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**

$$\begin{aligned} ||\boldsymbol{x}|| &\geq 0 \\ ||\boldsymbol{x}|| &= 0 \quad \Leftrightarrow \quad \boldsymbol{x} = \boldsymbol{0} \\ ||\alpha \boldsymbol{x}|| &= |\alpha| \cdot ||\boldsymbol{x}|| \\ ||\boldsymbol{x} + \boldsymbol{y}|| &\leq ||\boldsymbol{x}|| + ||\boldsymbol{y}|| \end{aligned}$$

(triangle inequality) implies continuity

- **A norm is uniquely defined by its unit sphere:** *Surface defined by space of vectors* $\mathbb{V} \subset \mathbb{R}^n$ *such that* $\forall \boldsymbol{x} \in \mathbb{V}, ||\boldsymbol{x}|| = 1$
- p**-norms** $||\boldsymbol{x}||_p = \left(\sum_i |x_i|^p \right)^{1/p}$
  - $p = 1$ *gives sum of absolute values of entry (unit sphere is diamond-like)*
  - $p = \infty$ *gives maximum entry in absolute value (unit sphere is box-like)*
  - $p = 2$ *gives Euclidean distance metric (unit sphere is spherical)*

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**

  The $p = 2$ vector norm is the Eucledian inner-product norm,

$$||\boldsymbol{x}||_2 = \sqrt{\boldsymbol{x}^T \boldsymbol{x}}$$

  and due to Cauchy-Schwartz inequality $|\langle \boldsymbol{x}, \boldsymbol{y} \rangle| \leq \sqrt{\langle \boldsymbol{x}, \boldsymbol{x} \rangle \cdot \langle \boldsymbol{y}, \boldsymbol{y} \rangle}$*,*

$$|\boldsymbol{x}^T \boldsymbol{y}| \leq ||\boldsymbol{x}||_2 ||\boldsymbol{y}||_2.$$

  Other inner-products can be expressed as $\langle \boldsymbol{x}, \boldsymbol{y} \rangle = \boldsymbol{x}^T \boldsymbol{A} \boldsymbol{y}$ *where* $\boldsymbol{A}$ *is symmetric positive definite, yielding norms* $||\boldsymbol{x}||_{\boldsymbol{A}} = \sqrt{\boldsymbol{x}^T \boldsymbol{A} \boldsymbol{x}}$

Matrix Norms

- **Properties of matrix norms**:

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

- **Frobenius norm**:

$$
||\mathbf{A}||_F = \left(\sum_{i,j} a_{ij}^2 \right)^{1/2}
$$

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

  For any vector norm $||\cdot||$, *the induced matrix norm is*

$$
||\boldsymbol{A}|| = \max_{\boldsymbol{x} \neq \boldsymbol{0}} ||\boldsymbol{A}\boldsymbol{x}|| / ||\boldsymbol{x}|| = \max_{||\boldsymbol{x}||=1} ||\boldsymbol{A}\boldsymbol{x}||
$$

Induced Matrix Norms

- **Interpreting induced matrix norms**: *A matrix is uniquely defined with respect to a norm by a unit-ball, which is the space of vectors $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$ for all $\boldsymbol{x}$ on the unit-sphere of the norm.*

$$||\boldsymbol{A}||_p = \max_{||\boldsymbol{x}||_p=1} ||\boldsymbol{A}\boldsymbol{x}||_p$$

  is the maximum possible p-norm amplification due to application of $\boldsymbol{A}$

$$1/||\boldsymbol{A}^{-1}||_p = \min_{||\boldsymbol{x}||_p=1} ||\boldsymbol{A}\boldsymbol{x}||_p$$

  is the maximum possible p-norm reduction due to application of $\boldsymbol{A}$

- **General induced matrix norms**:

$$||\boldsymbol{A}||_{mp} = \max_{||\boldsymbol{x}||_p=1} ||\boldsymbol{A}\boldsymbol{x}||_m$$

  typically $m = p$ so we write $||\boldsymbol{A}||_p$ and almost always we have $p \in \{1, 2, \infty\}$.
  (Computing the matrix norm for certain choices of $m \neq p$ is NP-complete.)

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 max distance to center is given by the vector maximizing* $\max_{||\boldsymbol{x}||=1} ||\boldsymbol{A}\boldsymbol{x}||_2$.
  - *The min distance to center is given by the vector minimizing* $\min_{||\boldsymbol{x}||=1} ||\boldsymbol{A}\boldsymbol{x}||_2 = 1/(\max_{||\boldsymbol{x}||=1} ||\boldsymbol{A}^{-1}\boldsymbol{x}||_2)$.
  - *Thus, we have that* $\kappa(\boldsymbol{A}) = ||\boldsymbol{A}||_2||\boldsymbol{A}^{-1}||_2$
- The matrix condition number bounds the worst-case amplification of error in a matrix-vector product: *Consider* $\boldsymbol{y} + \boldsymbol{\delta y} = \boldsymbol{A}(\boldsymbol{x} + \boldsymbol{\delta x})$, *assume* $||\boldsymbol{x}||_2 = 1$
  - *In the worst case,* $||\boldsymbol{y}||_2$ *is minimized, that is* $||\boldsymbol{y}||_2 = 1/||\boldsymbol{A}^{-1}||_2$
  - *In the worst case,* $||\boldsymbol{\delta y}||_2$ *is maximized, that is* $||\boldsymbol{\delta y}||_2 = ||\boldsymbol{A}||_2||\boldsymbol{\delta y}||_2$
  - *So* $||\boldsymbol{\delta y}||_2/||\boldsymbol{y}||_2$ *is at most* $\kappa(\boldsymbol{A})||\boldsymbol{\delta x}||_2/||\boldsymbol{x}||_2$

Norms and Conditioning of Orthogonal Matrices

- **Orthogonal matrices**: *A matrix $\boldsymbol{Q}$ is orthogonal, if its square and its columns are orthonormal, or equivalently $\boldsymbol{Q}^T = \boldsymbol{Q}^{-1}$.*
- **Norm and condition number of orthogonal matrices**: *For any $||v||_2 = 1$,*

$$\begin{aligned} ||\boldsymbol{Q}\boldsymbol{v}||_2 &= \left(\left\langle \boldsymbol{v}^T\boldsymbol{Q}^T, \boldsymbol{Q}\boldsymbol{v}\right\rangle\right)^{1/2} = \left(\boldsymbol{v}^T\boldsymbol{Q}^T\boldsymbol{Q}\boldsymbol{v}\right)^{1/2} = \left(\boldsymbol{v}^T\boldsymbol{v}\right)^{1/2} \\ &= ||\boldsymbol{v}||_2 \end{aligned}$$

Consequently, $||\boldsymbol{Q}||_2 = ||\boldsymbol{Q}^{-1}||_2 = \kappa(\boldsymbol{Q}) = 1$.
$\boldsymbol{Q}\boldsymbol{v}$ expresses $\boldsymbol{v}$ in a coordinate system whose axes are columns of $\boldsymbol{Q}^T$

Singular Value Decomposition

- **The singular value decomposition (SVD):**
  We can express any matrix $\boldsymbol{A}$ as

$$\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$$

  where $\boldsymbol{U}$ and $\boldsymbol{V}$ are orthogonal, and $\boldsymbol{\Sigma}$ is square nonnegative and diagonal,

$$\boldsymbol{\Sigma} = \begin{bmatrix} \sigma_{max} & & \\ & \ddots & \\ & & \sigma_{min} \end{bmatrix}$$

  Any matrix is diagonal when expressed as an operator mapping vectors from a coordinate system given by $\boldsymbol{V}$ to a coordinate system given by $\boldsymbol{U}^T$.

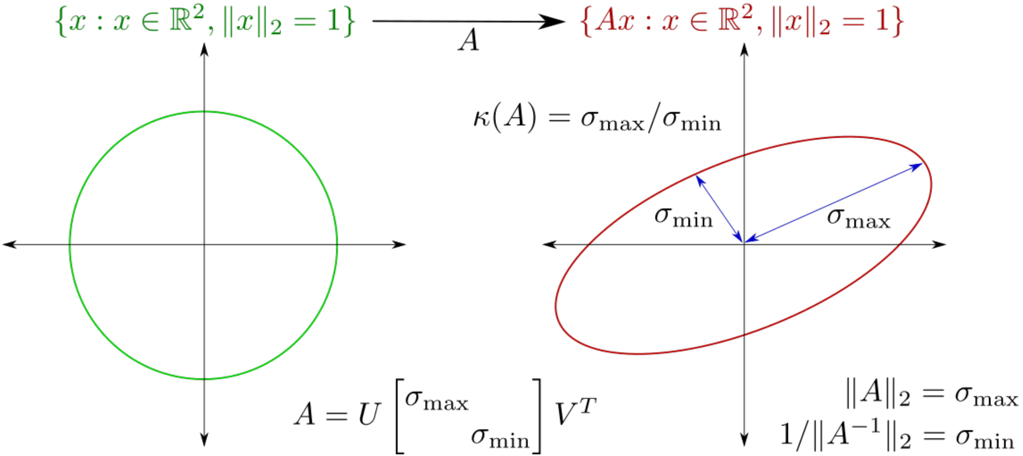
Norms and Conditioning via SVD

- **Norm and condition number in terms of singular values**:
  When multiplying a vector by matrix $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$
  - *Multiplication by* $\boldsymbol{V}^T$ *changes coordinate systems, leaving the norm unchanged*
  - *Multiplication by* $\boldsymbol{U}$ *changes coordinate systems, leaving the norm unchanged*

  so, only multiplication by $\boldsymbol{\Sigma}$ *has an effect on the vector norm*
  - *Note that* $||\boldsymbol{\Sigma}||_2 = \sigma_{max}$, $||\boldsymbol{\Sigma}^{-1}||_2 = 1/\sigma_{min}$, *so*

$$\kappa(\boldsymbol{A}) = \kappa(\boldsymbol{\Sigma}) = \frac{\sigma_{max}}{\sigma_{min}}$$

Visualization of Matrix Conditioning



Conditioning of Linear Systems

- **Lets now return to formally deriving the conditioning of solving $Ax = b$:**
  Consider a perturbation to the right-hand side (input) $\hat{\boldsymbol{b}} = \boldsymbol{b} + \boldsymbol{\delta b}$

$$\boldsymbol{A}\hat{\boldsymbol{x}} = \hat{\boldsymbol{b}}$$
$$\boldsymbol{A}(\boldsymbol{x} + \boldsymbol{\delta x}) = \boldsymbol{b} + \boldsymbol{\delta b}$$
$$\boldsymbol{A}\boldsymbol{\delta x} = \boldsymbol{\delta b}$$

  we wish to bound the size of the relative perturbation to the output $||\boldsymbol{\delta x}||/||\boldsymbol{x}||$ *with respect to the size of the relative perturbation the the input* $||\boldsymbol{\delta b}||/||\boldsymbol{b}||$

$$\boldsymbol{\delta x} = \boldsymbol{A}^{-1}\boldsymbol{\delta b}$$
$$\frac{||\boldsymbol{\delta x}||}{||\boldsymbol{x}||} = \frac{||\boldsymbol{A}^{-1}\boldsymbol{\delta b}||}{||\boldsymbol{x}||} \leq \frac{||\boldsymbol{A}^{-1}|| \cdot ||\boldsymbol{\delta b}||}{||\boldsymbol{x}||}$$

  we can use that $||\boldsymbol{x}|| \geq ||\boldsymbol{b}||/\sigma_{max} = ||\boldsymbol{b}||/||\boldsymbol{A}||$ *so*

$$\frac{||\boldsymbol{\delta x}||}{||\boldsymbol{x}||} \leq \underbrace{||\boldsymbol{A}|| \cdot ||\boldsymbol{A}^{-1}||}_{\kappa(\boldsymbol{A})} \cdot \frac{||\boldsymbol{\delta b}||}{||\boldsymbol{b}||} = \frac{\sigma_{max}||\boldsymbol{\delta b}||}{\sigma_{min}||\boldsymbol{b}||}$$

Conditioning of Linear Systems II

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

  In this case we solve the perturbed system

$$
\begin{aligned}
\hat{\boldsymbol{A}}\hat{\boldsymbol{x}} &= \boldsymbol{b} \\
\boldsymbol{A}\boldsymbol{x} + \boldsymbol{\delta A}\boldsymbol{x} &= \boldsymbol{b} - \hat{\boldsymbol{A}}\boldsymbol{\delta x} \\
\boldsymbol{\delta A}\boldsymbol{x} &= -\hat{\boldsymbol{A}}\boldsymbol{\delta x} \approx -\boldsymbol{A}\boldsymbol{\delta x}
\end{aligned}
$$

  we wish to bound the size of the relative perturbation to the output $||\boldsymbol{\delta x}||/||\boldsymbol{x}||$ *with respect to the size of the relative perturbation the the input* $||\boldsymbol{\delta A}||/||\boldsymbol{A}||$

$$
\begin{aligned}
\boldsymbol{\delta x} &= -\boldsymbol{A}^{-1}\boldsymbol{\delta A}\boldsymbol{x} \\
||\boldsymbol{\delta x}|| &= ||\boldsymbol{A}^{-1}\boldsymbol{\delta A}\boldsymbol{x}|| \le ||\boldsymbol{A}^{-1}|| \cdot ||\boldsymbol{\delta A}|| \cdot ||\boldsymbol{x}|| \\
\frac{||\boldsymbol{\delta x}||}{||\boldsymbol{x}||} &\le \underbrace{||\boldsymbol{A}^{-1}|| \cdot ||\boldsymbol{A}||}_{\kappa(\boldsymbol{A})} \cdot \frac{||\boldsymbol{\delta A}||}{||\boldsymbol{A}||}
\end{aligned}
$$

Solving Basic Linear Systems

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

  $x_i = b_i/d_{ii}$ *with total cost* $O(n)$

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

  $\boldsymbol{x} = \boldsymbol{Q}^T\boldsymbol{b}$ *with total cost* $O(n^2)$

- Given SVD $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$, solve $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$
  - *Compute* $\boldsymbol{z} = \boldsymbol{U}^T\boldsymbol{b}$
  - *Solve* $\boldsymbol{\Sigma}\boldsymbol{y} = \boldsymbol{z}$ *(diagonal)*
  - *Compute* $\boldsymbol{x} = \boldsymbol{V}\boldsymbol{x}$

Solving Triangular Systems

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

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

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

$$\begin{bmatrix} l_{11} & \\ \boldsymbol{l}_{21} & \boldsymbol{L}_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ \boldsymbol{x_2} \end{bmatrix} = \begin{bmatrix} b_1 \\ \boldsymbol{b_2} \end{bmatrix}$$

$x_1 = b_1/l_{11}$, *then solve recursively for* $\boldsymbol{x_2}$ *in* $\boldsymbol{L_{22}x_2} = \boldsymbol{b_2} - \boldsymbol{l_{21}}x_1$.

Solving Triangular Systems

- **Existence of solution to $\boldsymbol{Lx} = \boldsymbol{b}$:**
  If some $l_{ii} = 0$, the solution may not exist, and $\boldsymbol{L}^{-1}$ does not exist.
- **Uniqueness of solution:** *Even if some $l_{ii} = 0$ and $\boldsymbol{L}^{-1}$ does not exist, the system may have a solution. The solution will not be unique since columns of $\boldsymbol{L}$ are necessarily linearly dependent if a diagonal element is zero. May want to select solution minimizing norm of $\boldsymbol{x}$.*
- **Computational complexity of forward/backward substitution:**
  The recursive algorithm has the cost recurrence,

$$T(n) = T(n-1) + n = \sum_{i=1}^{n} i = n(n+1)/2.$$

The total cost is $n^2/2$ multiplications and $n^2/2$ additions to leading order.

Properties of Triangular Matrices

- **$\boldsymbol{Z} = \boldsymbol{X}\boldsymbol{Y}$ is lower triangular is $\boldsymbol{X}$ and $\boldsymbol{Y}$ are both lower triangular:**

$$\begin{bmatrix} z_{11} & \boldsymbol{z}_{12} \\ \boldsymbol{z}_{21} & \boldsymbol{Z}_{22} \end{bmatrix} = \begin{bmatrix} x_{11} & \\ \boldsymbol{x}_{21} & \boldsymbol{X}_{22} \end{bmatrix} \begin{bmatrix} y_{11} & \\ \boldsymbol{y}_{21} & \boldsymbol{Y}_{22} \end{bmatrix}.$$

  Clearly, $z_{11} = x_{11}y_{11}$ and $\boldsymbol{z}_{12} = 0$, then we proceed by the same argument for the triangular matrix product $\boldsymbol{Z}_{22} = \boldsymbol{X}_{22}\boldsymbol{Y}_{22}$.

- **$\boldsymbol{L}^{-1}$ is lower triangular if it exists:**

  We give a constructive proof by providing an algorithm for triangular matrix inversion. We need $\boldsymbol{Y} = \boldsymbol{X}^{-1}$ so

$$\begin{bmatrix} \boldsymbol{Y}_{11} & \\ \boldsymbol{Y}_{21} & \boldsymbol{Y}_{22} \end{bmatrix} \begin{bmatrix} \boldsymbol{X}_{11} & \\ \boldsymbol{X}_{21} & \boldsymbol{X}_{22} \end{bmatrix} = \begin{bmatrix} \boldsymbol{I} & \\ & \boldsymbol{I} \end{bmatrix},$$

  from which we can deduce

$$\boldsymbol{Y}_{11} = \boldsymbol{X}_{11}^{-1}, \quad \boldsymbol{Y}_{22} = \boldsymbol{X}_{22}^{-1}, \quad \boldsymbol{Y}_{21} = -\boldsymbol{Y}_{22}\boldsymbol{X}_{21}\boldsymbol{Y}_{11}.$$

LU Factorization

- **An *LU factorization* consists of a unit-diagonal lower-triangular *factor* L and upper-triangular factor U such that $A = LU$:**
  - *Unit-diagonal implies each $l_{ii} = 1$, leaving $n(n-1)/2$ unknowns in $\boldsymbol{L}$ and $n(n+1)/2$ unknowns in $\boldsymbol{U}$, for a total of n^2, the same as the size of $\boldsymbol{A}$.*
  - *For rectangular matrices $\boldsymbol{A} \in \mathbb{R}^{m \times n}$, one can consider a full LU factorization, with $\boldsymbol{L} \in \mathbb{R}^{m \times \max(m,n)}$ and $\boldsymbol{U} \in \mathbb{R}^{\max(m,n) \times n}$, but it is fully described by a reduced LU factorization, with lower-trapezoidal $\boldsymbol{L} \in \mathbb{R}^{m \times \min(m,n)}$ and upper-trapezoidal $\boldsymbol{U} \in \mathbb{R}^{\min(m,n) \times n}$.*
- **Given an LU factorization of A, we can solve the linear system $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$:**
  - *using forward substitution $\boldsymbol{L}\boldsymbol{y} = \boldsymbol{b}$*
  - *using backward substitution to solve $\boldsymbol{U}\boldsymbol{x} = \boldsymbol{y}$*

Backward substitution is the same as forward substitution with a reversal of the ordering of the elements of the vectors and the ordering of the rows/columns of the matrix.

Demo: *LU factorization*

Gaussian Elimination Algorithm

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

$$\begin{bmatrix} a_{11} & \boldsymbol{a}_{12} \\ \boldsymbol{a}_{21} & \boldsymbol{A}_{22} \end{bmatrix} = \begin{bmatrix} 1 & \\ \boldsymbol{l}_{21} & \boldsymbol{L}_{22} \end{bmatrix} \begin{bmatrix} u_{11} & \boldsymbol{u}_{12} \\ & \boldsymbol{U}_{22} \end{bmatrix} = \begin{bmatrix} \boldsymbol{L}_{11} & \\ \boldsymbol{L}_{21} & \boldsymbol{L}_{22} \end{bmatrix} \begin{bmatrix} \boldsymbol{U}_{11} & \boldsymbol{U}_{12} \\ & \boldsymbol{U}_{22} \end{bmatrix}$$

  - *First, observe* $\begin{bmatrix} u_{11} & \boldsymbol{u}_{12} \end{bmatrix} = \begin{bmatrix} a_{11} & \boldsymbol{a}_{12} \end{bmatrix}$
  - *To obtain* $\boldsymbol{l}_{21}$ *compute* $\boldsymbol{l}_{21} = \boldsymbol{a}_{21}/u_{11}$
  - *Obtain* $\boldsymbol{L}_{22}$ *and* $\boldsymbol{U}_{22}$ *by recursively computing LU of the Schur complement*

$$\boldsymbol{S} = \boldsymbol{A}_{22} - \boldsymbol{l}_{21}\boldsymbol{u}_{12}$$

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

  Computing $\boldsymbol{l}_{21} = \boldsymbol{a}_{21}/u_{11}$ *requires* $O(n)$ *operations, finding* $\boldsymbol{S}$ *requires* $2n^2$*, so to leading order the complexity of LU is*

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

Existence of LU Factorization

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

  Proceeding with Gaussian elimination we obtain

$$\begin{bmatrix} 3 & 2 \\ 6 & 4 \\ 0 & 3 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 2 & 1 \\ 0 & l_{32} \end{bmatrix} \begin{bmatrix} 3 & 2 \\ 0 & u_{21} \end{bmatrix}.$$

  Then we need that $4 = 4 + u_{21}$ *so* $u_{21} = 0$, *but at the same time* $l_{32}u_{21} = 3$.
  More generally, if and only if for any partitioning $\begin{bmatrix} \boldsymbol{A}_{11} & \boldsymbol{A}_{12} \\ \boldsymbol{A}_{21} & \boldsymbol{A}_{22} \end{bmatrix}$ *the leading minor is singular* ($\det(\boldsymbol{A}_{11}) = 0$), $\boldsymbol{A}$ *has no LU factorization.*
- **Permutation of rows enables us to transform the matrix so the LU factorization does exist:**

  Gaussian elimination can only fail if dividing by zero. At every recursive step of Gaussian elimination, if the leading entry of the first row is zero, we permute it with a row with an leading nonzero (if $\boldsymbol{a}_{21} = \boldsymbol{0}$, *we set* $u_{11} = 0$ *and* $\boldsymbol{l}_{21} = \boldsymbol{0}$*).*

Gaussian Elimination with Partial Pivoting

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

  Based on our argument above, for any matrix $\boldsymbol{A}$ there exists a permutation matrix $\boldsymbol{P}$ that can permute the rows of $\boldsymbol{A}$ to permit an LU factorization,

  $$\boldsymbol{P}\boldsymbol{A} = \boldsymbol{L}\boldsymbol{U}.$$

  Partial pivoting finds such a permutation matrix $\boldsymbol{P}$ one row at a time. The ith row is selected to maximize the magnitude of the leading element (over elements in the first column), which becomes the entry u_{ii}. This selection ensures that we are never forced to divide by zero during Gaussian elimination and that the magnitude of any element in $\boldsymbol{L}$ is at most 1.

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

  If we permute row i_j .o be the leading (ith) row at the ith step, the overall permutation matrix is given by $\boldsymbol{P}^T = \prod_{i=1}^{n-1} \boldsymbol{P}_{ii_j}$.

Partial Pivoting Example

- Lets consider again the matrix $\boldsymbol{A} = \begin{bmatrix} 3 & 2 \\ 6 & 4 \\ 0 & 3 \end{bmatrix}$.
  - *The largest magnitude element in the first column is* 6, *so we select this as our pivot and perform the first step of LU*

$$\underbrace{\begin{bmatrix} & 1 & \\ 1 & & \\ & & 1 \end{bmatrix}}_{\boldsymbol{P}_1} \begin{bmatrix} 6 & 4 \\ 3 & 2 \\ 0 & 3 \end{bmatrix} = \begin{bmatrix} 1 \\ 1/2 \\ 0 \end{bmatrix} \begin{bmatrix} 6 & 4 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 2 - (1/2) \cdot 4 \\ 0 & 3 - 0 \cdot 4 \end{bmatrix}$$

  - *The Schur complement is* $\begin{bmatrix} 0 & 3 \end{bmatrix}^T$ *and we proceed with pivoted LU,*

$$\underbrace{\begin{bmatrix} & 1 \\ 1 & \end{bmatrix}}_{\boldsymbol{P}_2} \begin{bmatrix} 0 \\ 3 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \begin{bmatrix} 3 \end{bmatrix}$$

  - *The overall LU factorization is then given by* $\boldsymbol{P}_1 \begin{bmatrix} 1 & \\ & \boldsymbol{P}_2 \end{bmatrix} \boldsymbol{A} = \begin{bmatrix} 1 & \\ 0 & 1 \\ 1/2 & 0 \end{bmatrix} \begin{bmatrix} 6 & 4 \\ & 3 \end{bmatrix}$

Complete Pivoting

- ***Complete pivoting*** **permutes rows and columns to make divisor u_{ii} is maximal at each step:**
  - *Partial pivoting ensures that the magnitude of the multipliers satisfies* $|\boldsymbol{l}_{21}| = |\boldsymbol{a}_{21}|/|u_{11}| \leq \mathbf{1}$
  - *Complete pivoting also gives* $||\boldsymbol{u}_{12}||_\infty \leq |u_{11}|$ *and consequently* $|\boldsymbol{l}_{21}| \cdot ||\boldsymbol{u}_{12}||_\infty = |\boldsymbol{a}_{21}| \cdot ||\boldsymbol{u}_{12}||_\infty/|u_{11}| \leq |\boldsymbol{a}_{21}|$
  - *Complete pivoting yields a factorization of the form* $\boldsymbol{LU} = \boldsymbol{PAQ}$ *where* $\boldsymbol{P}$ *and* $\boldsymbol{Q}$ *are permutation matrices*
- **Complete pivoting is noticeably more expensive than partial pivoting:**
  - *Partial pivoting requires just* $O(n)$ *comparison operations and a row permutation*
  - *Complete pivoting requires* $O(n^2)$ *comparison operations, which somewhat increases the leading order cost of LU overall*

Round-off Error in LU

- **Lets consider factorization of $\begin{bmatrix} \epsilon & 1 \\ 1 & 1 \end{bmatrix}$ where $\epsilon < \epsilon_{\text{mach}}$:**
  - *Without pivoting we would compute* $\boldsymbol{L} = \begin{bmatrix} 1 & 0 \\ 1/\epsilon & 1 \end{bmatrix}$, $\boldsymbol{U} = \begin{bmatrix} \epsilon & 1 \\ 0 & 1 - 1/\epsilon \end{bmatrix}$
  - *Rounding yields* $fl(\boldsymbol{U}) = \begin{bmatrix} \epsilon & 1 \\ 0 & -1/\epsilon \end{bmatrix}$
  - *This leads to* $\boldsymbol{L} fl(\boldsymbol{U}) = \begin{bmatrix} \epsilon & 1 \\ 1 & 0 \end{bmatrix}$, *a backward error of* $\begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$
- **Permuting the rows of A in partial pivoting gives $\boldsymbol{P}\boldsymbol{A} = \begin{bmatrix} 1 & 1 \\ \epsilon & 1 \end{bmatrix}$**
  - *We now compute* $\boldsymbol{L} = \begin{bmatrix} 1 & 0 \\ \epsilon & 1 \end{bmatrix}$, $\boldsymbol{U} = \begin{bmatrix} 1 & 1 \\ 0 & 1 - \epsilon \end{bmatrix}$, *so* $fl(\boldsymbol{U}) = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$
  - *This leads to* $\boldsymbol{L} fl(\boldsymbol{U}) = \begin{bmatrix} 1 & 1 \\ \epsilon & 1 + \epsilon \end{bmatrix}$, *a backward error of* $\begin{bmatrix} 0 & 0 \\ 0 & \epsilon \end{bmatrix}$

Error Analysis of LU

- **The main source of round-off error in LU is in the computation of the Schur complement:**
  - *Recall that division is well-conditioned, while addition can be ill-conditioned*
  - *After k steps of LU, we are working on Schur complement $\boldsymbol{A}_{22} - \boldsymbol{L}_{21}\boldsymbol{U}_{12}$ where $\boldsymbol{A}_{22}$ is $(n-k)\times(n-k)$, $\boldsymbol{L}_{21}$ and $\boldsymbol{U}_{12}^T$ are $(n-k)\times k$*
  - *Partial pivoting and complete pivoting improve stability by making sure $\boldsymbol{L}_{21}\boldsymbol{U}_{12}$ is small in norm*
- **When computed in floating point, absolute backward error δA in LU (so $\hat{\boldsymbol{L}}\hat{\boldsymbol{U}} = \boldsymbol{A} + \boldsymbol{\delta A}$) is $|\delta a_{ij}| \leq \epsilon_{\text{mach}}(|\hat{\boldsymbol{L}}| \cdot |\hat{\boldsymbol{U}}|)_{ij}$**

  For any a_{ij} with $j \geq i$ (lower-triangle is similar), we compute

$$a_{ij} - \sum_{k=1}^{i} \hat{l}_{ik}\hat{u}_{kj} = a_{ij} - \langle \hat{\boldsymbol{l}}_i, \hat{\boldsymbol{u}}_j \rangle,$$

  which in floating point incurs round-off error at most $\epsilon_{\mathit{mach}}\langle |\hat{\boldsymbol{l}}_i|, |\hat{\boldsymbol{u}}_j| \rangle$. Using this, for complete pivoting, we can show $|\delta a_{ij}| \leq \epsilon_{\mathit{mach}} n^2 ||\boldsymbol{A}||_\infty$.

Helpful Matrix Properties

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

  Pivoting is not required if matrix is strictly diagonally dominant $\sum_{i \neq j} |a_{ij}| < |a_{ii}|$.

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

  $\boldsymbol{L} = \boldsymbol{U}$ *and pivoting is not required, Cholesky algorithm* $\boldsymbol{A} = \boldsymbol{L}\boldsymbol{L}^T$ *can be used (*$\boldsymbol{L}$ *in Cholesky is not unit-diagonal).*

- **Matrix is symmetric but indefinite**:

  Compute pivoted LDL factorization $\boldsymbol{P}\boldsymbol{A}\boldsymbol{P}^T = \boldsymbol{L}\boldsymbol{D}\boldsymbol{L}^T$ *(where* $\boldsymbol{L}$ *is lower-triangular and unit-diagonal, while* $\boldsymbol{D}$ *is diagonal)*

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

  LU without pivoting and Cholesky preserve banded structure and require only $O(nb^2)$ *work.*

Solving Many Linear Systems

- **Suppose we have computed $A = LU$ and want to solve $AX = B$ where B is $n \times k$ with $k < n$:**
  Cost is $O(n^2k)$ for solving the k independent linear systems
- **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

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

  - *Consequently we have* $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} + \frac{\boldsymbol{u}\boldsymbol{v}^T\boldsymbol{A}^{-1}}{1-\boldsymbol{v}^T\boldsymbol{A}^{-1}\boldsymbol{u}}\boldsymbol{b} = \boldsymbol{b} + \frac{\boldsymbol{v}^T\boldsymbol{A}^{-1}\boldsymbol{b}}{1-\boldsymbol{v}^T\boldsymbol{A}^{-1}\boldsymbol{u}}\boldsymbol{u}$
  - *Need not form $\boldsymbol{A}^{-1}$ or $\boldsymbol{L}^{-1}$ or $\boldsymbol{U}^{-1}$, suffices to use backward/forward substitution to solve $\boldsymbol{w}^T\boldsymbol{A} = \boldsymbol{v}^T$, i.e. solve $\boldsymbol{U}^T\boldsymbol{L}^T\boldsymbol{w} = \boldsymbol{v}$ and then solve*

$$\boldsymbol{L}\boldsymbol{U}\boldsymbol{x} = \boldsymbol{b} + \underbrace{\left(\frac{\boldsymbol{w}^T\boldsymbol{b}}{1 - \boldsymbol{w}^T\boldsymbol{u}}\right)}_{\textit{scalar}}\boldsymbol{u}$$