Accurate Matrix Computations in the Presence of Roundoff Errors

---

Plamen Koev

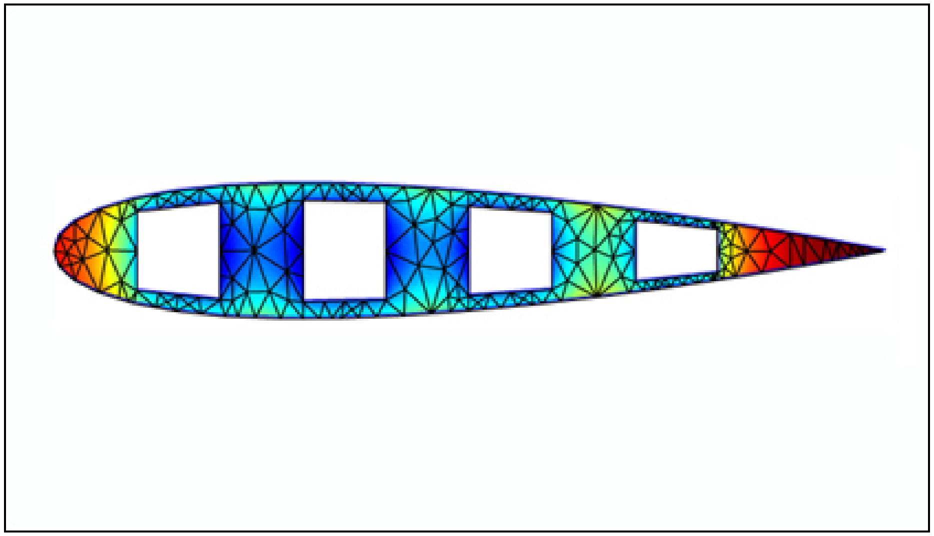
Massachusetts Institute of Technology

Joint work with James Demmel, Froilán Dopico

Universidad Carlos III de Madrid

January 12, 2005

Real-Life Problems often Modelled and Solved as Matrix Problems

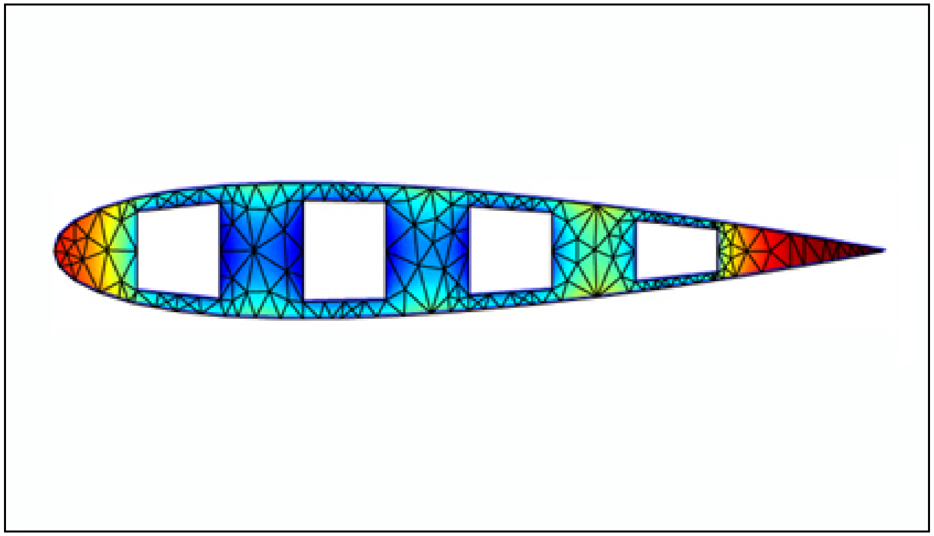


$$\Downarrow$$

$$\boldsymbol{Ax = b}, \quad \boldsymbol{Ax = \lambda x}$$

- **Solved in** ***finite precision arithmetic***

Real-Life Problems often Modelled and Solved as Matrix Problems



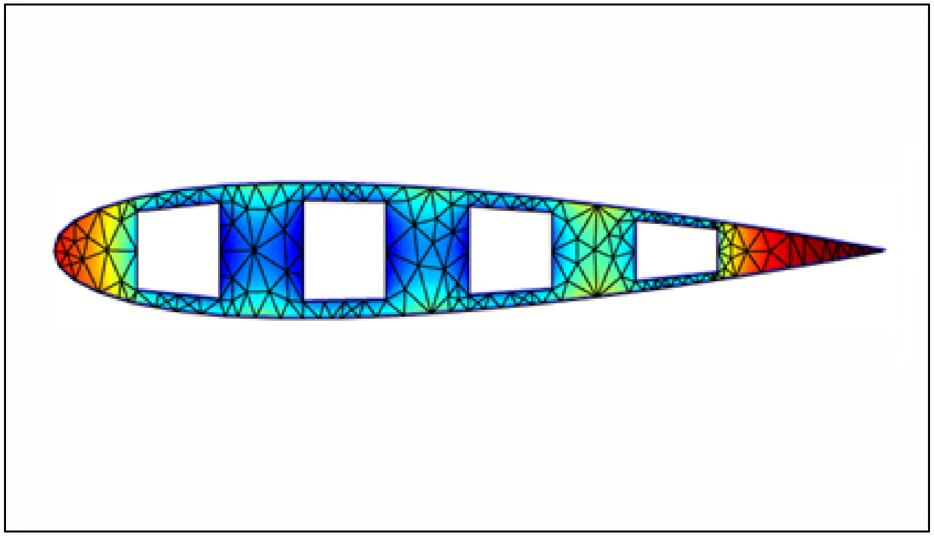
$$\Downarrow$$

$$Ax = b, \quad Ax = \lambda x$$

- **Solved in *finite precision arithmetic***
- **Roundoff Errors affect the accuracy:**

$$\det \begin{pmatrix} 1 & 3 \\ 3 & 9 \end{pmatrix} = 0, \quad \text{but} \quad \det \begin{pmatrix} 1 & 3 \\ 0.3 & 0.9 \end{pmatrix} = 1.1102\text{e} - 16$$

Real-Life Problems often Modelled and Solved as Matrix Problems



$$\Downarrow$$

$$\boldsymbol{Ax = b, \quad Ax = \lambda x}$$

- **Solved in** ***finite precision arithmetic***
- **Roundoff Errors affect the accuracy:**

$$\det\begin{pmatrix} 1 & 3 \\ 3 & 9 \end{pmatrix} = 0, \quad \text{but} \quad \det\begin{pmatrix} 1 & 3 \\ 0.3 & 0.9 \end{pmatrix} = 1.1102\text{e} - 16$$

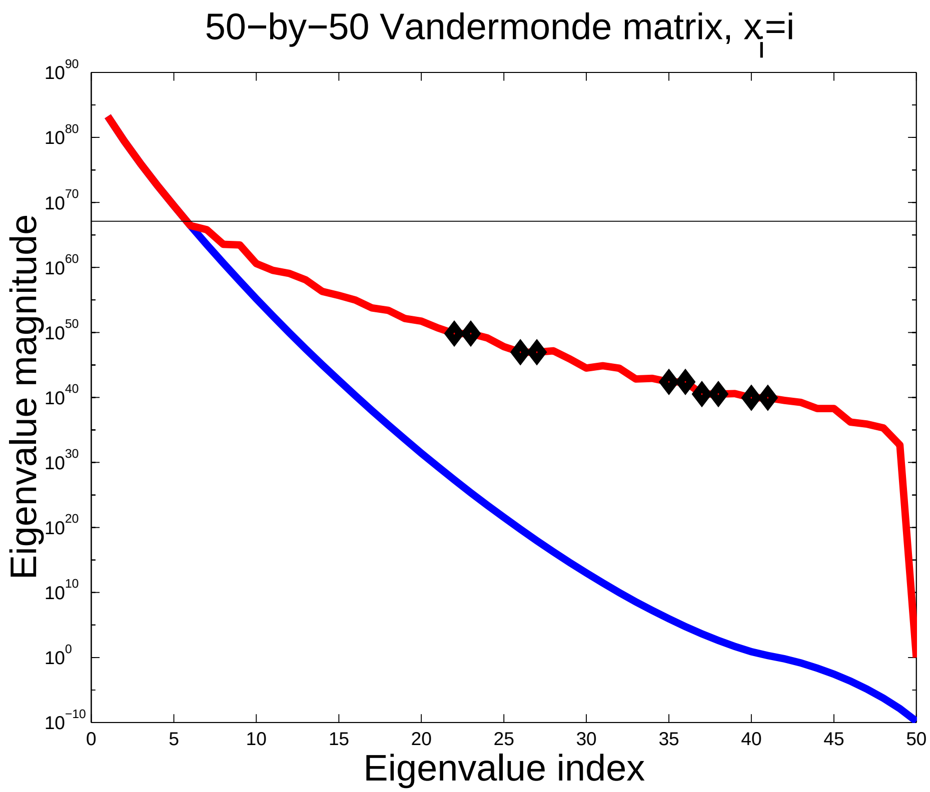
- **The quantity $\varepsilon = 1.1102\text{e}{-16} \approx 10^{-16}$ is called Machine Precision**

Example: 50×50 Vandermonde Matrix $\boldsymbol{V} = [i^{j-1}]_{i,j=1}^{50}$

$$\boldsymbol{V} = \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & 2 & 2^2 & \dots & 2^{49} \\ & \vdots & & & \\ 1 & 50 & 50^2 & & 50^{49} \end{bmatrix}$$

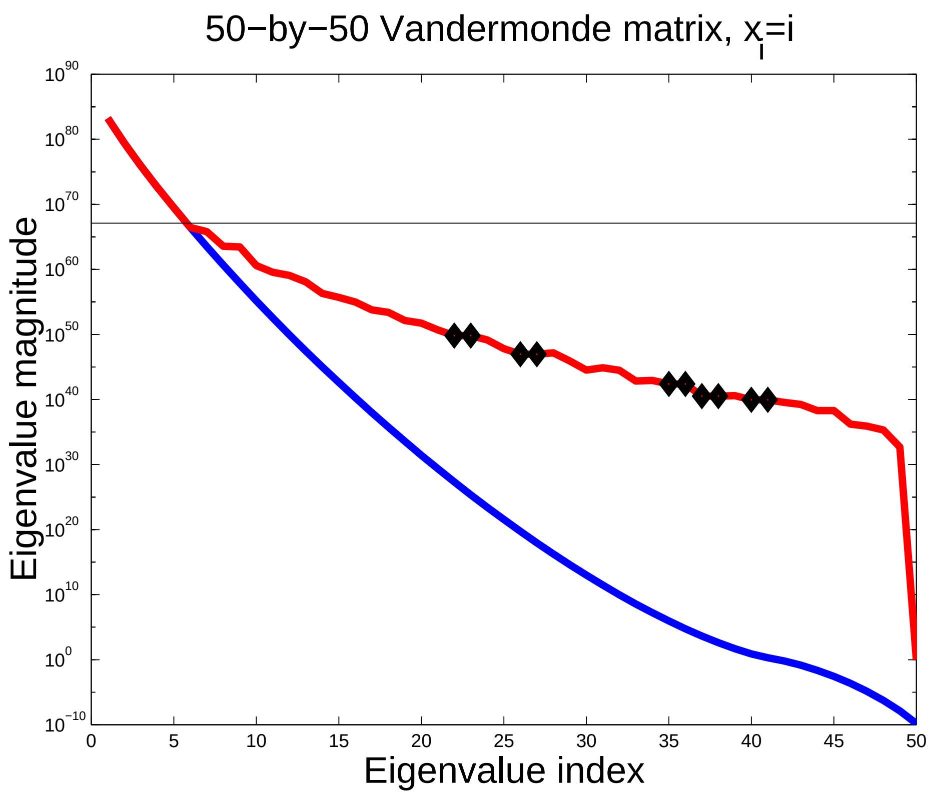
Example: 50 × 50 Vandermonde Matrix $V = [i^{j-1}]_{i,j=1}^{50}$

$$V = \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & 2 & 2^2 & \dots & 2^{49} \\ & \vdots & & & \\ 1 & 50 & 50^2 & & 50^{49} \end{bmatrix}$$



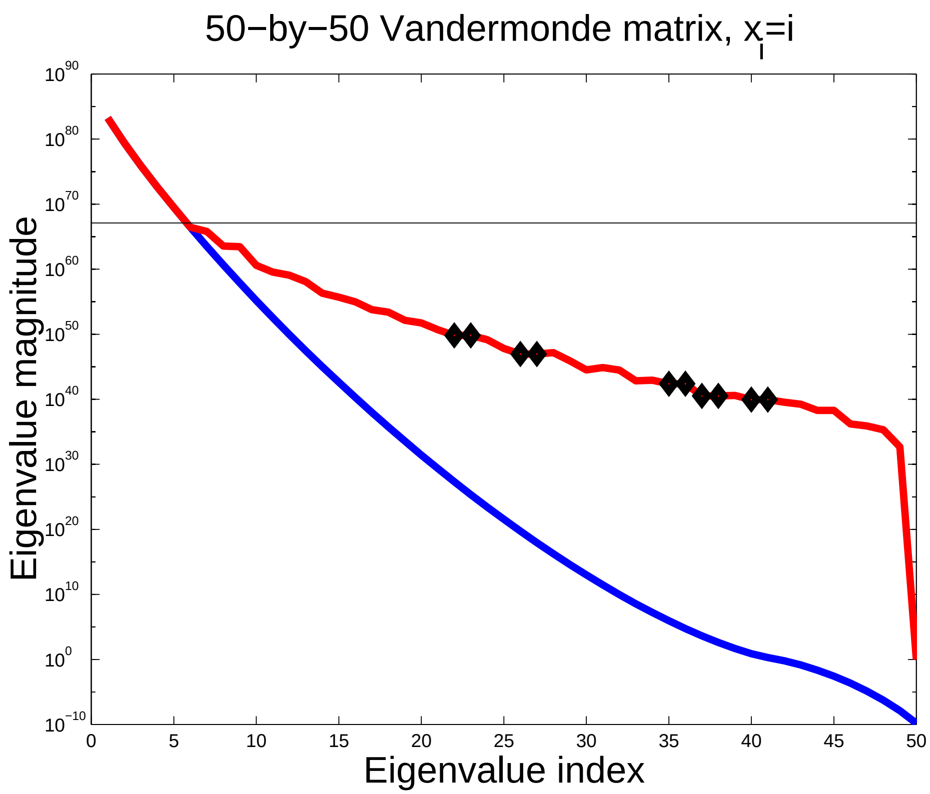
Example: 50 × 50 Vandermonde Matrix $V = [i^{j-1}]_{i,j=1}^{50}$

$$V = \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & 2 & 2^2 & \dots & 2^{49} \\ & \vdots & & & \\ 1 & 50 & 50^2 & & 50^{49} \end{bmatrix}$$



- **Blue=Accurate Algorithm**, **Red=Traditional Algorithm**

Example: 50 × 50 Vandermonde Matrix $V = [i^{j-1}]_{i,j=1}^{50}$



- **Blue=Accurate Algorithm**, **Red=Traditional Algorithm**
- **With traditional algorithm:**
  Only eigenvalues $\geq \|V\|\varepsilon \approx 10^{83} \cdot 10^{-16} = 10^{67}$ **are accurate**

What is the Problem?

- **We are computing "small" quantities (e.g., eigenvalues), when the norm of the matrix is "large" ($\|V\| \approx 10^{83}$)**
- **The quantity $\kappa(V) \equiv \|V\| \cdot \|V^{-1}\|$ is called *Condition number***
- **$\kappa(A) \geq 1$ for all A. Equality for A – orthogonal**
- **When $\kappa(A)$ is "small" (e.g., < 1000), A is *Well Conditioned***
- **Otherwise *Ill Conditioned***
- **A – well conditioned $\Rightarrow$ traditional algorithms Always Accurate**
- **A – ill conditioned $\Rightarrow$ likely inaccurate**

Problem in Computing with Ill Conditioned Matrices —Subtractive Cancellation

- **Subtractive Cancellation** means:

$$\text{“small”} = \text{“LARGE”} - \text{“LARGE”}$$

- if $\hat{a} \approx a > 0$ and $\hat{b} \approx b > 0$ to (say) 9 digits, then

$$\left.\begin{array}{rcl} \hat{a} \cdot \hat{b} & \approx & a \cdot b \\ \hat{a} + \hat{b} & \approx & a + b \\ \hat{a}/\hat{b} & \approx & a/b \end{array}\right\} \quad \text{to about 9 digits}$$

BUT

$$\hat{a} - \hat{b}$$

may have **no correct digits** if $a \approx b$:

$$\begin{array}{rr} & .123456789xxx \\ - & .123456789yyy \\ \hline & .000000000zzz \end{array}$$

- So: **NO SUBTRACTIONS** $\Longrightarrow$ **ACCURACY**!

Accurate Computations with Matrices

- **Not always possible:**
  - **if A is *unstructured*, then general theory applies**
- **If A is *structured*, accurate computations are possible when:**
  - **The structure is explicitly revealed**
  - **The quantities of interest are accurately determined**
  - **Subtraction-free computation is possible**

Examples of Structures Revealed Explicitly

- Vandermonde $V = [x_i^{j-1}]_{i,j=1}^n$, parameters: x_i
- Cauchy $V = \left[\frac{1}{x_i + y_j}\right]$, parameters: x_i, y_j
- M-matrices

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \dots & a_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \dots & a_{nn} \end{bmatrix}; \qquad \begin{array}{l} a_{ii} \geq 0 \\ a_{ij} \leq 0, \; i \neq j \\ \Sigma_{j=1}^n a_{ij} \geq 0 \end{array}$$

parameters: a_{ij} , $i \neq j$ and $s_i = \Sigma_{j=1}^n a_{ij}$ (but NOT a_{ii} !!!)

- Other classes of structured matrices also possible:
  - Generalized and Polynomial Vandermonde
  - Green's matrices
  - Oscillatory, Totally Positive, Sign-regular
  - ...

Gaussian Elimination of a Diagonally Dominant M-matrix

- **Traditional Gaussian elimination:**

$$a'_{ij} = a_{ij} - \frac{a_{ik}a_{kj}}{a_{kk}}, \quad i \neq j;$$

(NO subtractive cancellation!)

$$a'_{ii} = a_{ii} - \frac{a_{ik}a_{ki}}{a_{kk}}$$

(possible subtractive cancellation)

- **Schur complement is also a Diagonally Dominant M-matrix**
- **New Algorithm:**
  - Diagonal elements: $a_{kk} = s_k - \Sigma_{j \neq k}\, a_{kj}$
  - Off diagonals: $a'_{ij} = a_{ij} - \frac{a_{ik}a_{kj}}{a_{kk}}$
  - Row sums: $s'_i = s_i - \frac{a_{ik}}{a_{kk}} s_k$
- **Schur complement determined accurately by a'_{ij}, s'_i**
- **No subtractive cancellation $\Longrightarrow$ Accuracy!**

Conclusions

- **Accurate computations with structured matrices are possible when**
  - **The matrix structure is revealed explicitly**
  - **The computed quantities are well determined**
  - **Subtractive cancellation is avoided**
- **Works for many classes of structured matrices**
- **Many open problems remain**
- **This talk, MATLAB software and papers:**

  `math.mit.edu/~plamen`