Accurate and Efficient Matrix Computations with Vandermonde Matrices Using Schur Functions

---

Plamen Koev
Department of Mathematics
University of California - Berkeley

Joint work with Prof. James Demmel

Supported by NSF and DOE

MIT, March 15, 2002

GOALS

- **Accurate (Small relative error) and Efficient ($O(n^3)$ or perhaps $O(n^p)$, independent of condition number) Linear Algebra**
  - A^{-1}
  - $Ax = b$
  - **LDU from GENP, GEPP, GECP**
  - **SVD**
- **Can't be done for general matrices, must be "structured"**
  - **Certain sparsity patterns**
  - **Cauchy**
  - **Vandermonde**
  - **...**
- **Goal of this talk: Accurate and Efficient Linear Algebra for Generalized Vandermonde Matrices**

Type of Matrix	$\det(A)$	A^{-1}	**Any minor**	**GENP GEPP**	**GECP**	**SVD**	**Small Forward Error in** $Ax = b$	**Small Backward Error in** $Ax = b$
Cauchy								
Totally Positive Cauchy								
Vandermonde								
Totally positive Vandermonde								
Polynomial Vandermonde								
Poly. Vand. Orth. poly.								
Generalized Vandermonde								
TP Generalized Vandermonde								

Totally Positive = Matrix with all minors > 0

OUTLINE

- **Model of arithmetic**
- **Classical method for achieving the goals for simple examples – The Björck-Pereyra Method for Vandermonde Matrices**
- **How and why it works?**
- **Application to TP Generalized Vandermonde matrices**

How can we lose accuracy in computing in floating point?

- $\mathrm{fl}(a \otimes b) = (a \otimes b)(1+\delta)$ **model of arithmetic with no over/underflow**
- **OK to multiply, divide, add positive numbers**
  Proof: $1+\delta$ **factors can be factored out**
- $x_i \pm x_j$, **where** x_i **and** x_j **are initial data (so exact)**
- $(x_i + y_j)(x_i - y_{j-1})x_{i+1}/(x_{i-1} - y_j)$ **- OK**
- **Cancellation when subtracting approximate results dangerous:**

```
   .12345xxx
 - .12345yyy
 -----------
   .00000zzz
```

- **We will compute everything using only allowable expressions**
- **Explains our interest in TP matrices, minors** > 0

Classical Example: A Vandermonde Linear System

- **Solve $Vy = b$, where V is Vandermonde:**

$$\begin{bmatrix} 1 & x_1 & \dots & x_1^{n-1} \\ 1 & x_2 & \dots & x_2^{n-1} \\ 1 & x_3 & \dots & x_3^{n-1} \\ & & \ddots & \\ 1 & x_n & \dots & x_n^{n-1} \end{bmatrix} \cdot \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} + \\ - \\ + \\ \vdots \\ - \end{bmatrix}$$

  and $0 < x_1 < \ldots < x_n$.

- **Equivalent to interpolation**
  - **Lagrange**
  - **Newton**
- **The Björck-Pereyra method solves $Vy = b$, by doing Newton interpolation**
  - **In $O(n^2)$ time**
  - **With small forward error: $|y_i - \hat{y}_i| \le O(\epsilon)|y_i|$**
  - **With small backward error: If $\hat{V}\hat{y} = b$ then $|V_{ij} - \hat{V}_{ij}| \le O(\epsilon)|V_{ij}|$.**
- **How does it work?**

The Björck-Pereyra Method (1970)

Polynomial interpolation:

x	1	2	3
$f(x)$	2	-1	14

by $f(x) = a_1 + a_2 x + a_3 x^2$

(alternating signs for $f(x)$ **for a purpose)**

E.g. solve: $\begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 2^2 \\ 1 & 3 & 3^2 \end{bmatrix} \cdot \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} 2 \\ -1 \\ 14 \end{bmatrix}$. **GE:** $\begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 4 \\ 1 & 3 & 9 \end{bmatrix} \sim \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 3 \\ 0 & 2 & 8 \end{bmatrix} \sim \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 3 \\ 0 & 0 & 2 \end{bmatrix}$

Instead: Newton Interpolation:

x	$f(x)$	$f[\cdot,\cdot]$	$f[\cdot,\cdot,\cdot]$
1	2	2	2
2	-1	$\frac{-1-2}{2-1} = -3$	-3
3	14	$\frac{14-(-1)}{3-2} = 15$	$\frac{15-(-3)}{3-1} = 9$

$$f(x) = 2 - 3(x-1) + 9(x-1)(x-2) =$$

$$2 - (x-1)[3 - 9(x-2)] = 2 - (x-1)(21 - 9x) = 2 + 21 - (9+21)x + 9x^2 = 23 - 30x + 9x^2$$

Matrix Interpretation of the Björck-Pereyra Method

x	$f(x)$	$f[\cdot,\cdot]$	$f[\cdot,\cdot,\cdot]$
1	2	2	2
2	-1	-3	-3
3	14	15	9

Matrix version: $\begin{bmatrix} 1 & & \\ -1 & 1 & \\ & -1 & 1 \end{bmatrix} \cdot \begin{bmatrix} 2 \\ -1 \\ 14 \end{bmatrix} = \begin{bmatrix} 2 \\ -3 \\ 15 \end{bmatrix}; \quad \begin{bmatrix} 1 & & \\ & 1 & \\ & -\frac{1}{2} & \frac{1}{2} \end{bmatrix} \cdot \begin{bmatrix} 2 \\ -3 \\ 15 \end{bmatrix} = \begin{bmatrix} 2 \\ -3 \\ 9 \end{bmatrix}$

yielding $f(x) = 2 - 3(x-1) + 9(x-1)(x-2)$
Now going back to the solution:

$$\begin{bmatrix} 1 & & \\ & 1 & -2 \\ & & 1 \end{bmatrix} \cdot \begin{bmatrix} 2 \\ -3 \\ 9 \end{bmatrix} = \begin{bmatrix} 1 \\ -21 \\ 9 \end{bmatrix}; \quad \begin{bmatrix} 1 & -1 & \\ & 1 & -1 \\ & & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ -21 \\ 9 \end{bmatrix} = \begin{bmatrix} 23 \\ -30 \\ 9 \end{bmatrix}$$

yielding $f(x) = 23 - 30x + 9x^2$

Matrix Interpretation of the Björck-Pereyra Method

- **Putting it all together**

$$y = V^{-1}b = \begin{bmatrix} 1 & -1 & \\ & 1 & -1 \\ & & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & & \\ & 1 & -2 \\ & & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & & \\ & 1 & \\ & -\frac{1}{2} & \frac{1}{2} \end{bmatrix} \cdot \begin{bmatrix} 1 & & \\ -1 & 1 & \\ & -1 & 1 \end{bmatrix} \cdot \begin{bmatrix} 2 \\ -1 \\ 14 \end{bmatrix} = \begin{bmatrix} 23 \\ -30 \\ 9 \end{bmatrix}$$

- **Notice:**
  - **Bidiagonal Decomposition of V^{-1} (accurate)**
  - **Checkerboard sign pattern**

$\Rightarrow$ **No subtractive cancellation**
$\Rightarrow$ **High relative accuracy**

- **Questions:**
  - **Which matrices have bidiagonal decomposition of their inverses?**
  - **Checkerboard signs?**
  - **Accurate?**

The Björck-Pereyra Method Dissected

- **Questions:**
  - **Which matrices have bidiagonal decomposition of their inverses?**
  - **Checkerboard signs?**
  - **Accurate?**
- **Answers:**
  - **All nonsingular matrices do**
    This is *Neville elimination* in matrix form:

$$\begin{bmatrix} 1 & & \\ -1 & 1 & \\ & -1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 4 \\ 1 & 3 & 9 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 3 \\ 0 & 1 & 5 \end{bmatrix}; \qquad \begin{bmatrix} 1 & & \\ & 1 & \\ & -\frac{1}{2} & \frac{1}{2} \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 3 \\ 0 & 1 & 5 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 3 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & -1 & \\ & 1 & -1 \\ & & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & & \\ & 1 & -2 \\ & & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & & \\ & 1 & \\ & -\frac{1}{2} & \frac{1}{2} \end{bmatrix} \cdot \begin{bmatrix} 1 & & \\ -1 & 1 & \\ & -1 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 4 \\ 1 & 3 & 9 \end{bmatrix} = \begin{bmatrix} 1 & & \\ & 1 & \\ & & 1 \end{bmatrix}$$

  - **Checkerboard sign pattern $\Longleftrightarrow$ Total positivity**
    (A is TP $\Longleftrightarrow$ all minors > 0)
  - **Accurate? Yes.**

Accuracy of the Björck-Pereyra Method

$$\left(\begin{bmatrix} 1 & x_1 & x_1^2 & x_1^3 \\ 1 & x_2 & x_2^2 & x_2^3 \\ 1 & x_3 & x_3^2 & x_3^3 \\ 1 & x_4 & x_4^2 & x_4^3 \end{bmatrix}\right)^{-1} = \begin{bmatrix} 1 & -x_1 & & \\ & 1 & -x_1 & \\ & & 1 & -x_1 \\ & & & 1 \end{bmatrix} \begin{bmatrix} 1 & & & \\ & 1 & -x_2 & \\ & & 1 & -x_2 \\ & & & 1 \end{bmatrix} \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1 & -x_3 \\ & & & 1 \end{bmatrix} \times$$

$$\begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & \frac{-1}{x_4-x_1} & \frac{1}{x_4-x_1} \end{bmatrix} \begin{bmatrix} 1 & & & \\ & 1 & & \\ & \frac{-1}{x_3-x_1} & \frac{1}{x_3-x_1} & \\ & & \frac{-1}{x_4-x_2} & \frac{1}{x_4-x_2} \end{bmatrix} \begin{bmatrix} 1 & & & \\ \frac{-1}{x_2-x_1} & \frac{1}{x_2-x_1} & & \\ & \frac{-1}{x_3-x_2} & \frac{1}{x_3-x_2} & \\ & & \frac{-1}{x_4-x_3} & \frac{1}{x_4-x_3} \end{bmatrix}$$

Accuracy ... OK

Other TP matrices? ... Yes. Cauchy

The Björck-Pereyra Method for Cauchy Matrices

- ***Cauchy*** **matrices**

$$C = \frac{1}{x_i - y_j}$$

- **TP, if** $x_1 > \ldots > x_n > y_1 > \ldots > y_n$

$$\left(\begin{bmatrix} \frac{1}{x_1-y_1} & \frac{1}{x_1-y_2} & \frac{1}{x_1-y_3} \\ \frac{1}{x_2-y_1} & \frac{1}{x_2-y_2} & \frac{1}{x_2-y_3} \\ \frac{1}{x_3-y_1} & \frac{1}{x_3-y_2} & \frac{1}{x_3-y_3} \end{bmatrix}\right)^{-1} = \begin{bmatrix} 1 & \frac{-(x_1-y_1)}{y_1-y_2} & 0 \\ 0 & \frac{x_1-y_2}{y_1-y_2} & \frac{-(x_1-y_2)}{y_2-y_3} \\ 0 & 0 & \frac{x_1-y_3}{y_2-y_3} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & \frac{-(x_2-y_1)}{y_1-y_3} \\ 0 & 0 & \frac{x_2-y_3}{y_1-y_3} \end{bmatrix} \times$$

$$\begin{bmatrix} x_1-y_1 & & \\ & x_2-y_2 & \\ & & x_3-y_3 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & \frac{-(x_1-y_2)}{x_3-x_1} & \frac{x_3-y_2}{x_3-x_1} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ \frac{-(x_1-y_1)}{x_2-x_1} & \frac{x_2-y_1}{x_2-x_1} & 0 \\ 0 & \frac{-(x_2-y_1)}{x_3-x_2} & \frac{x_3-y_1}{x_3-x_2} \end{bmatrix}$$

- **Unifying Characteristic?**

The Connection with Minors

- **Which TP matrices permit *accurate* bidiagonal decomposition?**
- **Each entry is *product of quotients of initial minors***

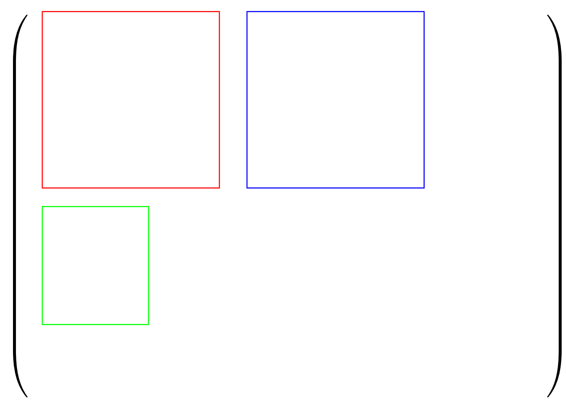
$$L^{(k)}_{i+1,i} = -\frac{\det(A(i-k+2:i+1,1:k))}{\det(A(i-k+2:i,1:k-1))} \cdot \frac{\det(A(i-k+1:i-1,1:k-1))}{\det(A(i-k+1:i,1:k))}$$

INITIAL MINORS

– Contiguous

– Include first row or column

$A =$



- **Initial minors of:**

Cauchy $\frac{\Pi_{i<j}(x_j - x_i)(y_j - y_i)}{\Pi_{i,j}(x_i + y_j)}$; **Vandermonde** $\prod_{i>j}(x_i - x_j)$

- **How did we think of minors?**
- **Gaussian Elimination and Neville Elimination**
  Each entry of $V = LDU$ is a quotient of minors, so not surprising

New results: Generalized Vandermonde Matrices

- **TP Matrices with initial minors that are easy to compute accurately Vandermonde and Generalized Vandermonde**

$$V = \begin{bmatrix} 1 & x_1 & \dots & x_1^{n-1} \\ 1 & x_2 & \dots & x_2^{n-1} \\ & & \ddots & \\ 1 & x_n & \dots & x_n^{n-1} \end{bmatrix}, \qquad G_\lambda = \begin{bmatrix} x_1^{\lambda_n} & x_1^{1+\lambda_{n-1}} & \dots & x_1^{n-1+\lambda_1} \\ x_2^{\lambda_n} & x_2^{1+\lambda_{n-1}} & \dots & x_2^{n-1+\lambda_1} \\ & & \ddots & \\ x_n^{\lambda_n} & x_n^{1+\lambda_{n-1}} & \dots & x_n^{n-1+\lambda_1} \end{bmatrix},$$

where $x_1 > x_2 > \cdots > x_n > 0$, $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n \geq 0$, $|\lambda| = \lambda_1 + \ldots + \lambda_n$

- **Initial Minors for G_λ?**

$$\det(G_\lambda) = \det(V) \cdot s_\lambda(x_1, x_2, \ldots, x_n)$$

- s_λ **- called Schur function**
  - **Polynomial with positive integer coefficients**
  - **Widely studied in combinatorics [MacDonald], group representation theory**

- **Example:**

$$\det\left(\begin{bmatrix} 1 & x_1^2 & x_1^4 \\ 1 & x_2^2 & x_2^4 \\ 1 & x_3^2 & x_3^4 \end{bmatrix}\right) = \det\left(\begin{bmatrix} 1 & x_1 & x_1^2 \\ 1 & x_2 & x_2^2 \\ 1 & x_3 & x_3^2 \end{bmatrix}\right) \cdot (2x_1x_2x_3 + x_1^2x_2 + x_1x_2^2 + x_1^2x_3 + x_1x_3^2 + x_2^2x_3 + x_2x_3^2)$$

Accuracy and Efficiency for Generalized Vandermonde Matrices

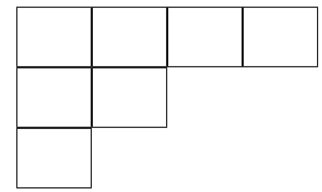
- **Example:**

$$\det\left(\begin{bmatrix} 1 & x_1^2 & x_1^4 \\ 1 & x_2^2 & x_2^4 \\ 1 & x_3^2 & x_3^4 \end{bmatrix}\right) = \det\left(\begin{bmatrix} 1 & x_1 & x_1^2 \\ 1 & x_2 & x_2^2 \\ 1 & x_3 & x_3^2 \end{bmatrix}\right) \cdot (2x_1x_2x_3 + x_1^2x_2 + x_1x_2^2 + x_1^2x_3 + x_1x_3^2 + x_2^2x_3 + x_2x_3^2)$$

- **Accuracy?**
  - $\det(V) = \Pi_{i>j}(x_i - x_j)$ **- YES.**
  - s_λ **- polynomials with** > 0 **coefficients - YES.**
- **Efficiency?**
  - $\det(V) = \Pi_{i>j}(x_i - x_j)$ **- OK.**
  - $s_\lambda(x_1, x_2, \ldots, x_n)$**?**
    - **Has exponentially many terms** $- n^{|\lambda|}$**, we'll look in detail**
  - **We are only interested in subtraction-free algorithms, thus**
    - $s_\lambda = f(\text{elementary symmetric polynomials})$ **- useless**
    - **Same for the Jacobi-Trudi identity**

The Schur Function $s_\lambda(x_1, x_2, \ldots, x_n)$

- **Def:** $\lambda = (\lambda_1, \lambda_2, ..., \lambda_n)$ **is a partition of** $|\lambda| = \lambda_1 + \lambda_2 + \cdots + \lambda_n$
- **Def: Young Diagrams** $\equiv$ **partitions:**

Partition: $7 = 4 + 2 + 1$, **Young diagram:** $\lambda = (4, 2, 1) =$ 

- **Def: Semistandard Young Tableau** T_λ**:**

1	**1**	**2**	**3**
2	**3**		
4			

- **Def:** $x^{T_\lambda} = x_1^2 x_2^2 x_3^2 x_4$
-

$$s_\lambda = \sum_{\substack{T_\lambda \\ \text{semi-std}}} x^{T_\lambda}$$

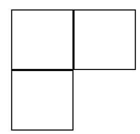
The Schur Function $s_\lambda(x_1, x_2, \ldots, x_n)$

$$s_\lambda = \sum_{\substack{T_\lambda \\ \text{semi-std}}} x^{T_\lambda}$$

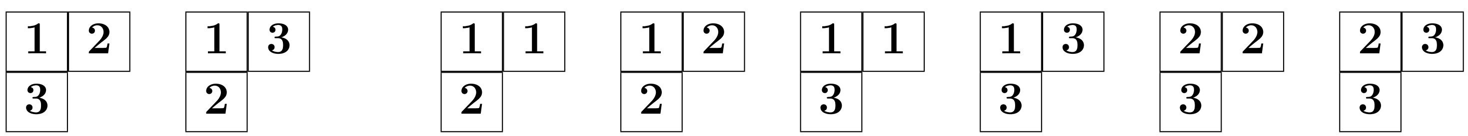
- **Example:**

$$\det\left(\begin{bmatrix} 1 & x_1^2 & x_1^4 \\ 1 & x_2^2 & x_2^4 \\ 1 & x_3^2 & x_3^4 \end{bmatrix}\right) = \det\left(\begin{bmatrix} 1 & x_1 & x_1^2 \\ 1 & x_2 & x_2^2 \\ 1 & x_3 & x_3^2 \end{bmatrix}\right) \cdot \underbrace{(2x_1x_2x_3 + x_1^2x_2 + x_1x_2^2 + x_1^2x_3 + x_1x_3^2 + x_2^2x_3 + x_2x_3^2)}_{s_\lambda}$$

- $\lambda = (2, 1)$**, Young diagram:**



- **Young tableaux:**

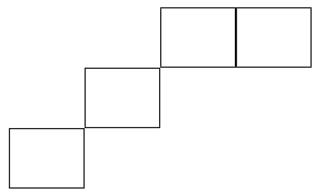


$$s_\lambda(x_1, x_2, x_3) = 2x_1x_2x_3 + x_1^2x_2 + x_1x_2^2 + x_1^2x_3 + x_1x_3^2 + x_2^2x_3 + x_2x_3^2$$

- **Exponentially many non-zero terms, need a more efficient approach**

The Schur Function $s_\lambda(x_1, x_2, \ldots, x_n)$

- **Lexicographic ordering of partitions**
- **If $\mu < \lambda$, then λ/μ – skew partition**
- **Young diagram $(4, 2, 1)/(2, 1)$:**



- **Macdonald:**

$$s_\lambda(x, y) = \sum_{\mu<\lambda} s_\mu(x) s_{\lambda/\mu}(y),$$

  x, y **- sets of indices**

- **Corollary:**

$$s_\lambda(x_1, x_2, ..., x_n) = \sum_{\mu} s_\mu(x_1, x_2, \ldots, x_{n-1}) x_n^{|\lambda|-|\mu|},$$

  summation over λ/μ - horizontal strip.

- **Example:**

$$s_{(2)}(x_1, \ldots, x_n) = \sum_{i \le j} x_i x_j = x_1 x_1 + (x_1 + x_2) x_2 + (x_1 + x_2 + x_3) x_3 + \ldots + (x_1 + \ldots + x_n) x_n$$

  Cost: $3n$, although $s_{(2)}$ has n^2 terms.

Computing The Schur Function

$$s_\lambda(x_1, x_2, ..., x_n) = \sum_\mu s_\mu(x_1, x_2, \ldots, x_{n-1}) x_n^{|\lambda|-|\mu|},$$

summation over λ/μ - horizontal strip.

- **Allows us to compute s_λ efficiently in many cases**
- **Exponentially faster than traditional $O(n^{|\lambda|})$ algorithm**

n/μ	(0)	(1)	$\ldots$	λ
1	1	x_1	$\ldots$	$s_\lambda(x_1)$
2	1	$x_1 + x_2$	$\ldots$	$s_\lambda(x_1, x_2)$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
n	1	$x_1 + x_2 + \ldots + x_n$	$\ldots$	$s_\lambda(x_1, x_2, \ldots, x_n)$

Table entries are $s_\mu(x_1, ..., x_i)$. Cost per entry:

$$F(\mu, i) = \#\{\nu < \mu | \mu/\nu \ - \ \text{horizonal strip}\}$$ $\ldots$**independent of n**

- **Total cost:**

$$G(\lambda, n) = \sum_{i=1}^{n} \sum_{\mu<\lambda} F(\mu, i)$$

- **Bound on total cost: If $\lambda_1 \geq \lambda_2 \geq \ldots \lambda_p \geq 0 = \lambda_{p+1} = \ldots = \lambda_n$ then**

$$G(\lambda, n) \leq (\lambda_1 + 1)(\lambda_2 + 1)^2 \ldots (\lambda_p + 1)^2 pn$$

Type of Matrix	$\det(A)$	A^{-1}	Any minor	GENP GEPP	GECP	SVD	NENP	$Ax = b$ Frwrd*	$Ax = b$ Bckwrd*
Cauchy	n^2	n^2	n^2	n^3	n^3	n^3	n^2	n^2	
TP Cauchy	n^2	n^2	n^2	n^3	n^3	n^3	n^2	n^2	n^2
Vandermonde	n^2					n^3	n^2	n^2	
TP Vandermonde	n^2	n^3	Exp	n^3	Exp	n^3	n^2	n^2	n^2
Polynomial Vandermonde Orth. Poly.	n^2			n^3		n^3			
Poly. Vand. Orth. poly.[1)] $0 < x_1 < \ldots < x_n$	n^2	n^3		n^3		n^3			
Generalized Vandermonde									
TP Generalized Vandermonde	$\Lambda n + n^2$	Λn^3	Exp	Λn^2	Exp	Exp	Λn^2	Λn^2	Λn^2

Big-O sense

***FORWARD BOUND:** $|x - \hat{x}| \leq O(\epsilon)|A^{-1}||b|$, **implying** $|x - \hat{x}| \leq O(\epsilon)|x|$ **for** x **checkerboard**
BACKWARD BOUND: $|A - \hat{A}| \leq O(\epsilon)|A|$, **where** $\hat{A}\hat{x} = b$.

1) **+ Other conditions on the signs of the three-term recurrence**

$\Lambda \leq (\lambda_1 + 1)(\lambda_2 + 1)^2 \ldots (\lambda_p + 1)^2 p$, **where** $\lambda = (\lambda_1, \ldots, \lambda_p)$.

Conclusions

- **TP Structured linear systems can be solved very accurately, if initial minors factor**
- **Implies accurate A^{-1}**
- **New application: Generalized Vandermonde Matrices**
- **New algorithm for computing the Schur function**
- **Not in this talk:**
  - **Accurate SVD of some Polynomial Vandermonde Matrices**
  - **Sometimes the SVD is easier than the inverse**

Open Problems

- **Totally Positive Matrices in general appear impossible. Proof?**
- **Characterize which structured matrices permit accurate and efficient linear algebra**

Resources

- **These slides:** `www.math.berkeley.edu/~plamen/mit02.pdf`
- **Reports:**

  - **J. Demmel and P. Koev, Necessary and sufficient conditions for accurate and efficient rational function evaluation and factorizations of rational matrices. In *Structured matrices in mathematics, computer science, and engineering. II (Boulder, CO, 1999)*, pages 117–143. Amer. Math. Soc., Providence, RI, 2001.**
  - **J. Demmel and P. Koev, Accurate and Efficient Matrix Computations with Totally Positive Generalized Vandermonde Matrices Using Schur Functions,** `www.math.berkeley.edu/~plamen/hagen.ps`
  - **J. Demmel, M. Gu, S. Eisenstat, I. Slapničar, K. Veselić, and Z. Drmač. Computing the singular value decomposition with high relative accuracy. *Lin. Alg. Appl.*, 299(1–3):21–80, 1999.** `www.cs.berkeley.edu/~demmel/DGESVD.ps`
  - **J. Demmel, Accurate SVDs for Structured Matrices, Accurate SVDs of structured matrices. *SIAM J. Mat. Anal. Appl.*, 21(2):562–580, 1999.** `www.netlib.org/lapack/lawns/lawn130.ps`