_2} & \dots & \mathbf{h_{m-2}} & \mathbf{h_{m-1}} \end{bmatrix} \begin{bmatrix} \mathbf{a_{i-(m)}} \\ \mathbf{a_{i-(m-1)}} \\ \mathbf{a_{i-(m-2)}} \\ \vdots \\ \mathbf{a_{i-2}} \\ \mathbf{a_{i-1}} \end{bmatrix}$$

- The characteristic polynomial of **degree m**:

$$h(x) = x^m + \mathbf{h_{m-1}}x^{m-1} + \cdots + \mathbf{h_1}x + \mathbf{h_0}$$

PN Sequences & Matrices

- If the vectors $\begin{bmatrix} \mathbf{a_{i-(m-1)}} & \mathbf{a_{i-(m-2)}} & \cdots & \mathbf{a_i} \end{bmatrix}^T$ go through all possible $2^m - 1$ nonzero states, the sequence $a_0 a_1 a_2 \cdots$ is called a **Pseudo Noise sequence**.
- A PN sequence generated by an **m**-degree polynomial is **periodic** with period $2^m - 1$.
- **Example:** For $h(x) = x^4 + x^3 + 1$ ($m = 4$), and the initial state $a_0 a_1 a_2 a_3 = 1000$:

$$\underbrace{100011110101100}_{\text{period 15}}\underbrace{100011110101100}_{\text{period 15}}\underbrace{100011110101100}_{\text{period 15}}\ldots\ldots$$

- A **Pseudo Noise Matrix** is one of the form

$$T = \begin{bmatrix} 0 & 0 & \ldots & 0 \\ 0 & & & \\ \vdots & & \widetilde{T} & \\ 0 & & & \end{bmatrix}$$

where $\widetilde{T}$ is a **circulant Hankel** matrix whose rows are **PN sequences**.

Example

- For the **PN Sequence** listed above

$$\boxed{100011110101100}\ \boxed{100011110101100}\ \boxed{100011110101100}\ \ldots$$

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

these $m = 4$ columns form a basis of the column space of $\widetilde{T}$

Theorem

- The $(0, 1)$ Hadamard-Sylvester matrices and the $(0, 1)$ PN matrices are equivalent, i.e., they can be obtained one from another via row and column permutations.
- Our proof applies to arbitrary n, and it is based on the observation that the $(0, 1)$ Hadamard-Sylvester matrices of the size $2^m \times 2^m$ have rank m.

Part II. Equivalence of Hadamard and Pseudo-Noise Matrices

Joint work with V.Olshevsky and L.Sakhnovich

Supplemental Slides

Unitary Hessenberg Matrices

$$
H = \begin{bmatrix}
-\rho_1\rho_0^* & -\rho_2\mu_1\rho_0^* & \dots & \dots & -\rho_{n-1}\mu_{n-2}\cdots\mu_1\rho_0^* & -\rho_n\mu_{n-1}\cdots\mu_1\rho_0^* \\
\mu_1 & -\rho_2\rho_1^* & \dots & \dots & -\rho_{n-1}\mu_{n-2}\cdots\mu_2\rho_1^* & -\rho_n\mu_{n-1}\cdots\mu_2\rho_1^* \\
0 & \mu_2 & \dots & \dots & -\rho_{n-1}\mu_{n-2}\cdots\mu_3\rho_2^* & -\rho_n\mu_{n-1}\cdots\mu_3\rho_2^* \\
\vdots & \ddots & \ddots & \ddots & \vdots & \vdots \\
\vdots & & \ddots & \ddots & -\rho_{n-1}\rho_{n-2}^* & \vdots \\
0 & \dots & \dots & 0 & \mu n-1 & -\rho_n\rho_{n-1}^*
\end{bmatrix}
$$

Definition of $^{\#}$ Notation

- Following notation commonly used in signal processing literature, we denote

$$f_k^{\#}(z) = z^k \left[f_k \left(\frac{1}{z^*} \right) \right]^*$$

- This **reverses the order** of the coefficients, and takes **complex conjugates**.
- **Example:** For $n = 3$,

$$f(z) = a_3 z^3 + a_2 z^2 + a_1 z + a_0$$

$$f(z)^{\#} = a_0^* z^3 + a_1^* z^2 + a_2^* z + a_3^*$$

Three-Term Recurrence Relations for Szegö Polynomials

$$\phi_0(x) = 1, \qquad \phi_1(x) = \frac{1}{\mu_1}(x \cdot \phi_0(x) - \rho_1 \cdot \phi_0(x)),$$

$$\phi_k(x) = [\frac{1}{\mu_k} \cdot x + \frac{\rho_k}{\rho_{k-1}} \frac{1}{\mu_k}] \cdot \phi_{k-1}(x) - \frac{\rho_k}{\rho_{k-1}} \frac{\mu_{k-1}}{\mu_k} \cdot x \cdot \phi_{k-2}(x).$$

Decomposition of H into Plane Rotations

- The details of the decomposition of H into plane rotations:

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

where

$$H(\rho_k) = \text{diag}\left\{ I_{k-1}, \begin{bmatrix} \rho_k & \mu_k \\ \mu_k & -\rho_k^* \end{bmatrix}, I_{n-k-1} \right\}$$

$$\tilde{H}(\rho_k) = \text{diag}\left\{ I_{n-1}, \rho_k \right\}$$

Decomposition of H into Implicit Shifts

➠ The details of a factorization of H using **Implicit Shifts**:

$$H - x_k I = R_0 \times R_1 \times \cdots \times R_{n-1} \times R_n$$

where

$$R_0 = \text{diag}\left\{\begin{bmatrix} 1 & -\rho_0^* \end{bmatrix}, I_{n-1}\right\}$$

$$R_k = \text{diag}\left\{ I_{k-1}, \begin{bmatrix} -x_k & 0 & 0 \\ \rho_k & 0 & \mu_k \\ \mu_k & 1 & -\rho_k^* \end{bmatrix}, I_{n-k-1} \right\}$$

for $k = 1, 2, \ldots, n-1$, and

$$R_n = \text{diag}\left\{ I_{n-1}, \begin{bmatrix} -x_n \\ \rho_n \end{bmatrix} \right\}$$

Definition

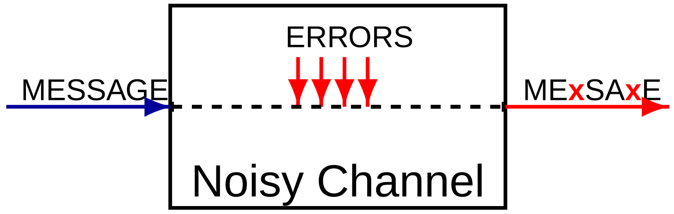
- A matrix $\mathbf{A}$ is **quasiseparable of order one** if

$$\max \mathsf{Rank} A_{12} = \max \mathsf{Rank} A_{21} = 1$$

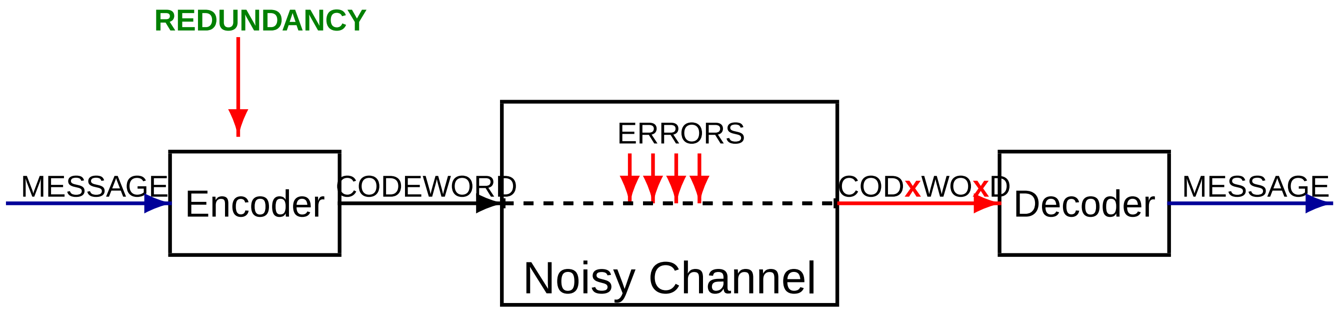
where the maxima are taken over all symmetric partitions of the form

$$\mathbf{A} = \left[\begin{array}{c|c} * & \mathbf{A_{12}} \\ \hline \mathbf{A_{21}} & * \end{array}\right]$$

Transmission over Noisy Channel, No Coding Theory



Transmission over Noisy Channel, With Coding Theory



A Decomposition of PN Matrices

- **Observation:** the PN matrix has rank m
- A **Decomposition** based on this observation:

$$\tilde{T} = \begin{bmatrix} M_1 \\ M_2 \end{bmatrix} \begin{bmatrix} I & R \end{bmatrix}$$

where $\begin{bmatrix} M_1 \\ M_2 \end{bmatrix}$ is a $2^m - 1$ x m matrix, and $\begin{bmatrix} I & R \end{bmatrix}$ is an m x $2^m - 1$ matrix.

- Uniqueness of rows of $\tilde{T}$ imply uniqueness of rows in $\begin{bmatrix} M_1 \\ M_2 \end{bmatrix}$
- Uniqueness of columns of $\tilde{T}$ imply uniqueness of columns in $\begin{bmatrix} I & R \end{bmatrix}$
- This uniqueness and the sizes of these matrices imply they contain **all possible binary m-tuples** as rows and columns.

A Decomposition of Hadamard-Sylvester Matrices

- Let H'_n denote the Hadamard-Sylvester matrix H_n with $(1, -1) \to (0, 1)$.
- Then H'_n has the decomposition

$$H'_n = \mathbf{L_m} \mathbf{L_m^T}$$

where $L_m = [l_{ij}]$ is an n x m matrix with $l_{ij} \in \{0, 1\}$ and the rows of L_m corresponding to all possible binary m-tuples.

Proof

This decomposition can be seen inductively:

- For $H'_2 = L_1 L_1^T$, we have $L_1 = [\,0 \;\; 1\,]^T$

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

- Assuming $H'_m = L_m L_m^T$ is true for size m one sees that

$$H'_{(m+1)} = \begin{bmatrix} H'_m & H'_m \\ H'_m & ''-''H'_m \end{bmatrix} = \underbrace{\begin{bmatrix} 0_m & \mathbf{L_m} \\ 1_m & \mathbf{L_m} \end{bmatrix}}_{L_{m+1}} \underbrace{\begin{bmatrix} {0_m}^T & {1_m}^T \\ \mathbf{L_m}^T & \mathbf{L_m}^T \end{bmatrix}}_{L_{m+1}^T}$$

where $0_m = \underbrace{[0 \;\; 0 \;\; \ldots \;\; 0]^T}_{m \text{ zeros}}$ $\qquad 1_m = \underbrace{[1 \;\; 1 \;\; \ldots \;\; 1]^T}_{m \text{ ones}}$

This demonstrates the equivalence!