ACCURATE SVDs OF
– Weakly Diagonally Dominant M-Matrices
– Polynomial Vandermonde Matrices

Plamen Koev
Department of Mathematics
M.I.T.

Joint work with James Demmel

Supported by NSF and DOE

Parallel Matrix Algorithms and Applications
Neuchatel, Switzerland, 9-10 November 2002

Accurate SVDs

- Goal: Compute the SVD of a matrix A and get the "right" answer
- Only use working precision, algorithms (n^3)
- Traditional algorithms only guarantee backward accuracy and small absolute error

$$|\sigma_i - \hat{\sigma}_i| \leq |\sigma_1|$$

- We want forward accuracy and small relative error

$$|\sigma_i - \hat{\sigma}_i| \leq |\sigma_i|$$

$$\theta(w_i, \hat{w}_i) \leq O(\epsilon)/\min_{i \neq j} \frac{|\sigma_j - \sigma_i|}{\sigma_i}$$

- E.g. compute the tiniest singular values accurately
- Motivation
  - SVD determined accurately by the data
  - Right SVD costs the same as the wrong SVD – $O(n^3)$
- Example: 100-by-100 Hilbert matrix $H(i,j) = 1/(i+j)$

What floating point operations preserve forward accuracy?

- $\mathrm{fl}(a \otimes b) = (a \otimes b)(1+\delta)$, no under/overflow
- How can we lose accuracy in computing in floating point?
  - 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)
  - Cancellation when subtracting approximate results dangerous:

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

- So for positive x_i:

$$(x_1 + x_2 + \ldots + x_n)(x_2 - x_1)/x_3 \quad - \quad \text{accurate}$$

$$(x_1 + x_2) - (x_3 + x_4) \quad - \quad \text{may be inaccurate}$$

- We will compute everything using only allowable expressions

How do we compute accurate SVDs?

- **Thm (Demmel, Gu, Eisenstat, Slapničar, Veselić, Drmač):**

  Suffices to "reveal the rank":

$$A = XDY$$

  where

  – X, Y – **well conditioned will small norm error**

  – D – **computed with small absolute error componentwise**

- **Examples rank revealing decompositions:**
  - **LDU with complete pivoting**
  - **QR with pivoting**
- **Cannot always be done, but possible for structured matrices**
- **Two such new results**
  - **Diagonally dominant M-matrices**
  - **Polynomial Vandermonde matrices**

Diagonally Dominant M-matrices?

- Def: M-Matrix

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \dots & a_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \dots & a_{nn} \end{bmatrix};$$

$a_{ii} \geq 0$

$a_{ij} \leq 0, \; i \neq j$

Row Sums $s_i = \sum_{j=1}^{n} a_{ij} \geq 0$

- Given: Row sums s_i and off diagonals $a_{ij}, i \neq j$.
- Diagonal elements computable accurately, sum of positives

$$a_{ii} = s_i - \sum_{j \neq i} a_{ij}$$

GECP on Weakly Diagonally Dominant M-Matrices

- Pivoting is diagonal, preserves structure
- One step of GE:
  - Off diagonals: $a_{ij} = a_{ij} - \frac{a_{ik}a_{kj}}{a_{kk}}$
  - Row sums: $s_i = s_i - \frac{a_{ik}}{a_{kk}} s_k$
- Everything is preserved in Schur complementation
  - Weak diagonal dominance
  - M-matrix structure
  - High relative accuracy in a_{ij} and s_i
- Yields LDU from GECP thus SVD to high relative accuracy componentwise
- Small changes in $a_{ij}, i \neq j$ and s_i only cause small relative changes in the SVD
- No ill-conditioned set

SVDs of Polynomial Vandermondes – Definitions and Goals

- P – basis of polynomials, deg$P_i = i - 1$

$$V(x) = \begin{bmatrix} P_0(x_1) & P_1(x_1) & \dots & P_{n-1}(x_1) \\ P_0(x_2) & P_1(x_2) & \dots & P_{n-1}(x_2) \\ \vdots & \vdots & \ddots & \vdots \\ P_0(x_n) & P_1(x_n) & \dots & P_{n-1}(x_n) \end{bmatrix},$$

  is called ***Polynomial Vandermonde*** matrix.

- Of particular interest: orthonormal polynomials, monomials
- $V(x)$ – Notoriously ill-conditioned

Demmel 1998: Accurate SVD of a Quasi-Cauchy Matrix

- Cauchy: $C(i,j) = \frac{1}{x_i + y_j}$.
- Quasi-Cauchy: $F(x,y) = D'CD'' = \frac{d'_i d''_j}{x_i + y_j}$.
- $F(x,y) \cdot D$ still quasi-Cauchy for diagonal D
- Can compute accurate SVD of $F(x,y)$ (Demmel 1998)
- Accurate SVD of $V_P(x)$?
- Trick:

$$V_P(x) = F(x,y) \cdot V_P(y)$$

- Can select y such that $V_P(y) = D \cdot Q$ =diagonal×orthogonal
- Then

$$V_P(x) = (F(x,y) \cdot D)Q = U \cdot \Sigma \cdot Z^T \cdot Q$$

How do we choose y in $V(x) = F(x,y) \cdot V(y)$

- ***Discrete Orthogonality Property***:

$$\sum_{i=1}^{n} d_i P_r(y_i) P_s(y_i) = \delta_{rs}$$

  where

  - y – roots of P_n
  - d_i – Christoffel numbers

- $DV(y) = Q$ – orthogonal, where $D = \mathrm{diag}(\sqrt{d_i})_{i=1}^{n}$
- $V(x) = \underbrace{F(x,y) \cdot D^{-1}}_{\text{CAUCHY}} \cdot \underbrace{D \cdot V(y)}_{Q} = \underbrace{U \cdot \Sigma \cdot Z}_{\text{DEMMEL '98}} \cdot Q$

Computing y = roots of P_n, and d_i

- Not a part of the algorithm, needs to be done once for all $V(x)$
- Roots with small error – OK
- y_i – eigenvalues of a tridiagonal matrix
  (from the three term recurrence)
- $d_i = \det(A(1:i, 1:i))$ – also accurate
- Extremely well conditioned for Chebyshev, Legendre, Laguerre, Hermite
- For Chebyshev, 2nd kind:

$$A = \frac{1}{2}\begin{bmatrix} 0 & 1 & & & & \\ 1 & 0 & 1 & & & \\ & 1 & 0 & 1 & & \\ & & \ddots & \ddots & \ddots & \\ & & & 1 & 0 & 1 \\ & & & & 1 & 0 \end{bmatrix}$$

Any Polynomial Vandermonde

- OK, if we can find y such that

$$V(y) = \text{DIAGONAL} \cdot (\text{WELL CONDITIONED}).$$

- Perfect Example: Ordinary Vandermonde (Demmel 1998)

$$V(x) = \left[x_{i-1}^{j}\right]_{i,j=1}^{n}; \qquad V\left(\left[e^{\frac{2i\pi}{n}}\right]_{i=1}^{n}\right) = \text{DFT}$$

- Unnormalized orthogonal polynomials?
  OK, if the normalization factors don't vary too much

$$V(x) \cdot F = C(x,y)V(x) \cdot F = C(x,y) \cdot D \cdot Q \cdot F$$

  still polynomial complexity for Chebyshev, Laguerre, Hermite, Legendre.

Conclusions

- **New $O(n^3)$ accurate SVD algorihtms for**
  - **M-matrices**
  - **Polynomial Vandermonde matrices**
- **Open problem:**
  Confluent Polynomial Vandermonde, we think No.

- **These slides:** `www.math.berkeley.edu/~plamen/neuchatel.pdf`
- **My Ph.D. thesis,** `www.math.berkeley.edu/~plamen/a.ps`