Lecture 4: Matrix Decomposition Mathods for $A\,\underline{x} = \underline{b}$

Prof. Krishna R. Pattipati

Dept. of Electrical and Computer Engineering
University of Connecticut
Contact: krishna@engr.uconn.edu (860) 486-2890

ECE 6435
Adv Numerical Methods in Sci Comp

Fall 2008
September 17, 2008

Outline of Lecture 4

- Why do we need to solve $A\underline{x} = \underline{b}$?
- Concepts of **forward elimination** and **backward substitution**
- Basic decomposition methods: *LU*, *QR*, Cholesky, SVD
- *LU* decomposition
- Sensitivity of the solution to $A\underline{x} = \underline{b}$
  - Error and residual
  - **Condition number** as an amplification factor for error
- Iterative improvement
- Estimation of condition number
- Solution when A is modified by a rank-one correction matrix

Background

- Key problem

  - Solution of $A\underline{x} = \underline{b} \Rightarrow \underline{b} = \sum_{i=1}^{n} x_i \underline{a}_i$

This is one of the most important problems in NUMERICAL ANALYSIS

  - Fact1: a solution exists only if $\underline{b}$ is a linear combination of the columns of A (EXISTENCE CONDITION)
    $\Rightarrow \underline{b} \in R(A)$
  - Fact 2: for an n x n matrix A, $A\underline{x} = \underline{b}$ has a unique solution if and only if $N(A)$ is null $\Rightarrow A\underline{x} = 0$ has the **only** solution $\underline{x} = 0$ (UNIQUENESS CONDITION)
    $\Rightarrow \text{Rank}(A) = n \Rightarrow \dim(R(A)) = n \Rightarrow A$ is invertible

- Restricted problem:

  Assume A is n x n, Rank(A) = $n \Rightarrow A$ is nonsingular
  We want to solve $A\underline{x} = \underline{b}$

Why Solve $A\,\underline{x} = \underline{b}$? - 1

- Why do we need to solve $A\underline{x} = \underline{b}$?

1) Data fitting via linear equations (occurs in a wide variety of applications including nonlinear programming, interpolation, regression, etc.)
   - Suppose, we want to fit an n^{th} order polynomial to the data

     $$\{x_i, f(x_i) : i = 0, 1, 2, \ldots, n\}$$

   - That is , want

$$f(x) = a_0 + a_1 x + a_2 x^2 + \ldots + a_n x^n$$

Why Solve $A\,\underline{x} = \underline{b}$? - 2

then, the problem of finding $\{a_i: i = 0, 1, 2, \ldots, n\}$ is equivalent to solving:

$$\begin{bmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^n \\ 1 & x_1 & x_1^2 & \cdots & x_1^n \\ 1 & x_n & x_n^2 & \cdots & x_n^n \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_n \end{bmatrix} = \begin{bmatrix} f(x_0) \\ f(x_1) \\ f(x_n) \end{bmatrix}$$

transpose of Van der Monde matrix

2) s.s solution to $\dot{\underline{x}} = A\underline{x} + B\underline{u}; \quad \Rightarrow \quad A\underline{x} = \underline{b}; \;\; \underline{b} = -B\underline{u}$

3) Solution of nonlinear equations via Newton's method

- $\underline{g}(x^*) \approx g(\underline{x}_k) + \nabla g^T(\underline{x}_k)(\underline{x}^* - \underline{x}_k) + \ldots$
- approximate $\underline{x}_{k+1}$ such that the following first order condition is satisfied:

$$\underline{g}(x^*) \approx g(\underline{x}_k) + J(\underline{x}_k)(\underline{x}_{k+1} - \underline{x}_k) = 0; \;\; \text{where } J(\underline{x}_k) = \nabla \underline{g}^T(\underline{x}_k)$$

$$\Rightarrow \;\; J(\underline{x}_k)(\underline{x}_{k+1} - \underline{x}_k) = -\underline{g}(\underline{x}_k) \Rightarrow \underline{x}_{k+1} = \underline{x}_k - [J(\underline{x}_k)]^{-1}\underline{g}(\underline{x}_k)$$

Why Solve $A\underline{x} = \underline{b}$? - 3

4) Minimization of a scalar function w.r.t. n variables $x_1, \ldots, x_n$

- Approximate $f(\underline{x})$ by a quadratic function around the current minimum:

$$f(\underline{x}_{k+1}) \approx f(\underline{x}_k) + \nabla \underline{f}^T(\underline{x}_k)(\underline{x}_{k+1} - \underline{x}_k)$$
$$+ (\underline{x}_{k+1} - \underline{x}_k)^T \nabla^2 \underline{f}(\underline{x}_k)(\underline{x}_{k+1} - \underline{x}_k)/2$$

- Want $\underline{x}_{k+1}$ to be the optimum of Quadratic function

$$\Rightarrow \nabla f(\underline{x}_{k+1}) = 0$$
$$\Rightarrow \nabla^2 \underline{f}(\underline{x}_k)(\underline{x}_{k+1} - \underline{x}_k) = -\nabla \underline{f}(\underline{x}_k)$$

5) In computing e^{At}, $\int e^{At}$ via Pade approximation, we come across solutions of a sequence of linear equations

$$A\underline{x}_i = \underline{b}_i, \quad i = 1, 2, \ldots$$

Exploit Key Facts - 1

- Method of Attack: Break it up into simpler subproblems
  $\Rightarrow$ Decomposition

- FACT1: DIAGONAL & TRIANGULAR SYSTEMS OF EQUATIONS ARE EASIER TO SOLVE
  - Lower triangular system of equations can be solved via **Forward Elimination**

$$\begin{bmatrix} l_{11} & 0 & \cdots & 0 \\ l_{21} & l_{22} & \cdots & 0 \\ l_{n1} & l_{n2} & \cdots & l_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_n \end{bmatrix} \Rightarrow \begin{array}{c} x_1 = b_1 / l_{11} \\ x_2 = (b_2 - l_{21}x_1) / l_{22}, \text{ etc} \\ \cdot \\ x_n = (b_n - \sum_{j=1}^{n-1} l_{nj} x_j) / l_{nn} \end{array}$$

  - FORWARD ELIMINATION requires O($n^2/2$) operations
  - Similarly, upper triangular system of equations can be solved via **backward substitution**

Exploit Key Facts - 2

$$\begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1n} \\ 0 & u_{22} & \cdots & u_{2n} \\ 0 & \cdots & \cdots & u_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_n \end{bmatrix} \Rightarrow \begin{array}{l} x_n = b_n / u_{nn} \\ x_{n-1} = (b_{n-1} - u_{n-1,n} x_n) / u_{n-1,n-1}, \text{ etc} \\ \cdot \\ x_1 = (b_1 - \sum_{j=2}^{n} u_{1j} x_j) / u_{11} \end{array}$$

– BACKWARD SUBSTITUTION requires O($n^2/2$) operations

- FACT2: ORTHOGONAL MATRICES ARE EASY TO INVERT. STABLE NUMERICALLY $\Rightarrow$ DO NOT "SCREW UP" THE ORIGINAL PROBLEM.

$$\Rightarrow \quad Q^{-1} = Q^T; \quad \|QAZ\|_F = \|A\|_F; \quad \|Q\underline{x}\|_2 = \|x\|_2, \text{ etc.}$$

Decomposition Methods - 1

- DECOMPOSITION METHODS BASED ON FACT1

  A is an $n \times n$ matrix

1) LU Decomposition (Doolittle decomposition) …. Lecture 4
   - Writes matrix $A=LU$ or $PA=LU$
   - P is a permutation matrix (permutes rows and columns)
   - Can also write it as : $PA=LDU$
   - Solution of $PA\underline{x} = P\underline{b} \Rightarrow LU\underline{x} = P\underline{b} \Rightarrow L\underline{y} = P\underline{b};\ U\underline{x} = \underline{y}$

   $\Rightarrow$
   - Once have L & U, can solve $A\underline{x}_i=\underline{b}_i$, $i \geq 1$ in $O(n^2)$ operations
   - One of the most widely used methods for solving linear systems

2) If $A=A^T$ and PD … Lecture 5

   $A = LL^T$ or $A = \bar{L}D\bar{L}^T$ (Cholesky decomposition)

   - One of the best methods for testing if A is a PD matrix.

Decomposition Methods - 2

DECOMPOSITION METHODS BASED ON FACTS 1 & 2

- Useful for general A, e.g., Least Squares Estimation… see Lectures 6-8

1) $A = QR; Q$ orthogonal $\Rightarrow Q^{-1} = Q^T$, R upper triangular…….Lectures 6-7

$$A\underline{x} = \underline{b} \Rightarrow QR\underline{x} = \underline{b} \text{ or } R\underline{x} = Q^T\underline{b} = \tilde{b}$$

$\Rightarrow$ upper triangular system of equations $\Rightarrow$ backward substitution

2) Singular Value Decomposition (SVD)…Lecture12

$A = U\Sigma V^T$; U, V are orthogonal

$$\Rightarrow U\Sigma V^T\underline{x} = \underline{b} \Rightarrow \Sigma V^T\underline{x} = U^T\underline{b};\ \Sigma\underline{y} = U^T\underline{b};\ \underline{y} = V^T\underline{x} \Rightarrow \underline{x} = V\underline{y}$$

LU Decomposition

LU Decomposition

- Belongs to the class of **direct methods**
- $A = LU \Rightarrow$ want to determine n^2+n entries from n^2 entries
  $\Rightarrow$ Can fix either L = unit lower Δ or U= unit upper Δ

$$L = \begin{bmatrix} 1 & 0 & .. & 0 \\ l_{21} & 1 & .. & 0 \\ \vdots & \vdots & & \vdots \\ l_{n1} & l_{n2} & .. & 1 \end{bmatrix} \qquad U = \begin{bmatrix} u_{11} & u_{12} & .. & u_{1n} \\ 0 & u_{22} & .. & u_{2n} \\ \vdots & \vdots & & \vdots \\ 0 & 0 & .. & u_{nn} \end{bmatrix}$$

LU Decomposition as Dyadic Sum

Example: $\begin{bmatrix} 1 & 1 \\ 2 & 7 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 2 & 1 \end{bmatrix}\begin{bmatrix} 1 & 1 \\ 0 & 5 \end{bmatrix}$

$\quad A \qquad L \qquad U$

$$A = LU \Rightarrow \begin{bmatrix} 1 & 0 & .. & 0 \\ l_{21} & 1 & .. & 0 \\ \vdots & \vdots & & \vdots \\ l_{n1} & l_{n2} & .. & 1 \end{bmatrix}\begin{bmatrix} u_{11} & u_{12} & .. & u_{1n} \\ 0 & u_{22} & .. & u_{2n} \\ \vdots & \vdots & & \vdots \\ 0 & 0 & .. & u_{nn} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & .. & a_{1n} \\ a_{21} & a_{22} & .. & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & .. & a_{nn} \end{bmatrix}$$

$$\Rightarrow A = \sum_{k=1}^{n} \underline{l}_k \underline{u}_k^T ; \; \underline{l}_k = \begin{bmatrix} 0 \\ . \\ 1 \\ l_{k+1,k} \\ . \\ l_{n,k} \end{bmatrix}; \; \underline{u}_k^T = \begin{bmatrix} 0 & .. & u_{kk} & u_{k,k+1} & .. & u_{k,n} \end{bmatrix}$$

$Note: \underline{l}_k \underline{u}_k^T$ has non-zero elements in the lower $(n-k+1)$ by $(n-k+1)$ block only

Decomposition Process

- The decomposition is accomplished in *n* passes
- On pass *k*, we get
  - a) u_{kk}
  - b) column *k* of *L*
  - c) row *k* of *U*
- Initially, we start with the first column of *A*

$$a_{11} = l_{11}u_{11} \;\Rightarrow\; u_{11} = a_{11} / l_{11} = a_{11} \;\Rightarrow\; l_{11} = 1 = a_{11} / u_{11}$$

$$a_{21} = l_{21}u_{11} \;\Rightarrow\; l_{21} = a_{21}/u_{11}$$

$$\bullet$$

$$a_{n1} = l_{n1}u_{11} \;\Rightarrow\; l_{n1} = a_{n1}/u_{11}$$

- Also $a_{1j}=u_{1j}l_{11} \Rightarrow u_{1j}= a_{1j}$

$\Rightarrow$ first row of U = first row of A

- Finished computing the first column of *L* and first row of *U*
- The sequence of computation is:

$$u_{11} \rightarrow Diag(U);\; \underline{l}_1 \rightarrow \text{ first column of } L;\; \underline{u}_1^T \rightarrow \text{ first row of } U \left(\text{remaining part}\right)$$

Practical Issues

- Note: a_{i1} and a_{1j} are used once and never again
  $\Rightarrow$ Can overwrite $\underline{l}_1$ and $\underline{u}_1^T$ in the first column and row of A

Except for $\underline{l}_{11}$, which we know is 1 any way

$$l_{i1} \leftarrow a_{i1} / a_{11}, \; i = 2, \ldots, n$$

$$u_{1j} \leftarrow a_{1j}, \; j = 1, 2, \ldots, n$$

- Problem: What if a_{11}=0 ?

Example: $\begin{bmatrix} 0 & 1 \\ 1 & 6 \end{bmatrix}$ nonsingular , but a_{11}=0

Concept of Pivoting - 1

- Pivoting Idea

1) Compute $l_{i1}, \ldots, l_{n1}$ except for division $\Rightarrow l_{i1}u_{11}$

2) Find the largest $|l_{i1}|$ relative to initial row i norm

$$\Rightarrow \frac{l_{i1}}{\sum_j |a_{ij}|} \quad \forall i$$

– Assume that the maximum occurs in row r_1

$$\Rightarrow r_1 = \arg\max_i \frac{l_{i1}}{\sum_j |a_{ij}|}$$

$\Rightarrow$ 3) Swap row r_1 and 1 in A and l_1. Let $IP(1) = r_1$.
What dose it mean ?

Concept of Pivoting - 2

Multiply A by

$$P_1^{r_1} \Rightarrow \begin{bmatrix} 0 & 0 & .. & 1 & 0 \\ 0 & 1 & .. & .. & 0 \\ 0 & 0 & 1 & .. & 0 \\ 1 & 0 & .. & .. & 0 \\ 0 & .. & .. & .. & 0 \end{bmatrix}$$

PERMUTATION MATRIX

Note: $P_1^{r_1}$ is a symmetric and orthogonal $(P_1^{r_1})^{-1} = P_1^{r_1}$

4) Divide throughout by (new) $l_{11} \neq 0$ to get $l_{21}, \ldots, l_{n1}$

$$\begin{bmatrix} 1 & 6 \\ 0 & 1 \end{bmatrix} = U$$

5) $u_{11} = l_{11}$ (new). In actuality, l_{11} replaces a_{11}.

- So, really have found the first LU factor, $\underline{l}_1 \underline{u}_1^T$ of $P_1^{r_1} A = \tilde{A}$ and not of A!!
  - Can we do it recursively? Is it useful? YES!!

LU Decomposition of *PA*

- Consider the situation at column $k \geq 2$. Get column k of L and row k of U from column k and row k of $\tilde{A}$

$$\underbrace{P_{k-1}^{r_{k-1}} P_{k-2}^{r_{k-2}} \ldots P_{1}^{r_1} A}_{\tilde{A}} = \sum_{i=1}^{k-1} \underline{l}_i \underline{u}_i^T + \underline{l}_k \underline{u}_k^T + \text{other terms}$$

$$\begin{bmatrix} 1 & 0 & 0 & .. & 0 & 0 \\ l_{21} & 1 & 0 & .. & 0 & 0 \\ : & : & : & & : & : \\ l_{i1} & l_{i2} & .. & 1 & .. & 0 \\ : & : & & : & & : \\ l_{n1} & l_{n2} & .. & .. & .. & 1 \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} & .. & u_{1k} & .. & u_{1n} \\ 0 & u_{22} & .. & u_{2k} & .. & u_{2n} \\ : & : & & : & & .. \\ 0 & 0 & .. & u_{kk} & .. & u_{kn} \\ : & : & & : & & .. \\ 0 & 0 & .. & 0 & .. & u_{nn} \end{bmatrix}$$

Decomposition Steps

step 1: $\tilde{a}_{ik} = \sum_{m=1}^{k} l_{im} u_{mk} \Rightarrow l_{ik} u_{kk} = \tilde{a}_{ik} - \sum_{m=1}^{k-1} l_{im} u_{mk};\ i = k, \ldots, n$

$$\tilde{l}_{ik} = \tilde{a}_{ik} - \sum_{m=1}^{k-1} l_{im} u_{mk};\ i = k, \ldots, n$$

If $k = n$, set $u_{nn} = \tilde{l}_{nn}$ and DONE. $IP(n) = n$

step 2: Find relative $\max_i |\tilde{l}_{ik}|$, $r_k = \text{row}\ (r_k \geq k)$

step 3: swap row k and row r_k in lower right $(n-k+1)$ subblock

of $\tilde{A}$. columns $\underline{l}_1, \ldots \underline{l}_k$ lower Δ since $r_k \geq k$

step 4: If $\tilde{l}_{kk} \neq 0$, $l_{ik} = \tilde{l}_{ik} / \tilde{l}_{kk};\ i = k+1, \ldots, n$

If $\tilde{l}_{kk} = 0$, then OK since $l_{ik} = 0$

step 5: Set $u_{kk} = \tilde{l}_{kk}$ and

$$u_{kj} = \tilde{a}_{kj} - \sum_{m=1}^{k-1} l_{km} u_{mj};\ j = k+1, \ldots, n; k^{th} \text{ row of } U$$

step 6: set $k = k + 1$ and go to step 1.

Practicalities & Insights - 1

- Comments
  - Don't need 3 matrices. All work can be done in place:

$$l_{ik} \leftarrow a_{ik} \quad i = k+1, \ldots, n$$
$$u_{kj} \leftarrow a_{kj} \quad j = k, \ldots, n$$

When done
$$\begin{bmatrix} u_{11} & u_{12} & .. & u_{1n} \\ l_{21} & u_{22} & .. & u_{2n} \\ \vdots & \vdots & & \vdots \\ l_{n1} & l_{n2} & .. & u_{nn} \end{bmatrix}$$

and vector *IP* that summarizes the permutation matrices

$$P_k^{r_k}, k = 1, 2, \ldots, n$$

  - $\det PA = \det P. \quad \det A = (-1)^{\#\text{Pivots}} \prod_{i=1}^{n} u_{ii}$
  - $P_k^{r_k}$ are symmetric and orthogonal so

$$A = P_1^{r_1} P_2^{r_2} \ldots P_n^{r_n} LU$$

Practicalities & Insights - 2

- $A^{-1} = U^{-1} L^{-1} P_n^{r_n} \ldots P_1^{r_1}$
- Number of operations

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

$$= \frac{n(n-1)(n+1)}{3} = O(\frac{n^3}{3})$$

- Relative round-off error

  $\| \bar{L}\bar{U} - PA \|$ proportional to $k(A) f(n) \varepsilon_m$,

  where $k(A) =$ condition number of A and $\varepsilon_m =$ machine accuracy
- Pivoting is essential. Otherwise, the method can be unstable
- Accumulate all inner products in DOUBLE PRECISION

Solution of $A\underline{x} = \underline{b}$ - 1

- Remaining step: solution of $A\underline{x}=\underline{b}$

$$PA\underline{x} = P\underline{b} \Rightarrow \underline{\tilde{b}} = P_n^{r_n} P_{n-1}^{r_{n-1}} \ldots P_1^{r_1} \underline{b}$$

$\Rightarrow$ swap $\underline{b}_1 \leftrightarrow \underline{b}_{r_1}$ etc. can do it in place

$$\Rightarrow LU\underline{x} = \underline{\tilde{b}}$$

- Solve:
  - $L\underline{y} = \underline{\tilde{b}}$; via FORWARD ELIMINATION and
  - $U\underline{x} = \underline{y}$; via BACKWARD SUBSTITUTION

$$\begin{bmatrix} 1 & 0 & 0 & .. & 0 & 0 \\ l_{21} & 1 & & & & 0 \\ : & & .. & & & 0 \\ l_{i1} & l_{i2} & .. & 1 & & 0 \\ : & & & & .. & 0 \\ l_{n1} & l_{n2} & .. & .. & .. & 1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ : \\ : \\ : \\ y_n \end{bmatrix} = \begin{bmatrix} \tilde{b}_1 \\ \tilde{b}_2 \\ : \\ : \\ : \\ \tilde{b}_n \end{bmatrix} \Rightarrow \begin{array}{l} y_1 = \tilde{b}_1 \\ y_2 = \tilde{b}_2 - l_{21} y_1 \\ y_k = \tilde{b}_k - \sum_{j=1}^{k-1} l_{kj} y_j ; k = 1, 2, .., n \end{array}$$

Solution of $A\underline{x} = \underline{b}$ - 2

- $O(n^2 - n)/2$ ops
- Can overwrite on $\tilde{b}_k$ with y_k

$$\begin{bmatrix} u_{11} & u_{12} & .. & u_{1k} & .. & u_{1n} \\ 0 & u_{22} & & & & u_{2n} \\ : & : & .. & & & : \\ 0 & 0 & .. & u_{kk} & & u_{kn} \\ : & : & & & .. & : \\ 0 & 0 & .. & .. & .. & u_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ : \\ : \\ : \\ x_n \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ : \\ : \\ : \\ y_n \end{bmatrix} \Rightarrow \begin{aligned} x_n &= y_n / u_{nn} \\ x_{n-1} &= \frac{y_{n-1} - u_{n-1,n} x_n}{u_{n-1,n-1}} \\ x_k &= (y_k - \sum_{i=k+1}^{n} l_{ki} x_i) / u_{kk} \end{aligned}$$

- $O(n^2 + n/2)$ ops
- Total ops $O(n^2) \Rightarrow$ Total $= O(n^3/3) + O(n^2)$

❑ Error bounds

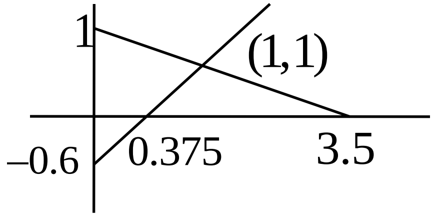
- For Doolittle with DP accumulation of products
  $\bar{L}\bar{U} = P(A+E);\ \|E\|_\infty \le n g_n \in_m \|A\|_\infty;\ g_n \le 2^{n-1},$
- Pessimistic estimate. Generally $g_n \approx \min(8, n)$

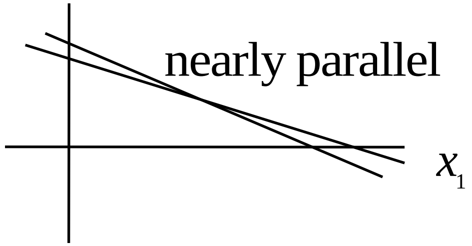
Sensitivity of Linear Systems

- Sensitive of linear systems

Example :

$$8x_1 - 5x_2 = 3 \qquad 0.66x_1 - 3.34x_2 = 4$$
$$4x_1 + 10x_2 = 14 \qquad 1.99x_1 + 10.01x_2 = 12$$





- Suppose $\underline{b} \to \underline{b} + \delta\underline{b}$ where

$$\underline{b} + \delta\underline{b} = \begin{bmatrix} 2.96 \\ 13.94 \end{bmatrix} \Rightarrow \frac{\|\delta\underline{b}\|_\infty}{\|\underline{b}\|_\infty} = 0.0043$$

$$x_{new} = \begin{bmatrix} 0.993 \\ 0.997 \end{bmatrix} \Rightarrow \frac{\|\delta\underline{x}\|_\infty}{\|\underline{x}\|_\infty} = 0.007 \Rightarrow 0.7\% \text{ change}$$

$\Rightarrow$ well-conditioned

$$\Rightarrow \quad \textit{Bode Sensitivity}: S_b^x = \frac{\|\delta\underline{x}\|_\infty}{\|\underline{x}\|_\infty} / \frac{\|\delta\underline{b}\|_\infty}{\|\underline{b}\|_\infty} = 1.63$$

$$\underline{b} + \delta\underline{b} = \begin{bmatrix} 3.96 \\ 11.94 \end{bmatrix} \Rightarrow \frac{\|\delta\underline{b}\|_\infty}{\|\underline{b}\|_\infty} = 0.005$$

$$x_{new} = \begin{bmatrix} 6 \\ 0 \end{bmatrix} \Rightarrow \frac{\|\delta\underline{x}\|_\infty}{\|\underline{x}\|_\infty} = 5 \Rightarrow 500\% \text{ change}$$

$\Rightarrow$ ill-conditioned

$$\Rightarrow \text{Bode Sensitivity} = \frac{\|\delta\underline{x}\|_\infty}{\|\underline{x}\|_\infty} / \frac{\|\delta\underline{b}\|_\infty}{\|\underline{b}\|_\infty} = 1000$$

Error Analysis - 1

Let $\underline{\hat{x}}$ be the computed solution to the linear system $A\underline{x} = \underline{b}$ and let $\underline{x}^*$ be the true solution. There are two measures of the discrepancy in $\underline{\hat{x}}$:

- error : $\underline{e} = \underline{x}^* - \underline{\hat{x}}$
- residual : $\underline{r} = \underline{b} - A\underline{\hat{x}} = A(\underline{x}^* - \underline{\hat{x}}) = A\underline{e}$

❑ Key results: (valid for **any** norm. we will use ∞-norm here)

- LU decomposition with partial pivoting is guaranteed to produce small residuals, i.e., small $\|\underline{r}\|$.

$$\| r \|_\infty \leq n g_n \| A \|_\infty \| \underline{\hat{x}} \|_\infty \varepsilon_m$$

- However, error depends on the condition number of A, i.e., how close A is to being "near singular".

$$\| \underline{e} \|_\infty \leq n g_n \kappa(A) \| \underline{\hat{x}} \|_\infty \varepsilon_m ; \; \kappa(A) = \| A \|_\infty \| A^{-1} \|_\infty$$

- Larger $\kappa(A) \Rightarrow$ error is larger or more sensitive to changes in A and $\underline{b}$. (Note: $\kappa(A)$ can be defined w.r.t. any norm)

❑ Consider changes in b only:

- Suppose that LU is exact, but the data vector $\underline{b}$ is "noisy".
- Q: How "sensitive" is the solution?

Error Analysis - 2

$$A(\underline{x}^* + \delta\underline{\hat{x}}) = \underline{b} + \delta\underline{b} = A\underline{x}^* + \delta\underline{b}$$

$$\Rightarrow A\delta\underline{x} = \delta\underline{b}$$

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

since, $1/\|x^*\| \le \|A\|/\|b\|$, we have

$$\boxed{\|A\underline{x}^*\| = \|\underline{b}\| \le \|A\|\|\underline{x}^*\| \Rightarrow \frac{1}{\|\underline{b}\|} \ge \frac{1}{\|A\|\|\underline{x}^*\|}}$$

$$\frac{\|\delta x\|}{\|x^*\|} \le \frac{\|A^{-1}\|\|A\|\|\delta\underline{b}\|}{\|\underline{b}\|} = \kappa(A)\frac{\|\delta\underline{b}\|}{\|\underline{b}\|}$$

$\boxed{\kappa(A) \text{ is like Bode Sensitivity}}$

- Consider changes in A only:
  - The computed solution $\underline{\hat{x}}$ is the true solution to $(A+E)\underline{\hat{x}} = \underline{b}$, where $\|E\|_\infty \le ng_n\varepsilon_m\|A\|_\infty$

$$\Rightarrow \underline{r} = \underline{b} - A\underline{\hat{x}} = E\underline{\hat{x}}$$

  - So,

$$\|\underline{r}\|_\infty = \|E\underline{\hat{x}}\|_\infty \le \|E\|_\infty\|\underline{\hat{x}}\|_\infty \le ng_n\varepsilon_m\|A\|_\infty\|\underline{\hat{x}}\|_\infty$$

$$\frac{\|\underline{r}\|_\infty}{\|A\|_\infty\|\underline{\hat{x}}\|_\infty} \le ng_n\varepsilon_m \Rightarrow \text{size of residuals is small}$$

Error Analysis - 3

- What about error, $\underline{e} = \underline{x}^* - \underline{\hat{x}}$?

  $\underline{e} = A^{-1}\underline{r}$

  $\|\underline{e}\|_\infty \le \|A^{-1}\|_\infty \|\underline{r}\|_\infty \le \|A^{-1}\|_\infty \|E\|_\infty \|\underline{\hat{x}}\|_\infty$

  $$\frac{\|\underline{e}\|_\infty}{\|\underline{\hat{x}}\|_\infty} \le \|A^{-1}\|_\infty \|A\|_\infty \frac{\|E\|_\infty}{\|A\|_\infty} = ng_n \|A^{-1}\|_\infty \|A\|_\infty \varepsilon_m = ng_n \varepsilon_m \kappa(A)$$

- So, condition A, $\kappa(A)$ is an amplification factor.
- $\kappa(A) \ge 1$ is a measure of how close A is to singularity.
- Larger $\kappa(A) \Leftrightarrow$ more difficult to solve $A\underline{x}=\underline{b}$

❑ Changes in both A and b

- It is easy to show that (e.g., by linearity and neglecting second order term $E\,\underline{e}$) that

$$\frac{\|\underline{e}\|_\infty}{\|\underline{\hat{x}}\|_\infty} \le [ng_n\varepsilon_m + \frac{\|\delta\underline{b}\|_\infty}{\|\underline{b}\|_\infty}]\kappa(A)$$

Test Matrices

- Some difficult test matrices:
  - Hilbert $a_{ij} = 1/(i+j-1)$; $\kappa(A) = 10^n$; n=size of matrix

$$\begin{bmatrix} 1 & 1/2 & 1/3 & 1/4 \\ 1/2 & 1/3 & 1/4 & 1/5 \\ 1/3 & 1/4 & 1/5 & 1/6 \\ 1/4 & 1/5 & 1/6 & 1/7 \end{bmatrix}$$

  - Poisson $a_{ij} = a_{i-1,j} + a_{i,j+1}$; $\kappa(A) = 10^n$; n=size of matrix

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 \\ 1 & 3 & 6 & 10 \\ 1 & 4 & 10 & 20 \end{bmatrix}$$

  - Others from books on test matrices

How to Reduce Errors?

- What can we do about errors?
  - Iterative improvement (method of residual correction)
  - Balancing the A matrix

- Iterative improvement
  - Suppose $A\underline{x}=\underline{b}$ has been solved via $PA=LU$
  - Suppose $\hat{\underline{x}} = \underline{x}_0$ is the solution
  - Can I improve the solution $\underline{x}_0$ knowing the residual $\underline{r}_0 = \underline{b} - A\underline{x}_0$?

    YES !!
  - $LU \sim O(n^3/3)$; $\underline{x}_0$ in $O(n^2)$

    Consider true $\underline{x}^* = \underline{x}_0 + \underline{e}$

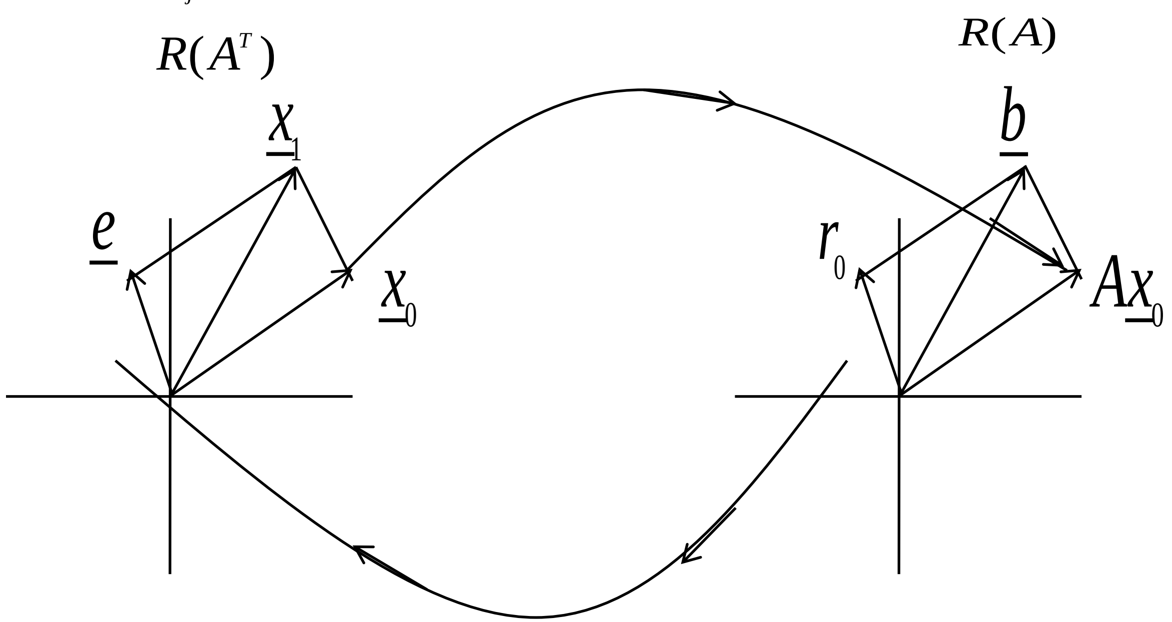
    $\Rightarrow A\underline{x}^* - \underline{b} = 0; A\underline{x}_0 - \underline{b} + A\underline{e} = 0;$

    $\Rightarrow A\underline{e} = \underline{b} - A\underline{x}_0 = \underline{r}_0$ (residual)
  - So, solve for $\underline{e}$ via $\bar{L}\bar{U}\underline{e} = P\,\underline{r}_0$ using decomposition obtained already !
  - Feasible to do, since requires only $O(n^2)$ operations.

Iterative Improvement - 1

- Then $\underline{x}_1 = \underline{x}_0 + \underline{e}$ is the improved solution. We can repeat the process.
- But, critical that we accumulate $\underline{r}_0 = \underline{b} - A\underline{x}_0$ in double precision. Otherwise, $\underline{e}$ obtained will be worthless.
- Geometrically,

$$r_{oi} = b_i - \sum_{j=1}^{n} a_{ij} x_{0j}$$



Iterative Improvement - 2

$$\underline{x}_0 = \underline{\hat{x}}$$
$$l = 0$$
$$\underline{r}_l = \underline{b} - A\underline{x}_l$$
$l=l+1$ Solve $\bar{L}\bar{U}\underline{e}_l = P\underline{r}_l$ for $\underline{e}_l$
$$\underline{x}_{l+1} = \underline{x}_l + \underline{e}_l$$

rapid convergence, but still up against finite word length

$\Rightarrow$ do 2-3 iterations

- Heuristic: If $k_\infty(A) = ||A||_\infty \, ||A^{-1}||_\infty = 2^q$ then after " l " iterations through the loop, $\underline{x}$ will have approximately $min(t, l(t-q))$ bits of accuracy
- Note: need original A stored somewhere to compute residual
- Stopping criteria
  - If (1) $||r||_\infty / ||x||_\infty < \varepsilon$
    (2) $l > l_{\max}$
    (3) $||\underline{e}_l|| / ||\underline{x}_l|| > ||\underline{e}_{l-1}|| / ||\underline{x}_{l-1}||$ ... guards against oscillations, when approaching ε_m

Convergence Analysis

- When does this process work ? i.e., will $\underline{x}_l \to \underline{x}^*$
  - Solution $\underline{e}_l = \underline{x}_{l+1} - \underline{x}_l$ satisfies

    $(A+E)\underline{e}_l = \underline{r}_l$ with $\| E \| \le \varepsilon_m f(n) \| A \|$; $f(n) = ng_n$ for ∞-norm
  - $A(I+F)\underline{e}_l = \underline{r}_l$; $F = A^{-1}E$ and $\| F \| \le k(A)\varepsilon_m f(n)$
  - Assume $\| F \| < 1/2$; so $(I+F)^{-1}$ exists and

$$(I+F)^{-1} = I - F + F^2 - F^3 + \ldots$$

$$\| (I+F)^{-1} \| \le 1 + \| F \| + \| F \|^2 + \| F \|^3 \ \ldots \ = 1/(1 - \| F \|)$$

$$A(I+F)\underline{e}_l = \underline{r}_l = \underline{b} - A\underline{x}_l$$

$$(I+F)\underline{e}_l = A^{-1}\underline{b} - \underline{x}_l = (\underline{x}^* - \underline{x}_l) \Rightarrow (I+F)(\underline{x}_{l+1} - \underline{x}_l) = (\underline{x}^* - \underline{x}_l)$$

$$(I+F)\underline{x}_{l+1} = F\underline{x}_l + \underline{x}^* \Rightarrow (I+F)(\underline{x}_{l+1} - \underline{x}^*) = F(\underline{x}_l - \underline{x}^*)$$

$$(\underline{x}_{l+1} - \underline{x}^*) = (I+F)^{-1} F (\underline{x}_l - \underline{x}^*)$$

$$\| \underline{e}_{l+1} \| \le \| (I+F)^{-1} \| \| F \| \| \underline{e}_l \| \le [\| F \| / (1\text{-}\| F \|)] \| \underline{e}_l \|$$

  - Since $\| F \| < 1/2 \ \Rightarrow \ \| \underline{e}_{l+1} \| \le \tau \| \underline{e}_l \|$ where $\tau = \| F \| / (1\text{-}\| F \|) < 1$

    $\Rightarrow$ linear convergence
  - If $\tau = 0.1$, pick up at least one digitaccuracy with each iteration.

Balancing

- Balancing
  - Can we transform $A \to \bar{A} \ni \kappa(\bar{A}) \ll \kappa(A)$ and solve $A\underline{x} = \underline{b}$ using $\bar{A}$. Yes, <u>in some cases</u>, but not by a scalar.
  - Need <u>Diagonal</u> scaling

$$\bar{A} = D^{-1}AD; D = \text{diag}(d_1 \ d_2 \ .... \ d_n)$$
$$= \text{diag}(2^{i_1} \ 2^{i_2} \ .... \ 2^{i_n})$$

$$A\underline{x} = \underline{b} \ \Rightarrow D^{-1}ADD^{-1}\underline{x} = D^{-1}\underline{b} \Rightarrow \bar{A}\underline{y} = \bar{b} \Rightarrow \text{solve for } \underline{y} \text{ and } \underline{x} = D\underline{y}$$

$$\bar{A} = [a_{ij}d_j / d_i]; \ \bar{b}_i = b_i / d_i;$$

$\Rightarrow$ Try to pick d_k such that k^{th} row of $\bar{A}$ and k^{th} column of $\bar{A}$ have $\approx$ same norm. That is, $\sum_j |\bar{a}_{kj}| \approx \sum_i |\bar{a}_{ik}|$

$$\Rightarrow \sum_j |a_{kj}d_j / d_k| = \sum_j |a_{jk}d_k / d_j|$$

$\Rightarrow$ Balancing is useful.

$\Rightarrow$ Note that similarity transformation has no effect on $k(A)$ of a symmetric matrix.

$\Rightarrow$ Usually standard controllable form and standard observable form have worst $\kappa(A)$

Estimation of $\kappa(A)$ - 1

- Estimation of $\kappa(A)$: Method 1

  Assume $\delta \underline{b} = 0$

  $\| \underline{e} \|_\infty \leq n g_n \kappa(A) \| \hat{\underline{x}} \|_\infty \, \varepsilon_m$

  From the 1[st] step of iterative process, obtain $\underline{e}$ and $\hat{\underline{x}}_1$.

  Estimate $k(A) \approx [1/\varepsilon_m n g_n][\| \underline{e} \|_\infty / \| \hat{\underline{x}}_1 \|_\infty]$

  Do this on 1st step only. Use $g_n \approx 1$.

  Generally, the estimate is not very accurate !

- Estimation of $\kappa(A)$: Method 2 (provides a good estimate)
  - Consider $\kappa(A) = ||A||_\infty \, ||A^{-1}||_\infty$
  - Can easily obtain $||A||_\infty$
  - The problem is to get $||A^{-1}||_\infty$.
  - Consider $A\underline{y}=\underline{d}$

    $\Rightarrow \; ||y||_\infty \leq ||A^{-1}||_\infty \, ||d||_\infty$

    $||A^{-1}||_\infty \geq ||y||_\infty \, / \, ||d||_\infty$
  - Choose d_k from (-1,1). can do it if A is upper triangular.

    Idea: choose $\underline{d} \ni \| y \|_\infty$ is as large as possible !

Estimation of $\kappa(A)$ - 2

- Read: A.K. Cline, C.B. Moler, G.W. Stewart and J.H. Wilkinson, " An estimate for the condition number of a matrix," SIAM J. of Numerical Analysis, vol. 16, 1979, pp. 368-375.
- Suppose A is upper triangular

$$\begin{bmatrix} a_{11} & a_{12} & .. & a_{1k} & .. & a_{1n} \\ 0 & a_{22} & & & & a_{2n} \\ \vdots & & .. & & & \vdots \\ 0 & 0 & .. & a_{kk} & & a_{kn} \\ \vdots & & & & .. & \vdots \\ 0 & 0 & .. & .. & .. & a_{nn} \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ \vdots \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ \vdots \\ \vdots \\ d_n \end{bmatrix} \Rightarrow y_k = (d_k - p_k)/a_{kk}$$

$$\text{where } p_k = \sum_{j=k+1}^{n} a_{kj} y_j$$

Estimation of $\kappa(A)$ - 3

- Computation of p_i^s :

  Initally let $p_i = 0 \quad i = 1, 2, ..., n$

  For $k = n, ..., 1$ DO

  $$y_k = (d_k - p_k) / a_{kk}$$

  $$p_i = p_i + a_{ik} y_k ; \; i = 1, 2, ..., k-1$$

  End DO

- Q: Can we pick d_k such that $||y||_\infty$ is large $\Rightarrow ||y||_\infty >> ||d||_\infty$
  A: select d_k from (-1,1) according to whether $(1-p_k) / a_{kk}$ or $-(1+p_k)/ a_{kk}$ is large, i.e., $y_k = (-\text{sign}(p_k) - p_k)/ a_{kk}$
- Since $||d||_\infty = 1 \Rightarrow || \kappa ||_\infty = ||A||_\infty ||y||_\infty$, $|| \kappa ||_\infty$ = condition number of A using ∞ - norm.

Estimation of $\kappa(A)$ - 4

- A more complicated estimator:
  - Encourage growth in y_k and running sums of $p_1, \ldots, p_{k-1}$
  - Algorithm:

$$\text{Let } w_1, w_2, \ldots, w_n \text{ be a set of weights } (w_i \propto 1/|a_{ii}|)$$

$$p_i = 0 \quad i = 1, 2, \ldots, n$$

$$\text{For } k = n, \ldots, 1$$

$$y_k^+ = (1 - p_k)/a_{kk}$$

$$y_k^- = -(1 + p_k)/a_{kk}$$

$$s^+ = |y_k^+| + \sum_{i=1}^{k-1} w_i \, |p_i + a_{ik} y_k^+|$$

$$s^- = |y_k^-| + \sum_{i=1}^{k-1} w_i \, |p_i + a_{ik} y_k^-|$$

Estimation of $\kappa(A)$ - 5

if $s^{+} \geq s^{-}$ then

$$y_k = y_k^{+}$$

else

$$y_k = y_k^{-}$$

end if

$$p_i = p_i + a_{ik} y_k;\ i = 1, 2, \ldots, k-1$$

end

- Requires $O(5n^2/2)$ flops.

$\Rightarrow$ can devise a lower Δ version easily.

- For general A: know LU of PA
  - Recall that

$$\| A^{-1} \| = 1 / \min_{\underline{x}} \frac{\| A\underline{x} \|}{\| \underline{x} \|} = \max_{\underline{x}} \frac{\| \underline{x} \|}{\| A\underline{x} \|} = \max_{\underline{y}} \frac{\| A^{-1}\underline{y} \|}{\| \underline{y} \|}$$

where $\underline{y} = A\underline{x}$.

Estimation of $\kappa(A)$ - 6

- Idea: pick $\underline{y}$ **carefully**, solve $A\underline{z} = \underline{y}$ using LU factors and use $\|A^{-1}\|_\infty = \|\underline{z}\|_\infty / \|\underline{y}\|_\infty$.
- How to pick y:
  – Suppose A is ill-conditioned $\Rightarrow$ U is ill-conditioned, L is generally OK.
  – Recall that $A^\dagger = V\Sigma^\dagger U^T$ where V and U are orthogonal (note: U is not an LU factor here !!) $\Rightarrow$ vector $\underline{y}$ tends to be rich in the direction of left singular vector associated with $\sigma_{\min}(A)$.
  – One way of getting such a $\underline{y}$ is to solve : $A^T P\,\underline{y}=\underline{d}$, where $\underline{d}$ is a vector with $\pm$ 1 elements, which are chosen to maximize $\|\underline{y}\|_\infty$
- So, to get $\underline{y}$:

1) Solve $U^T\underline{w}= \underline{d}$ using a lower Δ version of the algorithm.
2) Solve $L^T[P\underline{y}] = \underline{w} \Rightarrow \underline{y} = P(L^{-1})^T(U^{-1})^T\underline{d} = (A^{-1})^T\underline{d}$ or solves $A^T\underline{y} = \underline{d}$

Estimation of $\kappa(A)$ - 7

To get $\underline{z}$:

3) Solve $L\,\underline{r}=P\,\underline{y}$

4) Solve $U\,\underline{z} = \underline{r} \Rightarrow \underline{z} = U^{-1}L^{-1}P\underline{y} = A^{-1}\underline{y}$ or solves $A\underline{z}=\underline{y}$

$$||\underline{z}||_\infty \leq ||A^{-1}||_\infty\,||\underline{y}||_\infty$$

$$||\kappa(A)||_\infty \approx ||A||_\infty \cdot ||\underline{z}||_\infty / ||\underline{y}||_\infty$$

- Example: consider the following 2×2 matrix and its factors

$$A = \begin{bmatrix} 0.66 & 3.34 \\ 1.99 & 10.01 \end{bmatrix}$$

$$= PLU = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}\begin{bmatrix} 1 & 0 \\ -0.3317 & 10.01 \end{bmatrix}\begin{bmatrix} 1.99 & 10.01 \\ 0 & 0.0201 \end{bmatrix}$$

- Using $\underline{d} = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$, step 1 gives: $\underline{w} = \begin{bmatrix} 0.5025 \\ -300.0075 \end{bmatrix}$

Estimation of $\kappa(A)$ - 8

- step 2 gives: $P\underline{y} = \begin{bmatrix} -99.0100 \\ -300.0075 \end{bmatrix}$
- step 3 gives: $\underline{r} = \begin{bmatrix} -99.0100 \\ -332.8491 \end{bmatrix}$
- step 4 gives: $\underline{z} = \begin{bmatrix} 83248 \\ -16560 \end{bmatrix}$
- $\| \kappa(A) \|_\infty \approx \|A\|_\infty . \|\underline{z}\|_\infty / \|\underline{y}\|_\infty$ = 12* 83248/300.0075 = 3329.8
- Actual condition number of A using ∞–norm = 4005
- The estimate is within 16.854% of actual value.

❑ Rank-one updates

- Suppose have solved $A\underline{x}=\underline{b}$. But, now want to solve a slightly modified problem: $\tilde{A}\underline{\tilde{x}} = \underline{b}$ where $\tilde{A} = A + \underline{u}\,\underline{v}^T$
- Know from Sherman-Morrison-Woodbury formula that:

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

Rank One Updates

- So, to solve

$$\tilde{A}\underline{\tilde{x}} = [A + \underline{u}\underline{v}^T]\underline{x} = \underline{b}:$$

(a) solve $A\underline{x} = \underline{b} \Rightarrow \underline{x} = A^{-1}\underline{b}$

(b) solve $A\underline{y} = \underline{u} \Rightarrow \underline{y} = A^{-1}\underline{u}$

(c) solve $A^T\underline{z} = \underline{v} \Rightarrow \underline{z} = [A^{-1}]^T\underline{v}$

(d) obtain $\alpha = 1/(1 - \underline{v}^T\underline{y})$

(e) obtain $\beta = \underline{z}^T\underline{b}$

(f) obtain $\underline{\tilde{x}} = \underline{x} + \alpha\beta\underline{y}$

- For LU updates with rank-one corrections to a matrix, see:

"P.E. Gill, G.H. Golub, W. Murray, and M.A. Saunders, " Methods for modifying matrix factorizations ," Mathematics of Computation, Vol. 28, pp. 311-350, 1974 .

Summary

- ❑ Why do we need to solve $A\underline{x} = \underline{b}$?
- ❑ Concepts of **forward elimination** and **backward substitution**
- ❑ Basic decomposition methods: *LU*, *QR*, Cholesky, SVD
- ❑ *LU* decomposition
- ❑ Sensitivity of the solution to $A\underline{x} = \underline{b}$
  - Error and residual
  - **Condition number** as an amplification factor for error
- ❑ Iterative improvement
- ❑ Estimation of condition number
- ❑ Solution when A is modified by a rank-one correction matrix