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

- **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:**

$$||\boldsymbol{A}|| \geq 0$$
$$||\boldsymbol{A}|| = 0 \quad \Leftrightarrow \quad \boldsymbol{A} = \boldsymbol{0}$$
$$||\alpha \boldsymbol{A}|| = |\alpha| \cdot ||\boldsymbol{A}||$$
$$||\boldsymbol{A} + \boldsymbol{B}|| \leq ||\boldsymbol{A}|| + ||\boldsymbol{B}|| \quad \text{(triangle inequality)}$$

- **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 $\boldsymbol{A}(\boldsymbol{x} + \boldsymbol{\delta x}) = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{A}\boldsymbol{\delta x}$, 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**:

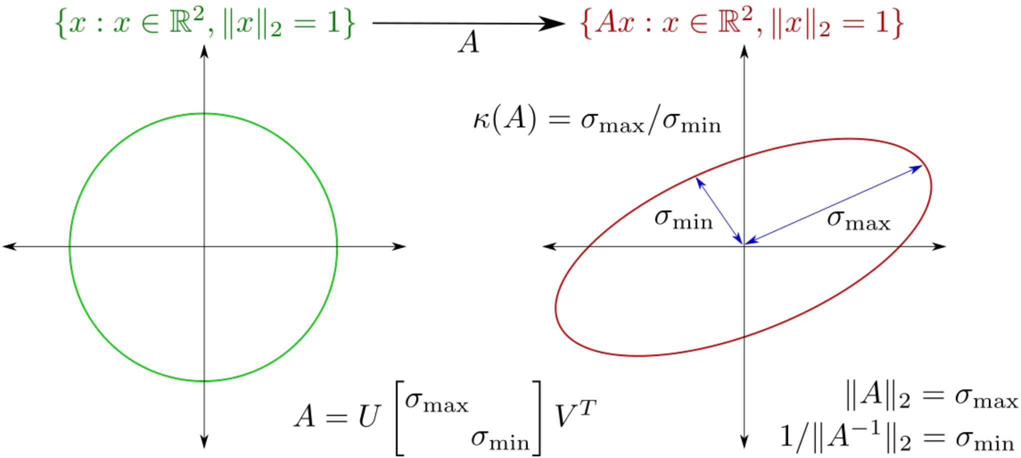
Singular Value Decomposition

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

Norms and Conditioning via SVD

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

Visualization of Matrix Conditioning



Conditioning of Linear Systems

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

Conditioning of Linear Systems II

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

Solving Basic Linear Systems

- Solve $\boldsymbol{D}\boldsymbol{x} = \boldsymbol{b}$ if $\boldsymbol{D}$ is diagonal
- Solve $\boldsymbol{Q}\boldsymbol{x} = \boldsymbol{b}$ if $\boldsymbol{Q}$ is orthogonal
- Given SVD $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$, solve $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$

Solving Triangular Systems

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

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

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

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

Solving Triangular Systems

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

- **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:

LU Factorization

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

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

Gaussian Elimination Algorithm

Demo: *LU factorization*

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

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

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:**

Gaussian Elimination with Partial Pivoting

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

- **A row permutation corresponds to an application of a *row permutation matrix* $\boldsymbol{P}_{jk} = \boldsymbol{I} - (\boldsymbol{e}_j - \boldsymbol{e}_k)(\boldsymbol{e}_j - \boldsymbol{e}_k)^T$:**

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 $\hat{L}\hat{U} = A + \delta A$) is** $|\delta a_{ij}| \leq \epsilon_{\text{mach}}(|\hat{L}| \cdot |\hat{U}|)_{ij}$

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

Demo: Sherman-Morrison
Activity: Sherman-Morrison-Woodbury Formula

- **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

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