Matrix Inverse and Condition

Berlin Chen
Department of Computer Science & Information Engineering
National Taiwan Normal University

Reference:

1. *Applied Numerical Methods with MATLAB for Engineers*, Chapter 11 & Teaching material

Chapter Objectives

- Knowing how to determine the matrix inverse in an efficient manner based on *LU* factorization
- Understanding how the matrix inverse can be used to assess stimulus-response characteristics of engineering systems
- Understanding the meaning of matrix and vector norms and how they are computed
- Knowing how to use norms to compute the matrix condition number
- Understanding how the magnitude of the condition number can be used to estimate the precision of solutions of linear algebraic equations

Matrix Inverse (1/4)

- Recall that if a matrix $[A]$ is ***square***, there would be another matrix $[A]^{-1}$, called the ***inverse*** of $[A]$, for which $[A][A]^{-1}=[A]^{-1}[A]=[I]$ ([I]: identity matrix)

- The ***inverse*** can be computed in ***a column by column fashion*** by generating solutions with ***unit*** vectors as the right-hand-side constants:
  - A three-variable system

$$[A]\{x_1\} = \begin{Bmatrix} 1 \\ 0 \\ 0 \end{Bmatrix} \quad [A]\{x_2\} = \begin{Bmatrix} 0 \\ 1 \\ 0 \end{Bmatrix} \quad [A]\{x_3\} = \begin{Bmatrix} 0 \\ 0 \\ 1 \end{Bmatrix}$$

$$[A]^{-1} = \begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix}$$

Matrix Inverse (2/4)

- Recall that *LU* factorization can be used to efficiently evaluate a system for multiple right-hand-side vectors - thus, it is ideal for evaluating the multiple unit vectors needed to compute the inverse

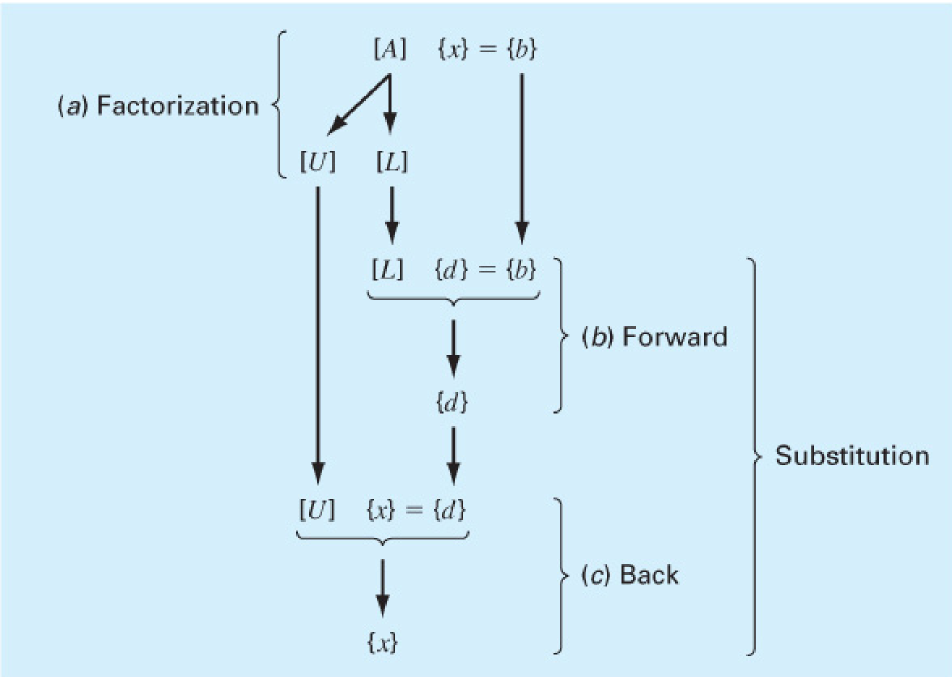

FIGURE 10.1
The steps in *LU* factorization.

1. To solve [*A*]{*x*}={*b*}, first decompose [*A*] to get [*L*][*U*]{*x*}={*b*} (let {*d*}=[*U*]{*x*}={*d*})
2. Set up and solve [*L*]{*d*}={*b*}, where {*d*} can be found using *forward* substitution
3. Set up and solve [*U*]{*x*}={*d*}, where {*x*} can be found using *backward* substitution

Matrix Inverse (3/4)

Example 11.1

Problem Statement. Employ LU factorization to determine the matrix inverse for the system from Example 10.1:

$$[A] = \begin{bmatrix} 3 & -0.1 & -0.2 \\ 0.1 & 7 & -0.3 \\ 0.3 & -0.2 & 10 \end{bmatrix}$$

Recall that the factorization resulted in the following lower and upper triangular matrices:

$$[U] = \begin{bmatrix} 3 & -0.1 & -0.2 \\ 0 & 7.00333 & -0.293333 \\ 0 & 0 & 10.0120 \end{bmatrix} \qquad [L] = \begin{bmatrix} 1 & 0 & 0 \\ 0.0333333 & 1 & 0 \\ 0.100000 & -0.0271300 & 1 \end{bmatrix}$$

Solution. The first column of the matrix inverse can be determined by performing the forward-substitution solution procedure with a unit vector (with 1 in the first row) as the right-hand-side vector. Thus, the lower triangular system can be set up as (recall Eq. [10.8])

$$\begin{bmatrix} 1 & 0 & 0 \\ 0.0333333 & 1 & 0 \\ 0.100000 & -0.0271300 & 1 \end{bmatrix} \begin{Bmatrix} d_1 \\ d_2 \\ d_3 \end{Bmatrix} = \begin{Bmatrix} 1 \\ 0 \\ 0 \end{Bmatrix}$$

and solved with forward substitution for $\{d\}^T = \lfloor 1 \quad -0.03333 \quad -0.1009 \rfloor$. This vector can then be used as the right-hand side of the upper triangular system (recall Eq. [10.3]):

$$\begin{bmatrix} 3 & -0.1 & -0.2 \\ 0 & 7.00333 & -0.293333 \\ 0 & 0 & 10.0120 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix} = \begin{Bmatrix} 1 \\ -0.03333 \\ -0.1009 \end{Bmatrix}$$

which can be solved by back substitution for $\{x\}^T = \lfloor 0.33249 \quad -0.00518 \quad -0.01008 \rfloor$, which is the first column of the matrix inverse:

$$[A]^{-1} = \begin{bmatrix} 0.33249 & 0 & 0 \\ -0.00518 & 0 & 0 \\ -0.01008 & 0 & 0 \end{bmatrix}$$

Matrix Inverse (4/4)

Example 11.1

To determine the second column, Eq. (10.8) is formulated as

$$\begin{bmatrix} 1 & 0 & 0 \\ 0.0333333 & 1 & 0 \\ 0.100000 & -0.0271300 & 1 \end{bmatrix} \begin{Bmatrix} d_1 \\ d_2 \\ d_3 \end{Bmatrix} = \begin{Bmatrix} 0 \\ 1 \\ 0 \end{Bmatrix}$$

This can be solved for $\{d\}$, and the results are used with Eq. (10.3) to determine $\{x\}^T = \lfloor 0.004944 \quad 0.142903 \quad 0.00271 \rfloor$, which is the second column of the matrix inverse:

$$[A]^{-1} = \begin{bmatrix} 0.33249 & 0.004944 & 0 \\ -0.00518 & 0.142903 & 0 \\ -0.01008 & 0.002710 & 0 \end{bmatrix}$$

Finally, the same procedures can be implemented with $\{b\}^T = \lfloor 0 \quad 0 \quad 1 \rfloor$ to solve for $\{x\}^T = \lfloor 0.006798 \quad 0.004183 \quad 0.09988 \rfloor$, which is the final column of the matrix inverse:

$$[A]^{-1} = \begin{bmatrix} 0.33249 & 0.004944 & 0.006798 \\ -0.00518 & 0.142903 & 0.004183 \\ -0.01008 & 0.002710 & 0.099880 \end{bmatrix}$$

The validity of this result can be checked by verifying that $[A][A]^{-1} = [I]$.

Stimulus-Response Computations (1/3)

- Many systems can be modeled as a linear combination of equations, and thus written as a matrix equation:

$$[\text{Interactions}]\{\text{response}\} = \{\text{stimuli}\}$$

- The system response can thus be found using the matrix inverse

Stimulus-Response Computations (2/3)

- Example: Three Bungee Jumpers

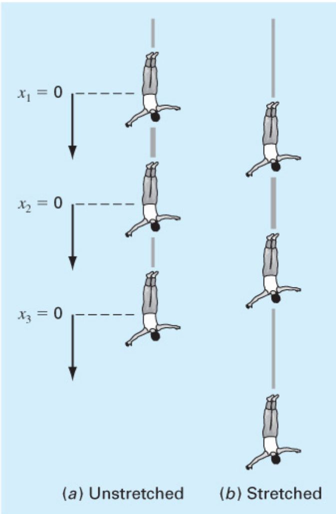

FIGURE 8.1
Three individuals connected by bungee cords.

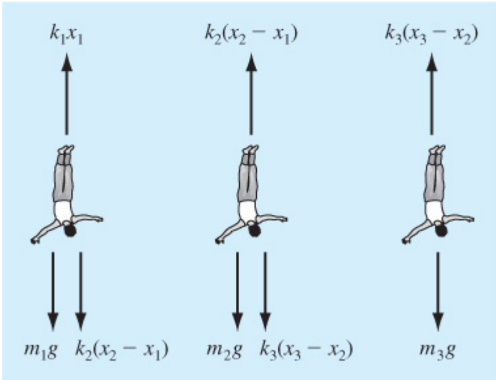

FIGURE 8.2
Free-body diagrams.

compute the displacement of each of the jumpers when coming to the equilibrium positions

$$
\begin{aligned}
m_1\frac{d^2x_1}{dt^2} &= m_1g + k_2(x_2 - x_1) - k_1x_1 \\
m_2\frac{d^2x_2}{dt^2} &= m_2g + k_3(x_3 - x_2) + k_2(x_1 - x_2) \\
m_3\frac{d^2x_3}{dt^2} &= m_3g + k_3(x_2 - x_3)
\end{aligned}
\quad\Longrightarrow\quad
\begin{aligned}
(k_1 + k_2)x_1 \quad - k_2x_2 \qquad\qquad &= m_1g \\
-k_2x_1 + (k_2 + k_3)x_2 - k_3x_3 &= m_2g \\
-k_3x_2 + k_3x_3 &= m_3g
\end{aligned}
\quad\Longrightarrow\quad
[A]\{x\} = \{b\}
$$

stimuli/forcing function

interactions

response

Stimulus-Response Computations (3/3)

- The matrix inverse provides a powerful technique for understanding the interrelationships of component parts of complicated systems

$$[A]\{x\} = \{b\}$$

$$\Rightarrow \{x\} = [A]^{-1}\{b\}$$

or

$$x_1 = a_{11}^{-1}b_1 + a_{12}^{-1}b_2 + a_{13}^{-1}b_3$$
$$x_2 = a_{21}^{-1}b_1 + a_{22}^{-1}b_2 + a_{23}^{-1}b_3$$
$$x_3 = a_{31}^{-1}b_1 + a_{32}^{-1}b_2 + a_{33}^{-1}b_3$$

Each of its element a_{ij}^{-1} represents the response of a single part of the system to a unit stimulus of any other part of the system.

where

$$[A]^{-1} = \begin{bmatrix} a_{11}^{-1} & a_{12}^{-1} & a_{13}^{-1} \\ a_{21}^{-1} & a_{22}^{-1} & a_{22}^{-1} \\ a_{31}^{-1} & a_{32}^{-1} & a_{33}^{-1} \end{bmatrix}$$

Element a_{ij}^{-1} of the matrix inverse represents, for example, the force in member *i* due to a unit external force at node *j*.

Ill-Conditioned Systems

- Three direct methods for discerning whether systems are ill-conditioned
  1. Scale the matrix of coefficients [A] so that the largest element in each row is 1. Invert the scaled matrix and if there are elements of $[A]^{-1}$ that are several orders of magnitude greater than one, it is likely that the system is ill-conditioned
  2. Multiply the inverse $[A]^{-1}$ by the original coefficient matrix [A] and assess whether the result is close to the identity matrix [I], If not, it indicates ill-conditioning
  3. Invert the inverted matrix and assess whether the result is sufficiently close to the original coefficient matrix. If not, it indicates ill-conditioning

- Can we obtain a single number serving as an indicator of ill-conditioned systems?

Vector and Matrix Norms

- A ***norm*** is a real-valued function that provides a measure of the size or “length” of multi-component mathematical entities such as vectors and matrices

- Vector norms and matrix norms may be computed differently

Vector Norms

- For a vector {*X*} of size *n*, the *p*-norm is:

$$\|X\|_p = \left(\sum_{i=1}^{n} |x_i|^p \right)^{1/p}$$

- Important examples of vector *p*-norms include:

$$p = 1 : \text{sum of the absolute values} \quad \|X\|_1 = \sum_{i=1}^{n} |x_i|$$

$$p = 2 : \text{Euclidian norm (length)} \quad \|X\|_2 = \|X\|_e = \sqrt{\sum_{i=1}^{n} x_i^2}$$

$$p = \infty : \text{maximum} - \text{magnitude} \quad \|X\|_\infty = \max_{1 \le i \le n} |x_i|$$

Matrix Norms

- Common matrix norms for a matrix [A] include:

$$\text{column-sum norm} \qquad \|A\|_1 = \max_{1 \le j \le n} \sum_{i=1}^{n} |a_{ij}|$$

$$\text{Frobenius norm} \qquad \|A\|_f = \sqrt{\sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij}^2}$$

$$\text{row-sum norm} \qquad \|A\|_\infty = \max_{1 \le i \le n} \sum_{j=1}^{n} |a_{ij}|$$

$$\text{spectral norm (2 norm)} \qquad \|A\|_2 = (\mu_{\max})^{1/2}$$

- **Note:** $\mu_{\max}$ is the largest eigenvalue of $[A]^T[A]$

Matrix Condition Number

- The ***matrix condition number*** Cond[A] is obtained by calculating Cond[A]=$||A||\cdot||A^{-1}||$
- In can be shown that:

Ralston & Rabinowitz, 1978

$$\frac{\|\Delta X\|}{\|X\|} \le \mathrm{Cond}[A]\frac{\|\Delta A\|}{\|A\|} \qquad \text{given that } [A]\{x\}=\{b\}$$

- The relative error of the norm of the computed solution can be as large as the relative error of the norm of the coefficients of [A] multiplied by the condition number

- If the coefficients of [A] are known to t digit precision (rounding errors are on the order of 10^{-t}), the solution [X] may be valid to only t-$\log_{10}$(Cond[A]) digits
  - If the conditional number is much greater than 1, it is suggested that the system is prone to being ill-conditioned

MATLAB Commands (1/3)

- MATLAB has built-in functions to compute both norms and condition numbers:
  - `norm(`*X*`,`*p*`)`
    - Compute the *p* norm of vector *X*, where *p* can be any number, `inf`, or `'fro'` (for the **Euclidean** norm)
  - `norm(`*A*`,`*p*`)`
    - Compute a norm of matrix *A*, where *p* can be 1, 2, `inf`, or `'fro'` (for the **Frobenius** norm)
  - `cond(`*X*`,`*p*`)` or `cond(`*A*`,`*p*`)`
    - Calculate the condition number of vector *X* or matrix *A* using the norm specified by *p*

MATLAB Commands (2/3)

Example 11.4

Problem Statement. Use MATLAB to evaluate both the norms and condition numbers for the scaled Hilbert matrix previously analyzed in Example 11.3:

$$[A] = \begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{3} \\ 1 & \frac{2}{3} & \frac{1}{2} \\ 1 & \frac{3}{4} & \frac{3}{5} \end{bmatrix}$$

(*a*) As in Example 11.3, first compute the row-sum versions ($p =$ `inf`). (*b*) Also compute the Frobenius ($p =$ `'fro'`) and the spectral ($p = 2$) condition numbers.

Solution: (*a*) First, enter the matrix:

```
>> A = [1 1/2 1/3;1 2/3 1/2;1 3/4 3/5];
```

Then, the row-sum norm and condition number can be computed as

```
>> norm(A,inf)

ans =
    2.3500

>> cond(A,inf)

ans =
  451.2000
```

$$\|A\|_\infty = \max_{1 \le i \le n} \sum_{j=1}^{n} |a_{ij}|$$

MATLAB Commands (3/3)

Example 11.4

(b) The condition numbers based on the Frobenius and spectral norms are

```
>> cond(A,'fro')

ans =
   368.0866

>> cond(A)

ans =
   366.3503
```

$$\|A\|_f = \sqrt{\sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij}^2}$$

$$\|A\|_2 = \left(\mu_{\max}\right)^{1/2}$$