CS 450: Numerical Anlaysis[1]

Linear Systems

University of Illinois at Urbana-Champaign

[1] *These slides have been drafted by Edgar Solomonik as lecture templates and supplementary material for the book "Scientific Computing: An Introductory Survey" by Michael T. Heath (slides).*

Vector Norms

Demo: Vector Norms

- **Properties of vector norms**
- **A norm is uniquely defined by its unit sphere:**
- p**-norms**

Inner-Product Spaces

- **Properties of inner-product spaces**: Inner products $\langle \boldsymbol{x}, \boldsymbol{y} \rangle$ must satisfy

$$
\begin{aligned}
\langle \boldsymbol{x}, \boldsymbol{x} \rangle &\geq 0 \\
\langle \boldsymbol{x}, \boldsymbol{x} \rangle &= 0 \quad \Leftrightarrow \quad \boldsymbol{x} = \boldsymbol{0} \\
\langle \boldsymbol{x}, \boldsymbol{y} \rangle &= \langle \boldsymbol{y}, \boldsymbol{x} \rangle \\
\langle \boldsymbol{x}, \boldsymbol{y} + \boldsymbol{z} \rangle &= \langle \boldsymbol{x}, \boldsymbol{y} \rangle + \langle \boldsymbol{x}, \boldsymbol{z} \rangle \\
\langle \alpha \boldsymbol{x}, \boldsymbol{y} \rangle &= \alpha \langle \boldsymbol{x}, \boldsymbol{y} \rangle
\end{aligned}
$$

- **Inner-product-based vector norms**

Matrix Norms

- **Properties of matrix norms:**

$$
\begin{aligned}
||\boldsymbol{A}|| &\geq 0 \\
||\boldsymbol{A}|| &= 0 \quad \Leftrightarrow \quad \boldsymbol{A} = \mathbf{0} \\
||\alpha \boldsymbol{A}|| &= |\alpha| \cdot ||\boldsymbol{A}|| \\
||\boldsymbol{A} + \boldsymbol{B}|| &\leq ||\boldsymbol{A}|| + ||\boldsymbol{B}|| \quad (\textit{triangle inequality})
\end{aligned}
$$

- **Frobenius norm:**

- **Operator/induced/subordinate matrix norms:**

Induced Matrix Norms

- **Interpreting induced matrix norms**:

- **General induced matrix norms**:

Matrix Condition Number

Demo: Conditioning of 2x2 Matrices
Demo: Condition number visualized

- **Definition**: $\kappa(\boldsymbol{A}) = ||\boldsymbol{A}|| \cdot ||\boldsymbol{A}^{-1}||$ is the ratio between the shortest/longest distances from the unit-ball center to any point on the surface.

- **Intuitive derivation**:

$$\kappa(\boldsymbol{A}) = \max_{\text{inputs}} \max_{\text{perturbations in input}} \left| \frac{\text{relative perturbation in output}}{\text{relative perturbation in input}} \right|$$

since a matrix is a linear operator, we can decouple its action on the input $\boldsymbol{x}$ and the perturbation $\boldsymbol{\delta x}$ since $\boldsymbol{A}(\boldsymbol{x} + \boldsymbol{\delta x}) = \boldsymbol{A x} + \boldsymbol{A \delta x}$, so

$$\kappa(\boldsymbol{A}) = \left| \frac{\overbrace{\max_{\text{perturbations in input}} \text{relative perturbation growth}}^{||\boldsymbol{A}||}}{\underbrace{\max_{\text{inputs}} \text{relative input reduction}}_{1/||\boldsymbol{A}^{-1}||}} \right|$$

Matrix Conditioning

- The matrix condition number $\kappa(\boldsymbol{A})$ is the ratio between the max and min distance from the surface to the center of the unit ball transformed by $\kappa(\boldsymbol{A})$:

- The matrix condition number bounds the worst-case amplification of error in a matrix-vector product:

Norms and Conditioning of Orthogonal Matrices

- **Orthogonal matrices**:

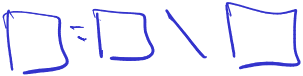
- **Norm and condition number of orthogonal matrices**:

Singular Value Decomposition

- **The singular value decomposition (SVD):**

$$\sigma_i \geq 0$$

$$A = USV^T \qquad S = \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_n \end{bmatrix}$$



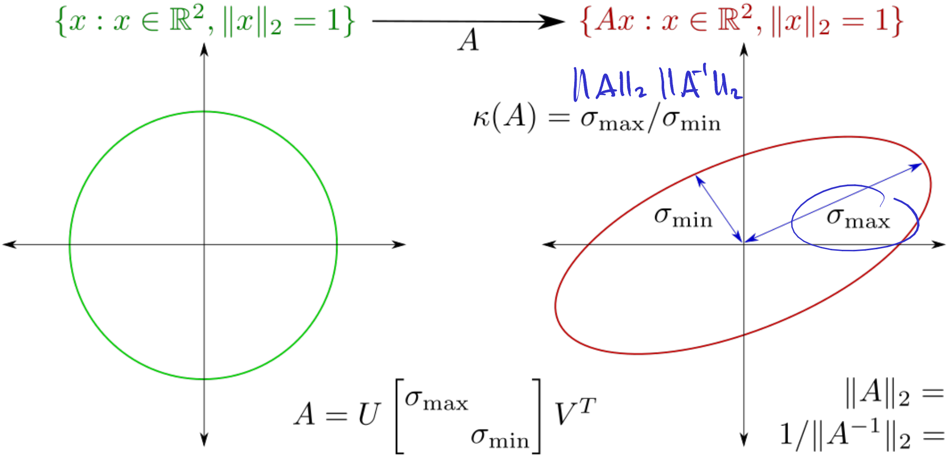
$$A = \begin{bmatrix} u^{(1)} & \cdots & u^{(n)} \end{bmatrix} \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_n \end{bmatrix} \begin{bmatrix} v^{(1)} & \cdots & v^{(n)} \end{bmatrix}^T$$

$$A = \sum_{i=1}^{n} \sigma_i u^{(i)} v^{(i)T}$$

Norms and Conditioning via SVD

- **Norm and condition number in terms of singular values**:

Visualization of Matrix Conditioning



$$\|A\|_2 \, \|A^{-1}\|_2$$

$$\kappa(A) = \sigma_{\max}/\sigma_{\min}$$

$$A = U \begin{bmatrix} \sigma_{\max} & \\ & \sigma_{\min} \end{bmatrix} V^T$$

$$\|A\|_2 = \sigma_{\max}$$

$$1/\|A^{-1}\|_2 = \sigma_{\min}$$

$$A^{-1} = V \begin{bmatrix} 1/\sigma_{\max} & \\ & 1/\sigma_{\min} \end{bmatrix} U^T$$

$$\|A\|_2 = \max_{\substack{\|x\|=1 \\ x \in \mathbb{R}^n}} \|Ax\|_2 = \sigma_{max}(A)$$

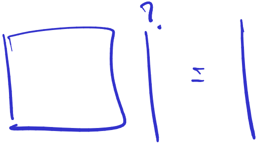
$$A = \sum_{i=1}^{n} \sigma_i u^{(i)} v^{(i)T}$$

$$A v^{(1)} = \sigma_1 u^{(1)} \underbrace{\left(v^{(1)T} v^{(1)}\right)}_{1}$$

$$= \sigma_1 u^{(1)}$$

Conditioning of Linear Systems

- **Lets now return to formally deriving the conditioning of solving $Ax = b$:**



$$\frac{\|\delta x\|}{\|x\|} \le \kappa(A) \frac{\|\delta b\|}{\|b\|}$$

$$b \to b + \delta b \implies x \to x + \delta x$$

$$\|Mx\|_2 \le \|M\|_2 \cdot \|x\|$$

$$\|AB\|_F \le \|A\| \cdot \|B\|$$

$$A(x + \delta x) = b + \delta b \implies A\,\delta x = \delta b$$

$$\delta x = A^{-1} \delta b \implies \|\delta x\| \le \|A^{-1}\| \|\delta b\|$$

$$\frac{\|\delta x\|}{\|x\|} \le \frac{\|A^{-1}\| \|\delta b\| \|A\|}{\|b\|}$$

$$\|A\| \|x\| \ge \|b\|$$

$$\|x\| \ge \|b\| / \|A\|$$

Conditioning of Linear Systems II

- **Consider perturbations to the input coefficients $\hat{A} = A + \delta A$:**

Solving Basic Linear Systems

$A \in \mathbb{R}^{n \times n}$

$O(n)$

- Solve $\boldsymbol{D}\boldsymbol{x} = \boldsymbol{b}$ if $\boldsymbol{D}$ is diagonal

$x = D^{-1} b \qquad x_i = b_i / d_i \qquad D^{-1} = \begin{bmatrix} 1/d_1 & & \\ & \ddots & \\ & & 1/d_n \end{bmatrix}$

- Solve $\boldsymbol{Q}\boldsymbol{x} = \boldsymbol{b}$ if $\boldsymbol{Q}$ is orthogonal

$Q^{-1} = Q^T \qquad x = Q^T b \qquad O(n^2) \qquad x_i = \sum_j q_{ij} b_j$

for i
  for j
    $x_i \leftarrow q_{ij} b_j$

- Given SVD $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$, solve $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$

$x = V S^{-1} U^T b \leftarrow U S V^T x = b$

$U y = b \Rightarrow y = U^T b$

$S z = y \Rightarrow y = S^{-1} z \quad | \quad V^T x = z \Rightarrow x = V z$

***Demo:** Coding back-substitution*

Solving Triangular Systems

- $Lx = b$ **if** L **is lower-triangular is solved by forward substitution:**

$$
\begin{aligned}
l_{11}x_1 &= b_1 & & x_1 = b_1/l_{11} \\
l_{21}x_1 + l_{22}x_2 &= b_2 & \Rightarrow \quad & x_2 = (b_2 - l_{21}x_1)/l_{22} \\
l_{31}x_1 + l_{32}x_2 + l_{33}x_3 &= b_3 & & x_3 = (b_3 - l_{31}x_1 - l_{32}x_2)/l_{33} \\
&\vdots & & \vdots
\end{aligned}
$$

$$
\begin{pmatrix} l_{11} & & \\ l_{21} & l_{22} & \\ l_{31} & \cdots & l_{33} \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ \vdots \\ b_n \end{pmatrix}
$$

- **Algorithm can also be formulated recursively by blocks:**

$$
\begin{bmatrix} L_{11} & \\ L_{21} & L_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}
$$

solve $L_{11}x_1 = b_1$ then using $L_{21}x_1 + L_{22}x_2 = b_2$

solve $L_{22}x_2 = b_2 - L_{21}x_1$

$T(n) = 2T(n/2) + O(n^2)$
$= O(n^2)$

Solving Triangular Systems

- **Existence of solution to $Lx = b$:**

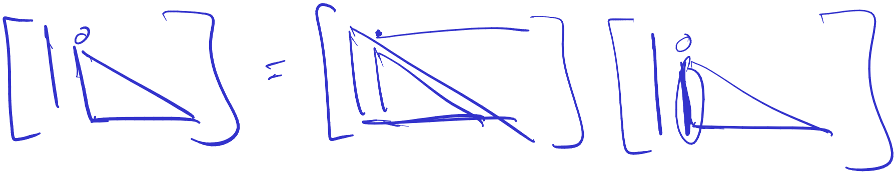
  if $l_{ii} = 0$, sol'n does not exist

- **Uniqueness of solution:**

- **Computational complexity of forward/backward substitution:**

Properties of Triangular Matrices

- $Z = XY$ is lower triangular is X and Y are both lower triangular:



- L^{-1} is lower triangular if it exists:

$L^{-1} = Y$

$$\begin{bmatrix} Y_{11} & \\ Y_{21} & Y_{22} \end{bmatrix} \begin{bmatrix} L_{11} & \\ L_{21} & L_{22} \end{bmatrix} = \begin{bmatrix} I & \\ & I \end{bmatrix}$$

$$Y_{11} = L_{11}^{-1}$$

$$Y_{21} L_{11} + Y_{22} L_{21} = 0$$

$$Y_{22} L_{22} = I$$

LU Factorization

- An ***LU factorization*** consists of a unit-diagonal lower-triangular ***factor*** L and upper-triangular factor U such that $A = LU$:

$$L = \begin{bmatrix} 1 & & & \\ l_{21} & 1 & & \\ l_{31} & l_{32} & 1 & \\ \vdots & & & \ddots \end{bmatrix} \qquad U = \begin{bmatrix} u_{11} & \cdots & \cdots \\ & \ddots & \vdots \\ & & u_{nn} \end{bmatrix}$$

- Given an LU factorization of A, we can solve the linear system $Ax = b$:

$$L U \overset{?}{x} = b$$

$$L \overset{?}{y} = b$$

$$U \overset{?}{x} = y$$

Gaussian Elimination Algorithm

- **Algorithm for factorization is derived from equations given by $A = LU$:**

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & A_{22} \end{bmatrix} = \begin{bmatrix} 1 & \\ l_{21} & L_{22} \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} \\ & U_{22} \end{bmatrix}$$

$$\begin{bmatrix} a_{11} & a_{12} \end{bmatrix} = \begin{bmatrix} 1 & \end{bmatrix} U \Rightarrow u_{11} = a_{11},\ u_{12} = a_{12}$$

$$\begin{bmatrix} a_{11} \\ a_{21} \end{bmatrix} = \begin{bmatrix} 1 & \\ l_{21} & L_{22} \end{bmatrix} \begin{bmatrix} u_{11} \\ \end{bmatrix} = \begin{bmatrix} u_{11} \\ l_{21} u_{11} \end{bmatrix} \Rightarrow \boxed{l_{21} = a_{21} / u_{11}}\ O(n)$$

- **The computational complexity of LU is $O(n^3)$:**

$$\begin{bmatrix} l_{21} & L_{22} \end{bmatrix} \begin{bmatrix} u_{12} \\ U_{22} \end{bmatrix} = A_{22}$$

$$n \cdot O(n^2) = O(n^3)$$

$$\sum_{i=1}^{n} 2(n-i)^2 \approx \frac{2n^3}{3}$$

$$A_{22} = L_{22} U_{22} + l_{21} u_{12}$$

Schur complement: $\boxed{A_{22} - l_{21} u_{12}} = L_{22} U_{22}$, $O(n^2)$

Existence of LU Factorization

- **The LU factorization may not exist:** Consider matrix $\begin{bmatrix} 3 & 2 \\ 6 & 4 \\ 0 & 3 \end{bmatrix}$.

- **Permutation of rows enables us to transform the matrix so the LU factorization does exist:**

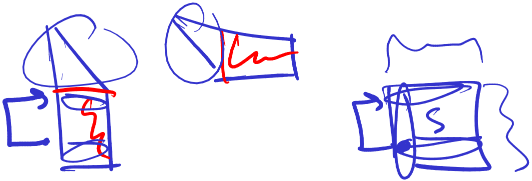
$$PA = LU \qquad A = P^T L U$$

$$w = P \cdot v \qquad w_i = v_{\pi(i)}$$

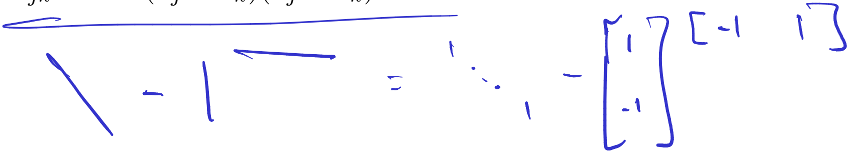
Demo: *LU with Partial Pivoting*

Gaussian Elimination with Partial Pivoting

- ***Partial pivoting*** **permutes rows to make divisor u_{ii} is maximal at each step:**



- **A row permutation corresponds to an application of a *row permutation matrix* $P_{jk} = I - (e_j - e_k)(e_j - e_k)^T$:**



Partial Pivoting Example

- Lets consider again the matrix $\boldsymbol{A} = \begin{bmatrix} 3 & 2 \\ 6 & 4 \\ 0 & 3 \end{bmatrix}$.

Complete Pivoting

- ***Complete pivoting*** **permutes rows and columns to make divisor u_{ii} is maximal at each step:**

- **Complete pivoting is noticeably more expensive than partial pivoting:**

Round-off Error in LU

- **Lets consider factorization of $\begin{bmatrix} \epsilon & 1 \\ 1 & 1 \end{bmatrix}$ where $\epsilon < \epsilon_{\text{mach}}$:**

- **Permuting the rows of A in partial pivoting gives $PA = \begin{bmatrix} 1 & 1 \\ \epsilon & 1 \end{bmatrix}$**

Error Analysis of LU

- **The main source of round-off error in LU is in the computation of the Schur complement:**

- **When computed in floating point, absolute backward error δA in LU (so $\hat{\boldsymbol{L}}\hat{\boldsymbol{U}} = \boldsymbol{A} + \boldsymbol{\delta A}$) is $|\delta a_{ij}| \leq \epsilon_{\text{mach}}(|\hat{\boldsymbol{L}}| \cdot |\hat{\boldsymbol{U}}|)_{ij}$**

Helpful Matrix Properties

- **Matrix is *diagonally dominant*, so $\sum_{i \neq j} |a_{ij}| \leq |a_{ii}|$:**

- **Matrix is *symmetric positive definite (SPD)*, so $\forall_{\boldsymbol{x} \neq 0}, \boldsymbol{x}^T \boldsymbol{A} \boldsymbol{x} > 0$:**

- **Matrix is symmetric but indefinite:**

- **Matrix is *banded*, $a_{ij} = 0$ if $|i - j| > b$:**

Solving Many Linear Systems

- **Suppose we have computed $A = LU$ and want to solve $AX = B$ where B is $n \times k$ with $k < n$:**

- **Suppose we have computed $A = LU$ and now want to solve a perturbed system $(\boldsymbol{A} - \boldsymbol{u}\boldsymbol{v}^T)\boldsymbol{x} = \boldsymbol{b}$:**
  Can use the *Sherman-Morrison-Woodbury* formula

$$(\boldsymbol{A} - \boldsymbol{u}\boldsymbol{v}^T)^{-1} = \boldsymbol{A}^{-1} + \frac{\boldsymbol{A}^{-1}\boldsymbol{u}\boldsymbol{v}^T\boldsymbol{A}^{-1}}{1 - \boldsymbol{v}^T\boldsymbol{A}^{-1}\boldsymbol{u}}$$