MATLAB 与工程应用

Linear Algebraic Equations

- Matrix Methods for Linear Equations
- The Left Division Method
- Underdetermined Systems
- Overdetermined Systems
- A General Solution Program

Matrix Methods for Linear Equations

Matrix notation enables us to represent multiple equations as a single matrix equation. For example, consider the following set.

$$2x_1 + 9x_2 = 5$$

$$3x_1 - 4x_2 = 7$$

This set can be expressed in vector-matrix form as

$$\begin{bmatrix} 2 & 9 \\ 3 & -4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 5 \\ 7 \end{bmatrix}$$

which can be represented in the following compact form

$$\mathbf{Ax} = \mathbf{b}$$

where we have defined the following matrices and vectors:

$$\mathbf{A} = \begin{bmatrix} 2 & 9 \\ 3 & -4 \end{bmatrix} \qquad \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \qquad \mathbf{b} = \begin{bmatrix} 5 \\ 7 \end{bmatrix}$$

$$\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$$

SINGULAR MATRIX

The inverse of a matrix A is defined only if A is square and nonsingular. A matrix is singular if its determinant $|A|$ is zero. If A is singular, then a unique solution does not exist. The MATLAB functions inv(A) and det(A) compute the inverse the determinant of the matrix A. If the inv(A) function is applied to a singular matrix, MATLAB will issue a warning to that effect.

ILL-CONDITIONED SET

An *ill-conditioned* set of equations is a set that is close to being singular. The ill-conditioned status depends on the accuracy with which the solution calculations are made. When internal numerical accuracy used by MATLAB is insuf - cient to obtain a solution, it prints the message warning that the matrix is close to singular and that the results might be inaccurate.

The Left Division Method

MATLAB provides the *left division method* for solving the equation set $\mathbf{Ax} = \mathbf{b}$. This method is based on Gauss elimination. To use the left division method to solve for **x**, you type `x = A\b`. If $|\mathbf{A}| = 0$ or if the number of equations does not equal the number of unknowns, then you need to use the other methods to be presented later.

Left Division Method with Three Unknowns

Use the left division method to solve the following set.

$$\begin{aligned} 3x_1 + 2x_2 - 9x_3 &= -65 \\ -9x_1 - 5x_2 + 2x_3 &= 16 \\ 6x_1 + 7x_2 + 3x_3 &= 5 \end{aligned}$$

The Matrix Inverse Method

Solve the following equations, using the matrix inverse.

$$2x_1 + 9x_2 = 5$$
$$3x_1 - 4x_2 = 7$$

■ Solution

The matrix **A** and the vector **b** are

$$\mathbf{A} = \begin{bmatrix} 2 & 9 \\ 3 & -4 \end{bmatrix} \qquad \mathbf{b} = \begin{bmatrix} 5 \\ 7 \end{bmatrix}$$

The session is

```
>>A = [2,9;3,-4]; b = [5;7];
>>x = inv(A)*b
x =
    2.3714
    0.0286
```

The solution is $x_1 = 2.3714$ and $x_2 = 0.0286$. MATLAB did not issue a warning, so the solution is unique.

The solution form $\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$ is rarely applied in practice to obtain numerical solutions to sets of many equations, because calculation of the matrix inverse is likely to introduce greater numerical inaccuracy than the left division method to be introduced.

Matrix Rank

The matrix inverse method will warn us if a *unique* solution does not exist, but it does not tell us whether there is no solution or an in nite number of solutions. In addition, the method is limited to cases where the matrix **A** is square, that is, cases where the number of equations equals the number of unknowns. For this reason we now introduce a method that allows us to determine easily whether an equation set has a solution and whether it is unique. The method requires the concept of the *rank* of a matrix.

Definition of Matrix Rank

An $m \times n$ matrix A has a rank $r \geq 1$ if and only if $|A|$ contains a nonzero $r \times r$ determinant and every square subdeterminant with $r + 1$ or more rows is zero

Existence and Uniqueness of Solutions. The set $Ax = b$ with m equations and n unknowns has solutions if and only if (1) $rank(A) = rank([A\ b])$. Let $r = rank(A)$. If condition (1) is satisfied and if $r = n$, then the solution is unique. If condition (1) is satisfied but $r < n$, there are an infinite number of solutions, and r unknown variables can be expressed as linear combinations of the other $n - r$ unknown variables, whose values are arbitrary.

Homogeneous case. The homogeneous set $\mathbf{Ax} = \mathbf{0}$ is a special case in which $\mathbf{b} = \mathbf{0}$. For this case, rank($\mathbf{A}$) = rank([$\mathbf{A}$ $\mathbf{b}$]) always, and thus the set always has the trivial solution $\mathbf{x} = \mathbf{0}$. A nonzero solution, in which at least one unknown is nonzero, exists if and only if rank($\mathbf{A}$) < n. If $m < n$, the homogeneous set always has a nonzero solution.

This test implies that if $\mathbf{A}$ is square and of dimension $n \times n$, then rank([$\mathbf{A}$ $\mathbf{b}$]) = rank($\mathbf{A}$), and a unique solution exists for any $\mathbf{b}$ if rank($\mathbf{A}$) = n.

■ Solution

The matrices **A** and **b** are

$$\mathbf{A} = \begin{bmatrix} 3 & 2 & -9 \\ -9 & -5 & 2 \\ 6 & 7 & 3 \end{bmatrix} \qquad \mathbf{b} = \begin{bmatrix} -65 \\ 16 \\ 5 \end{bmatrix}$$

The session is

```
>>A = [3,2,-9;-9,-5,2;6,7,3];
>>rank(A)
ans =
   3
```

Because **A** is 3×3 and rank(**A**) = 3, which is the number of unknowns, a unique solution exists. It is obtained by continuing the session as follows.

```
>>b = [-65;16;5];
>>x = A\b
x =
    2.0000
   -4.0000
    7.0000
```

This answer gives the vector **x,** which corresponds to the solution $x_1 = 2,\ x_2 = -4,\ x_3 = 7$.

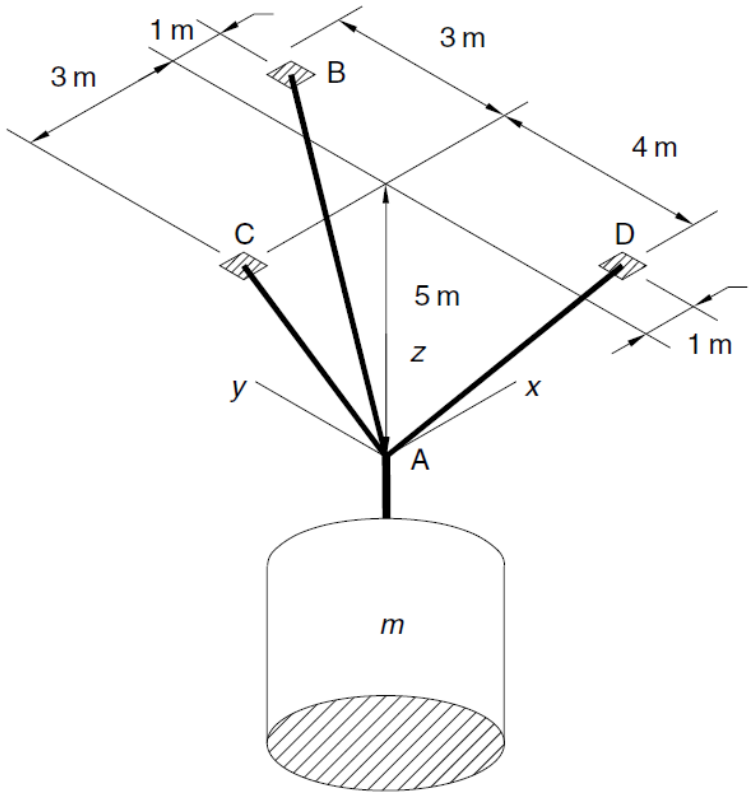
Calculation of Cable Tension

A mass m is suspended by three cables attached at three points B, C, and D, as shown in the Figure. Let T_1, T_2, and T_3 be the tensions in the three cables AB, AC, and AD, respectively. If the mass m is stationary, the sum of the tension components in the x, in the y, and in the z directions must each be zero. This gives the following three equations:

$$\frac{T_1}{\sqrt{35}} - \frac{3T_2}{\sqrt{34}} + \frac{T_3}{\sqrt{42}} = 0$$

$$\frac{3T_1}{\sqrt{35}} - \frac{4T_3}{\sqrt{42}} = 0$$

$$\frac{5T_1}{\sqrt{35}} + \frac{5T_2}{\sqrt{34}} + \frac{5T_3}{\sqrt{42}} - mg = 0$$



A mass suspended by three cables

Determine T_1, T_2, and T_3 in terms of an unspecified value of the weight mg.

■ **Solution**

If we set $mg = 1$, the equations have the form $\mathbf{AT} = \mathbf{b}$ where

$$\mathbf{A} = \begin{bmatrix} \frac{1}{\sqrt{35}} & -\frac{3}{\sqrt{34}} & \frac{1}{\sqrt{42}} \\ \frac{3}{\sqrt{35}} & 0 & -\frac{4}{\sqrt{42}} \\ \frac{5}{\sqrt{35}} & \frac{5}{\sqrt{34}} & \frac{5}{\sqrt{42}} \end{bmatrix} \qquad \mathbf{T} = \begin{bmatrix} T_1 \\ T_2 \\ T_3 \end{bmatrix} \qquad \mathbf{b} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

The script file to solve this system is

```
% File cable.m
s34 = sqrt(34); s35 = sqrt(35); s42 = sqrt(42);
A1 = [1/s35, -3/s34, 1/s42];
A2 = [3/s35, 0, -4/s42];
A3 = [5/s35, 5/s34, 5/s42];
A = [A1; A2; A3];
b = [0; 0; 1];
rank(A)
rank([A, b])
T = A\b
```

When this file is executed by typing *cable*, we find that $rank(A) = rank([A\ b] = 3$ and obtain the values $T_1 = 0.5071, T_2 = 0.2915, and\ T_3 = 0.4166$. Because A is 3×3 and $rank(A) = 3$, which is the number of unknowns, the solution is unique. Using the linearity property, we multiply these results by mg and obtain the general solution $T_1 = 0.5071mg$, $T_2 = 0.2915\ mg$, and $T_3 = 0.4166\ mg$.

Underdetermined Systems

An underdetermined system does not contain enough information to determine all the unknown variables, usually but not always because it has fewer equations than unknowns. Thus an infinite number of solutions can exist, with one or more of unknowns dependent on the remaining unknowns. **The left division method works for square and nonsquare A matrices.** However, if **A** is not square, the left division method can give answers that might be misinterpreted. We will show how to interpret MATLAB results correctly.

When there are fewer equations than unknowns, the left division method might give a solution with some of the unknowns set equal to zero, but this is not the general solution. An infinite number of solutions might exist even when the number of equations equals the number of unknowns. This can occur when $|A| = 0$. For such systems the left division method generates an error message warning us that the matrix **A** is singular.

PSEUDOINVERSE METHOD

In such cases the pseudoinverse method $x = pinv(A) * b$ gives one solution, *the minimum nor solution*. In cases where there are an infinite number of solutions, the **rref** function can be used to express some of the unknowns in terms of the remaining unknowns, whose values are arbitrary.

An Underdetermined Set with Three Equations and Three Unknowns

Show that the following set does not have a unique solution. How many of the unknowns will be undetermined? Interpret the results given by the left division method.

$$\begin{aligned} 2x_1 - 4x_2 + 5x_3 &= -4 \\ -4x_1 - 2x_2 + 3x_3 &= 4 \\ 2x_1 + 6x_2 - 8x_3 &= 0 \end{aligned}$$

■ Solution

A MATLAB session to check the ranks is

```
>>A = [2,-4,5;-4,-2,3;2,6,-8];
>>b = [-4;4;0];
>>rank(A)
ans =
  2
>>rank([A, b])
ans =
  2
>>x = A\b
Warning: Matrix is singular to working precision.
ans =
   NaN
   NaN
   NaN
```

Because the ranks of **A** and [**A b**] are equal, a solution exists. However, because the number of unknowns is 3 and is 1 greater than the rank of **A,** one of the unknowns will be undetermined. An in nite number of solutions exist, and we can solve for only two of the unknowns in terms of the third unknown. The set is underdetermined because there are fewer than three independent equations; the third equation can be obtained from the rst two. To see this, add the rst and second equations, to obtain $-2x_1 - 6x_2 + 8x_3 = 0$, which is equivalent to the third equation.

Note that we could also tell that the matrix **A** is singular because its rank is less than 3. If we use the left division method, MATLAB returns a message warning that the problem is singular, and it does not produce an answer.

The `pinv` Function and the Euclidean Norm

The `pinv` function (which stands for "pseudoinverse") can be used to obtain a solution of an underdetermined set. To solve the equation set $\mathbf{Ax} = \mathbf{b}$ using the `pinv` function, you type `x = pinv(A)*b`. The `pinv` function gives a solution that gives the minimum value of the *Euclidean norm*, which is the magnitude of the solution vector `x`. The magnitude of a vector $\mathbf{v}$ in three-dimensional space, having components x, y, z, is $\sqrt{x^2 + y^2 + z^2}$. It can be computed using matrix multiplication and the transpose as follows.

$$\sqrt{\mathbf{v}^T\mathbf{v}} = \sqrt{[x\ y\ z]^T \begin{bmatrix} x \\ y \\ z \end{bmatrix}} = \sqrt{x^2 + y^2 + z^2}$$

The generalization of this formula to an n-dimensional vector $\mathbf{v}$ gives the magnitude of the vector and is the Euclidean norm N. Thus

$$N = \sqrt{\mathbf{v}^T\mathbf{v}}$$

The MATLAB function `norm(v)` computes the Euclidean norm.

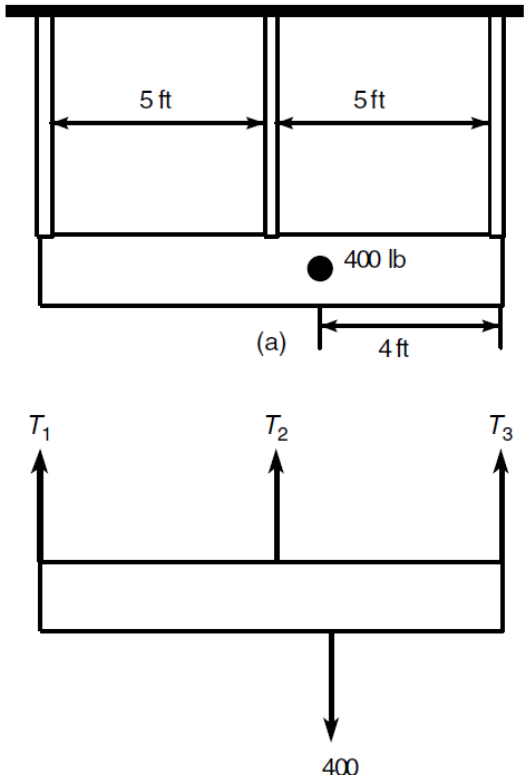
A Statically Indeterminate Problem

Determine the forces in the three equally spaced supports that hold up a light xture. The supports are 5 ft apart. The xture weighs 400 lb, and its mass center is 4 ft from the right end. Obtain the solution using the MATLAB left-division method and the pseudoinverse method.

$$T_1 + T_2 + T_3 - 400 = 0$$
$$400(4) - 10T_1 - 5T_2 = 0$$

or

$$T_1 + T_2 + T_3 = 400$$
$$10T_1 + 5T_2 + 0T_3 = 1600$$



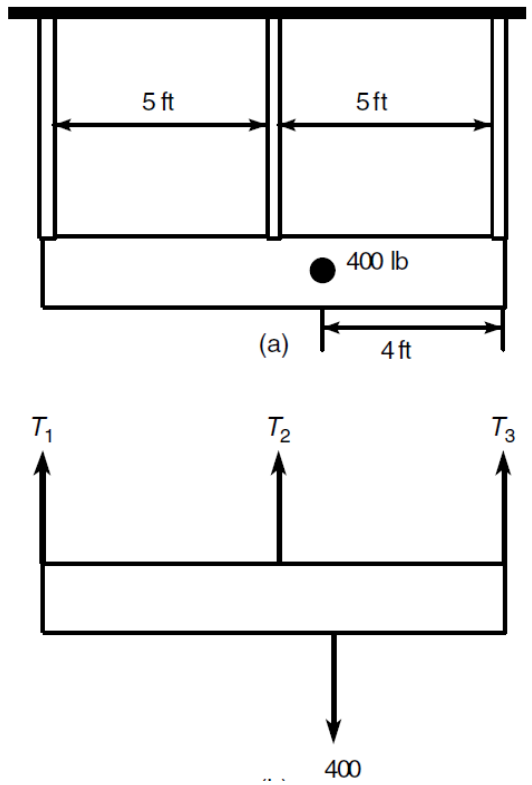
A light fixture and its free-body diagram

Because there are more unknowns than equations, the set is underdetermined. Thus we cannot determine a unique set of values for the forces. Such a problem, when the equations of statics do not give enough equations, is called *statically indeterminate*. These equations can be written in the matrix form **AT** = **b** as follows:

$$\begin{bmatrix} 1 & 1 & 1 \\ 10 & 5 & 0 \end{bmatrix} \begin{bmatrix} T_1 \\ T_2 \\ T_3 \end{bmatrix} = \begin{bmatrix} 400 \\ 1600 \end{bmatrix}$$

The MATLAB session is

```
>>A = [1,1,1;10,5,0];
>>b = [400;1600];
>>rank(A)
ans =
  2
>>rank([A, b])
ans =
  2
>>T = A\b
T =
   160.0000
   0
   240.0000
```



```
>>T = pinv(A)*b
T =
    93.3333
    133.3333
    173.3333
```

The left division answer corresponds to $T_1 = 160$, $T_2 = 0$, and $T_3 = 240$. This illustrates how the MATLAB left division operator produces a solution with one or more variables set to zero, for underdetermined sets having more unknowns than equations.

Because the ranks of **A** and [**A b**] are both 2, a solution exists, but it is not unique. Because the number of unknowns is 3, and is 1 greater than the rank of **A,** an in nite number of solutions exist, and we can solve for only two of the unknowns in terms of the third.

The pseudoinverse solution gives $T_1 = 93.3333$, $T_2 = 133.3333$, and $T_3 = 173.3333$. This is the minimum norm solution for real values of the variables. The minimum norm solution consists of the real values of T_1, T_2, and T_3 that minimize

$$N = \sqrt{T_1^2 + T_2^2 + T_3^2}$$

To understand what MATLAB is doing, note that we can solve Equations (8.3–2) and (8.3–3) to obtain T_1 and T_2 in terms of T_3 as $T_1 = T_3 - 80$ and $T_2 = 480 - 2T_3$. Then the Euclidean norm can be expressed as

$$N = \sqrt{(T_3 - 80)^2 + (480 - 2T_3)^2 + T_3^2} = \sqrt{6T_3^2 - 2080T_3 + 236{,}800}$$

The real value of T_3 that minimizes N can be found by plotting N versus T_3, or by using calculus. The answer is $T_3 = 173.3333$, the same as the minimum norm solution given by the pseudoinverse method.

When there are infinite number of solutions, we must decide whether the solutions given by the left division and the pseudoinverse methods are useful for applications. This must be done in the context of the specific application.

Find two solutions to the following set.

$$x_1 + 3x_2 + 2x_3 = 2$$

$$x_1 + x_2 + x_3 = 4$$

(Answer: Minimum norm solution: $x_1 = 4.33, x_2 = -1.67, x_3 = 1.34$.
Left division solution: $x_1 = 5, x_2 = -1, x_3 = 0$.)

The Reduced Row Echelon Form

We can express some of the unknowns in an underdetermined set as functions of the remaining unknowns. For equations

$$T_1 - T_3 = -80 \qquad T_2 + 2T_3 = 480$$

In matrix form these are

$$\begin{bmatrix} 1 & 0 & -1 \\ 0 & 1 & 2 \end{bmatrix} \begin{bmatrix} T_1 \\ T_2 \\ T_3 \end{bmatrix} = \begin{bmatrix} -80 \\ 480 \end{bmatrix}$$

The augmented matrix [**A b**] for the above set is

$$\begin{bmatrix} 1 & 0 & -1 & -80 \\ 0 & 1 & 2 & 480 \end{bmatrix}$$

We can always reduce an underdetermined set to such a form by multiplying the set's equations by suitable factors and adding the resulting equations to eliminate an unknown variable. The MATLAB `rref` function provides a procedure for reducing an equation set to this form, which is called the *reduced row echelon form*. Its syntax is `rref([A b])`. Its output is the augmented matrix [**C d**] that corresponds to the equation set **Cx** = **d.** This set is in reduced row echelon form.

The following underdetermined equation set has infinite number of solutions. Use the **rref** function to obtain the solutions

$$\begin{aligned} 2x_1 - 4x_2 + 5x_3 &= -4 \\ -4x_1 - 2x_2 + 3x_3 &= 4 \\ 2x_1 + 6x_2 - 8x_3 &= 0 \end{aligned}$$

■ Solution

The MATLAB session is

```
>>A = [2,-4,5;-4,-2,3;2,6,-8];
>>b = [-4;4;0];
>>rref([A, b])
ans =
   1     0    -0.1   -1.2000
   0     1    -1.3    0.4000
   0     0       0         0
```

The answer corresponds to the augmented matrix [**C** **d**], where

$$[\mathbf{C} \quad \mathbf{d}] = \begin{bmatrix} 1 & 0 & -0.1 & -1.2 \\ 0 & 1 & -1.3 & 0.4 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Supplementing Underdetermined Systems

Often the linear equations describing the application are underdetermined because not enough information has been specified to determine unique values of the unknowns. In such cases we might be able to include additional information, objectives, or constraints to find a unique solution

Production Planning

The following table shows how many hours reactors A and B need to produce 1 ton each of the chemical products 1, 2, and 3. The two reactors are available for 40 and 30 hr per week, respectively. Determine how many tons of each product can be produced each week.

Hours	Product 1	Product 2	Product 3
Reactor A	5	3	3
Reactor B	3	3	4

■ Solution

Let x, y, and z be the number of tons each of products 1, 2, and 3 that can be produced in one week. Using the data for reactor A, the equation for its usage in one week is

$$5x + 3y + 3z = 40$$

The data for reactor B gives

$$3x + 3y + 4z = 30$$

This system is underdetermined. The matrices for the equation $\mathbf{Ax} = \mathbf{b}$ are

$$\mathbf{A} = \begin{bmatrix} 5 & 3 & 3 \\ 3 & 3 & 4 \end{bmatrix} \qquad \mathbf{b} = \begin{bmatrix} 40 \\ 30 \end{bmatrix} \qquad \mathbf{x} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

Here the rank($\mathbf{A}$) = rank([$\mathbf{A}$ $\mathbf{b}$]) = 2, which is less than the number of unknowns. Thus an in nite number of solutions exist, and we can determine two of the variables in terms of the third.

Using the `rref` command `rref([A b])`, where `A = [5,3,3;3,3,4]` and `b = [40;30]`, we obtain the following reduced echelon augmented matrix:

$$\begin{bmatrix} 1 & 0 & -0.5 & 5 \\ 0 & 1 & 1.8333 & 5 \end{bmatrix}$$

This matrix gives the reduced system

$$x - 0.5z = 5$$
$$y + 1.8333z = 5$$

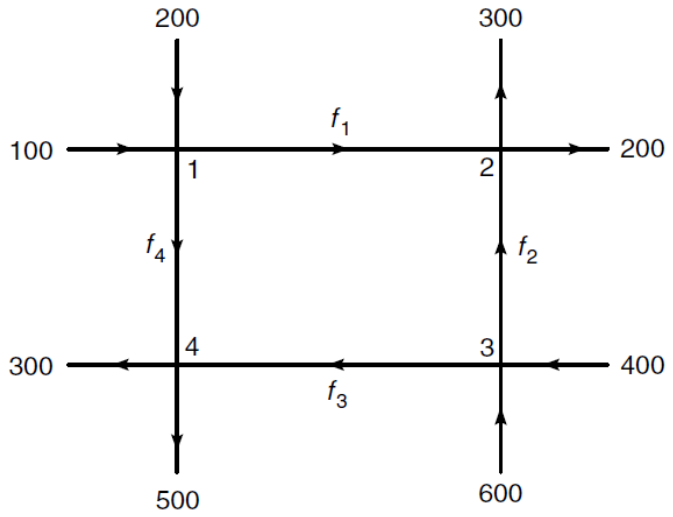
which can be easily solved as follows:

$$x = 5 + 0.5z$$
$$y = 5 - 1.8333z$$

Where z is arbitrary. However, z cannot be completely arbitrary if the solution is to be meaningful. For example, negative values of the variables have no meaning here; thus we require that $x \gg 0, y \geq 0, and\ z \geq 0$. From the solutions, to get meaning result, $0 \leq z \leq 2.737$ tons. The choice of z within this range must be made some other basis, such as product

Traffic Engineering

A traffic engineer wants to know if measurements of traffic flow entering and leaving a road network are sufficient to predict the traffic flow on each street in the network. For example, consider the network of one-way streets. The numbers shown are the measured traffic flows in the vehicles per hour. Assume that no vehicles park anywhere within the network. If possible, calculate the traffic flows $f_1, f_2, and\ f_4$. If this is not possible, suggest how to obtain the necessary information.

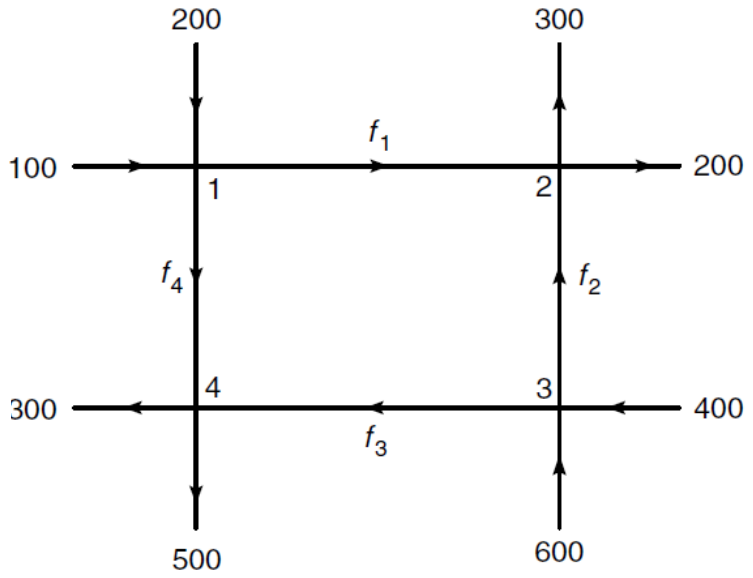


A network of one-way streets

■ Solution

The flow into intersection 1 must equal the flow out of the intersection, This gives

$$100 + 200 = f_1 + f_4$$



Similarly, for the other three intersections, we have

$$f_1 + f_2 = 300 + 200$$
$$600 + 400 = f_2 + f_3$$
$$f_3 + f_4 = 300 + 500$$

Putting these in the matrix form $\mathbf{Ax} = \mathbf{b}$, we obtain

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} 300 \\ 500 \\ 1000 \\ 800 \end{bmatrix} \quad \mathbf{x} = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{bmatrix}$$

First, check the ranks of **A** and [**A b**], using the MATLAB `rank` function. Both have a rank of 3, which is 1 less than the number of unknowns, so we can determine three of the unknowns in terms of the fourth. Thus we cannot determine all the traf c ows based on the given measurements.

Using the `rref([A b])` function produces the reduced augmented matrix

$$\begin{bmatrix} 1 & 0 & 0 & 1 & 300 \\ 0 & 1 & 0 & -1 & 200 \\ 0 & 0 & 1 & 1 & 800 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

which corresponds to the reduced system

$$f_1 + f_4 = 300$$

$$f_2 - f_4 = 200$$

$$f_3 + f_4 = 800$$

These can be solved easily as follows: $f_1 = 300 - f_4, f_2 = 200 + f_4$, and $f_3 = 800 - f_4$. If we could measure the ow on one of the internal roads, say f_4, then we could compute the other ows. So we recommend that the engineer arrange to have this additional measurement made.

Practice

Use the `rref`, `pinv`, and the left division methods to solve the following set.

$$
\begin{aligned}
3x_1 + 5x_2 + 6x_3 &= 6 \\
8x_1 - x_2 + 2x_3 &= 1 \\
5x_1 - 6x_2 - 4x_3 &= -5
\end{aligned}
$$

(Answer: There are an in nite number of solutions. The result obtained with the `rref` function is $x_1 = 0.2558 - 0.3721x_3$, $x_2 = 1.0465 - 0.9767x_3$, x_3 arbitrary. The `pinv` function gives $x_1 = 0.0571$, $x_2 = 0.5249$, $x_3 = 0.5340$. The left division method generates an error message.)

Practice

Use the `rref`, `pinv`, and left division methods to solve the following set.

$$3x_1 + 5x_2 + 6x_3 = 4$$

$$x_1 - 2x_2 - 3x_3 = 10$$

(Answer: There are an in nite number of solutions. The result obtained with the `rref` function is $x_1 = 0.2727x_3 + 5.2727$, $x_2 = -1.3636x_3 - 2.2626$, x_3 arbitrary. The solution obtained with left division is $x_1 = 4.8000$, $x_2 = 0$, $x_3 = -1.7333$. The one obtained with the pseudoinverse method is $x_1 = 4.8394$, $x_2 = -0.1972$, $x_3 = -1.5887$.)

Overdetermined Systems

An *overdetermined system* is a set of equations that has more independent equations than unknowns. Some overdetermined systems have exact solutions, and they can be obtained with the left division method `x = A\b`. For other overdetermined systems, no exact solution exists; in some of these cases, the left division method does not yield an answer, while in other cases the left division method gives an answer that satis es the equation set only in a "least-squares" sense. We will show what this means in the next example. When MATLAB gives an answer to an overdetermined set, it does not tell us whether the answer is the exact solution. We must determine this information ourselves, and we will now show how to do this.

The Least-Squares Method

Suppose we have the following three data points, and we want to nd the straight line $y = c_1x + c_2$ that best ts the data in some sense.

x	y
0	2
5	6
10	11

(a) Find the coef cients c_1 and c_2 using the least-squares criterion. (b) Find the coef - cients by using the left division method to solve the three equations (one for each data point) for the two unknowns c_1 and c_2. Compare with the answer from part (a).

■ Solution

LEAST-SQUARES METHOD

(a) Because two points de ne a straight line, unless we are extremely lucky , our three data points will not lie on the same straight line. A common criterion for obtaining the straight line that best ts the data is the *least-squares* criterion. According to this criterion, the line that minimizes J, the sum of the squares of the vertical differences between the line and the data points, is the "best" t. Here J is

$$J = \sum_{i=1}^{i=3} (c_1 x_i + c_2 - y_i)^2 = (0c_1 + c_2 - 2)^2 + (5c_1 + c_2 - 6)^2 + (10c_1 + c_2 - 11)^2$$

If you are familiar with calculus, you know that the values of c_1 and c_2 that minimize J are found by setting the partial derivatives $\partial J/\partial c_1$ and $\partial J/\partial c_2$ equal to zero.

$$\frac{\partial J}{\partial c_1} = 250c_1 + 30c_2 - 280 = 0$$

$$\frac{\partial J}{\partial c_2} = 30c_1 + 6c_2 - 38 = 0$$

The solution is $c_1 = 0.9$ and $c_2 = 11/6$. The best straight line in the least-squares sense is $y = 0.9x + 11/6$.

(b) Evaluating the equation $y = c_1x + c_2$ at each data point gives the following three equations, which are overdetermined because there are more equations than unknowns.

$$\begin{aligned} 0c_1 + c_2 &= 2 \\ 5c_1 + c_2 &= 6 \\ 10c_1 + c_2 &= 11 \end{aligned}$$

These equations can be written in the matrix form $\mathbf{Ax} = \mathbf{b}$ as follows.

$$\mathbf{Ax} = \begin{bmatrix} 0 & 1 \\ 5 & 0 \\ 10 & 1 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} = \begin{bmatrix} 2 \\ 6 \\ 11 \end{bmatrix} = \mathbf{b}$$

where

$$[\mathbf{A}\ \mathbf{b}] = \begin{bmatrix} 0 & 1 & 2 \\ 5 & 1 & 6 \\ 10 & 1 & 11 \end{bmatrix}$$

To use left division, the MATLAB session is

```
>>A = [0,1;5,1;10,1];
>>b = [2;6;11];
>>rank(A)
ans =
  2
>>rank([A, b])
ans =
  3
>>x = A\b
x =
   0.9000
   1.8333
>>A*x
ans =
  1.833
  6.333
  10.8333
```

This result for $\mathbf{x}$ agrees with the least-squares solution obtained previously: $c_1 = 0.9$, $c_2 = 11/6 = 1.8333$. The rank of $\mathbf{A}$ is 2, but the rank of $[\mathbf{A}\ \mathbf{b}]$ is 3, so no exact solution exists for c_1 and c_2. Note that `A*x` gives the y values generated by the line $y = 0.9x + 1.8333$ at the x data values $x = 0, 5, 10$. These are different from the right-hand sides of the original three equations. This is not unexpected, because the least-squares solution is not an exact solution of the equations.

Some overdetermined systems have an exact solution. The left division method sometimes gives an answer for overdetermined systems, but it does not indicate whether the answer is the exact solution. We need to check the ranks of $\mathbf{A}$ and $[\mathbf{A}\ \mathbf{b}]$ to know if the answer is the exact solution. The next example illustrates this situation.

An Overdetermined Set

Solve the following equations and discuss the solution for two cases: $c = 9$ and $c = 10$.

$$x_1 + x_2 = 1$$
$$x_1 + 2x_2 = 3$$
$$x_1 + 5x_2 = c$$

■ Solution

The coef cient matrix and the augmented matrix for this problem are

$$\mathbf{A} = \begin{bmatrix} 1 & 1 \\ 1 & 2 \\ 1 & 5 \end{bmatrix} \qquad [\mathbf{A}\ \mathbf{b}] = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 3 \\ 1 & 5 & c \end{bmatrix}$$

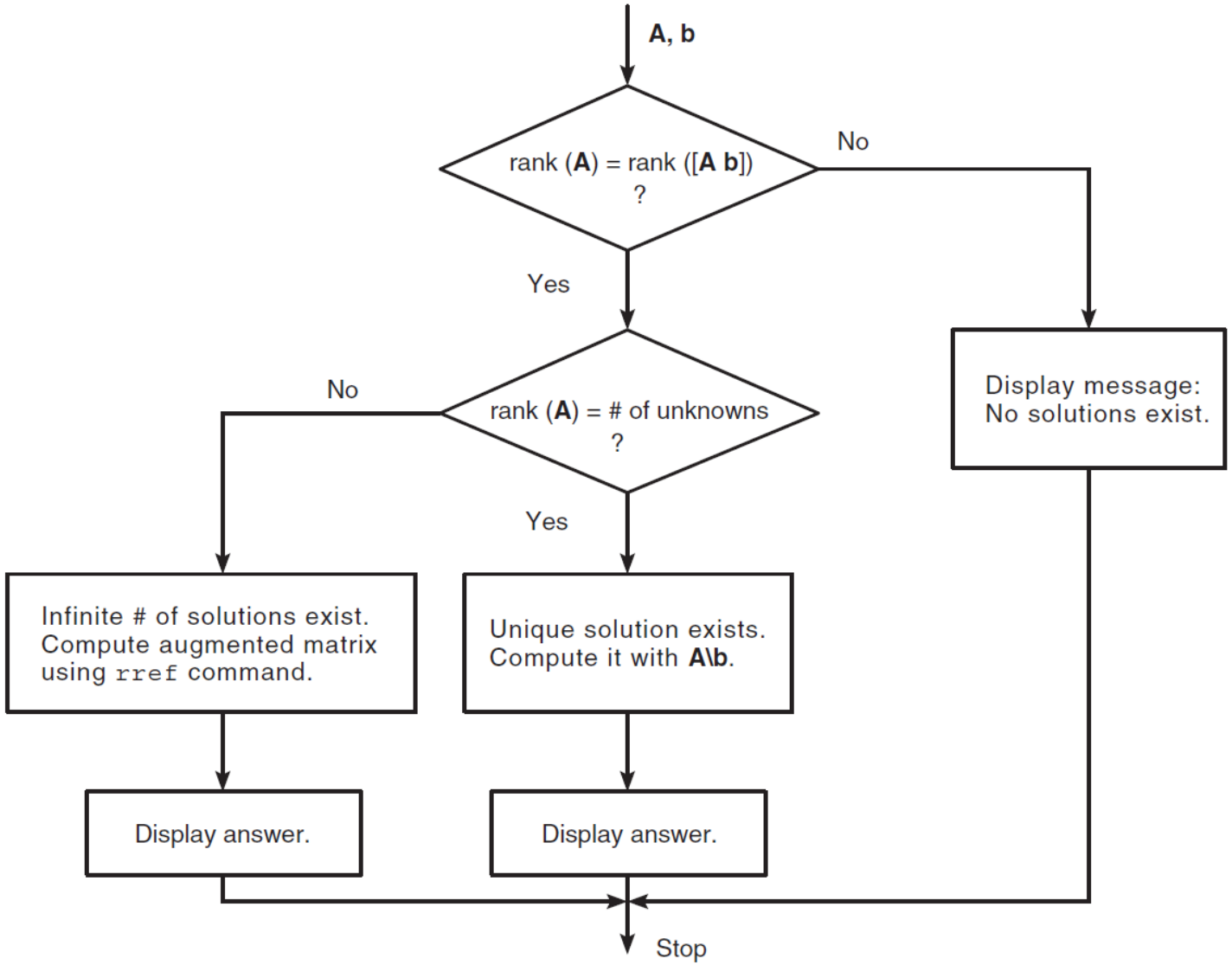
Making the computations in MATLAB, we nd that for $c = 9$, rank $(\mathbf{A})$ = rank$([\mathbf{A}\ \mathbf{b}]) = 2$. Thus the system has a solution, and because the number of unknowns (2) equals the rank of **A,** there is a unique solution. The left division method `A\b` gives this solution, which is $x_1 = -1$ and $x_2 = 2$.

For $c = 10$ we nd that rank$(\mathbf{A}) = 2$, but rank$([\mathbf{A}\ \mathbf{b}]) = 3$. Because rank$(\mathbf{A}) \neq$ rank$([\mathbf{A}\ \mathbf{b}])$, there is no solution. However, the left division method **A\b** gives $x_1 = -1.3846$ and $x_2 = 2.2692$, which is *not* an exact solution! This can be veri ed by substituting these values into the original equation set. This answer is the solution to the equation set in a least-squares sense. That is, these values are the values of x_1 and x_2 that minimize J, the sum of the squares of the differences between the equations' left- and right-hand sides.

$$J = (x_1 + x_2 - 1)^2 + (x_1 + 2x_2 - 3)^2 + (x_1 + 5x_2 - 10)^2$$

To interpret MATLAB answers correctly for an overdetermined system, first check the ranks of A and [A b] to see if an exact solution exists; if one does not exist, then we know that the left division answer is a least-squares solution.

A General Solution Program



Flowchart illustrating a program to solve linear equations

For underdetermined sets, MATLAB provides three ways of dealing with the equation set **Ax = b** (note that the matrix inverse method will never work with such sets):

1. The matrix left division method (which gives one speci c solution, but not the general solution).
2. The pseudoinverse method. Solve for `x` by typing `x = pinv(A)*b`. This gives the minimum norm solution.
3. The reduced row echelon form (RREF) method. This method uses the MATLAB command `rref` to obtain a general solution for some of the unknowns in terms of the other unknowns.

Matrix functions and commands for solving linear equations

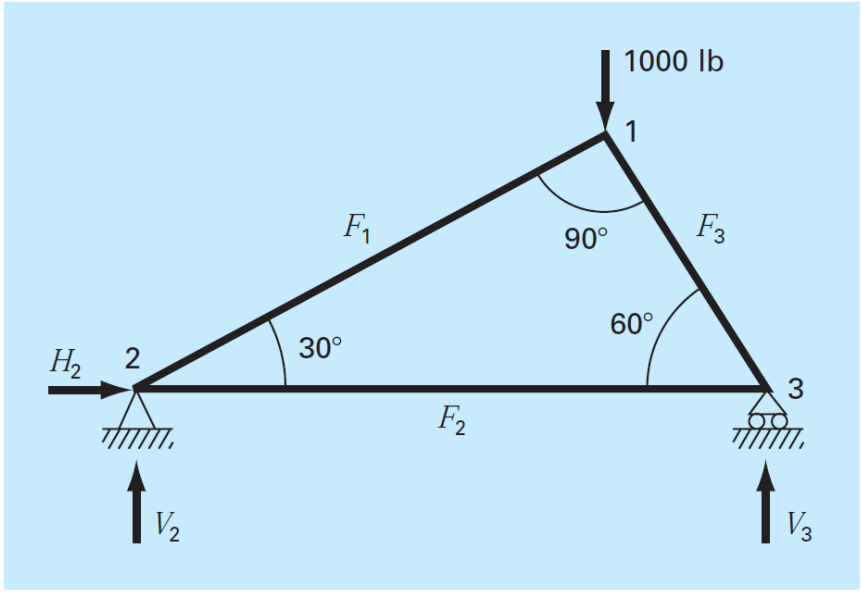
Function	Description
`det(A)`	Computes the determinant of the array **A.**
`inv(A)`	Computes the inverse of the matrix **A.**
`pinv(A)`	Computes the pseudoinverse of the matrix **A.**
`rank(A)`	Computes the rank of the matrix **A.**
`rref([A b])`	Computes the reduced row echelon form corresponding to the augmented matrix [**A b**].
`x = inv(A)*b`	Solves the matrix equation **Ax = b** using the matrix inverse.
`x = A\b`	Solves the matrix equation **Ax = b** using left division.

Case Studies: Linear Algebraic Equations

ANALYSIS OF A STATICALLY DETERMINATE TRUSS (CIVIL/ENVIRONMENTAL ENGINEERING)

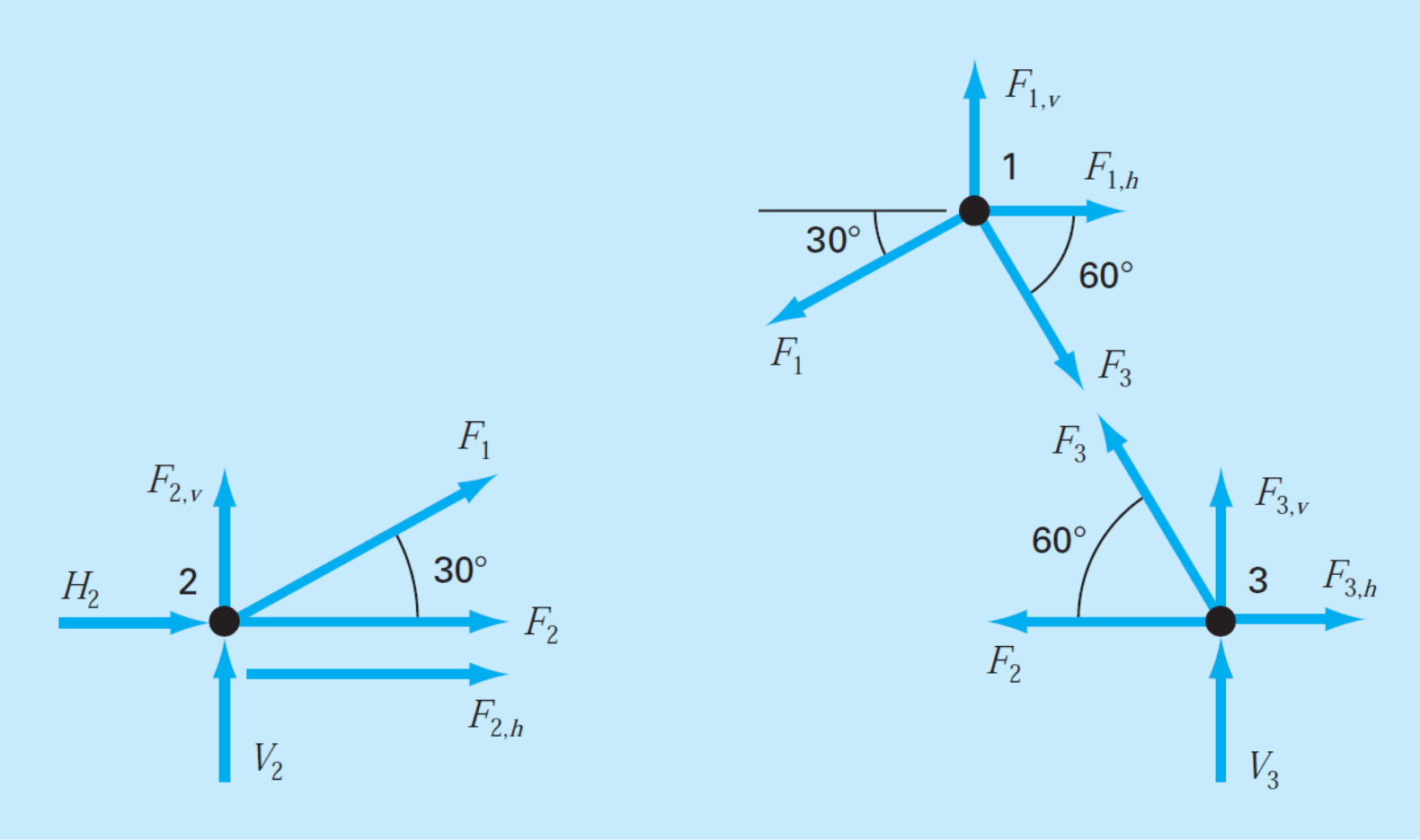
Background. An important problem in structural engineering is that of finding the forces and reactions associated with a statically determinate truss. Figure 12.4 shows an example of such a truss.

The forces (F) represent either tension or compression on the members of the truss. External reactions (H_2, V_2, and V_3) are forces that characterize how the truss interacts with the supporting surface. The hinge at node 2 can transmit both horizontal and vertical forces to the surface, whereas the roller at node 3 transmits only vertical forces. It is observed that the effect of the external loading of 1000 lb is distributed among the various members of the truss.



Forces on a statically determinate truss.

Free-body force diagrams for the nodes of a statically determinate truss



Solution. This type of structure can be described as a system of coupled linear algebraic equations. Free-body force diagrams are shown for each node in Fig. 12.5. The sum of the forces in both horizontal and vertical directions must be zero at each node, because the system is at rest. Therefore, for node 1,

$$\Sigma F_H = 0 = -F_1 \cos 30^\circ + F_3 \cos 60^\circ + F_{1,h} \tag{12.3}$$

$$\Sigma F_V = 0 = -F_1 \sin 30^\circ - F_3 \sin 60^\circ + F_{1,v} \tag{12.4}$$

for node 2,

$$\Sigma F_H = 0 = F_2 + F_1 \cos 30^\circ + F_{2,h} + H_2 \tag{12.5}$$

$$\Sigma F_V = 0 = F_1 \sin 30^\circ + F_{2,v} + V_2 \tag{12.6}$$

for node 3,

$$\Sigma F_H = 0 = -F_2 - F_3 \cos 60^\circ + F_{3,h} \tag{12.7}$$

$$\Sigma F_V = 0 = F_3 \sin 60^\circ + F_{3,v} + V_3 \tag{12.8}$$

where $F_{i,h}$ is the external horizontal force applied to node i (where a positive force is from left to right) and $F_{1,v}$ is the external vertical force applied to node i (where a positive force is upward). Thus, in this problem, the 1000-lb downward force on node 1 corresponds to $F_{1,v} = -1000$. For this case all other $F_{i,v}$'s and $F_{i,h}$'s are zero. Note that the directions of the internal forces and reactions are unknown. Proper application of Newton's laws requires only consistent assumptions regarding direction. Solutions are negative if the directions are assumed incorrectly. Also note that in this problem, the forces in all members are assumed to be in tension and act to pull adjoining nodes together. A negative solution therefore corresponds to compression. This problem can be written as the following system of six equations and six unknowns:

$$\begin{bmatrix} 0.866 & 0 & -0.5 & 0 & 0 & 0 \\ 0.5 & 0 & 0.866 & 0 & 0 & 0 \\ -0.866 & -1 & 0 & -1 & 0 & 0 \\ -0.5 & 0 & 0 & 0 & -1 & 0 \\ 0 & 1 & 0.5 & 0 & 0 & 0 \\ 0 & 0 & -0.866 & 0 & 0 & -1 \end{bmatrix} \begin{Bmatrix} F_1 \\ F_2 \\ F_3 \\ H_2 \\ V_2 \\ V_3 \end{Bmatrix} = \begin{Bmatrix} 0 \\ -1000 \\ 0 \\ 0 \\ 0 \\ 0 \end{Bmatrix} \tag{12.9}$$

Notice that, as formulated in Eq. (12.9), partial pivoting is required to avoid division by zero diagonal elements. Employing a pivot strategy, the system can be solved using any of the elimination techniques discussed in Chap. 9 or 10. However, because this problem is an ideal case study for demonstrating the utility of the matrix inverse, the LU decomposi-

$$F_1 = -500 \qquad F_2 = 433 \qquad F_3 = -866$$
$$H_2 = 0 \qquad V_2 = 250 \qquad V_3 = 750$$

and the matrix inverse is

$$[A]^{-1} = \begin{bmatrix} 0.866 & 0.5 & 0 & 0 & 0 & 0 \\ 0.25 & -0.433 & 0 & 0 & 1 & 0 \\ -0.5 & 0.866 & 0 & 0 & 0 & 0 \\ -1 & 0 & -1 & 0 & -1 & 0 \\ -0.433 & -0.25 & 0 & -1 & 0 & 0 \\ 0.433 & -0.75 & 0 & 0 & 0 & -1 \end{bmatrix}$$

Now, realize that the right-hand-side vector represents the externally applied horizontal and vertical forces on each node, as in

$$\{F\}^T = \lfloor F_{1,h} \quad F_{1,v} \quad F_{2,h} \quad F_{2,v} \quad F_{3,h} \quad F_{3,v} \rfloor \tag{12.10}$$

Because the external forces have no effect on the LU decomposition, the method need not be implemented over and over again to study the effect of different external forces on the truss. Rather, all that we have to do is perform the forward- and backward-substitution steps for each right-hand-side vector to efficiently obtain alternative solutions. For example,

SPRING-MASS SYSTEMS (MECHANICAL/AEROSPACE ENGINEERING)

Background. Idealized spring-mass systems play an important role in mechanical and other engineering problems. Figure 12.11 shows such a system. After they are released, the masses are pulled downward by the force of gravity. Notice that the resulting displacement of each spring in Fig. 12.11*b* is measured along local coordinates referenced to its initial position in Fig. 12.11*a*.

As introduced in Chap. 1, Newton's second law can be employed in conjunction with force balances to develop a mathematical model of the system. For each mass, the second law can be expressed as

$$m\frac{d^2x}{dt^2} = F_D - F_U \tag{12.13}$$

To simplify the analysis, we will assume that all the springs are identical and follow Hooke's law. A free-body diagram for the first mass is depicted in Fig. 12.12*a*. The upward force is merely a direct expression of Hooke's law:

$$F_U = kx_1 \tag{12.14}$$

The downward component consists of the two spring forces along with the action of gravity on the mass,

$$F_D = k(x_2 - x_1) + k(x_2 - x_1) = m_1 g \tag{12.15}$$

Note how the force component of the two springs is proportional to the displacement of the second mass, x_2, corrected for the displacement of the first mass, x_1.

Equations (12.14) and (12.15) can be substituted into Eq. (12.13) to give

$$m_1 \frac{d^2 x_1}{dt^2} = 2k(x_2 - x_1) + m_1 g - k x_1 \tag{12.16}$$

Thus, we have derived a second-order ordinary differential equation to describe the displacement of the first mass with respect to time. However, notice that the solution cannot be obtained because the model includes a second dependent variable, x_2. Consequently, free-body diagrams must be developed for the second and the third masses (Fig. 12.12*b* and *c*)

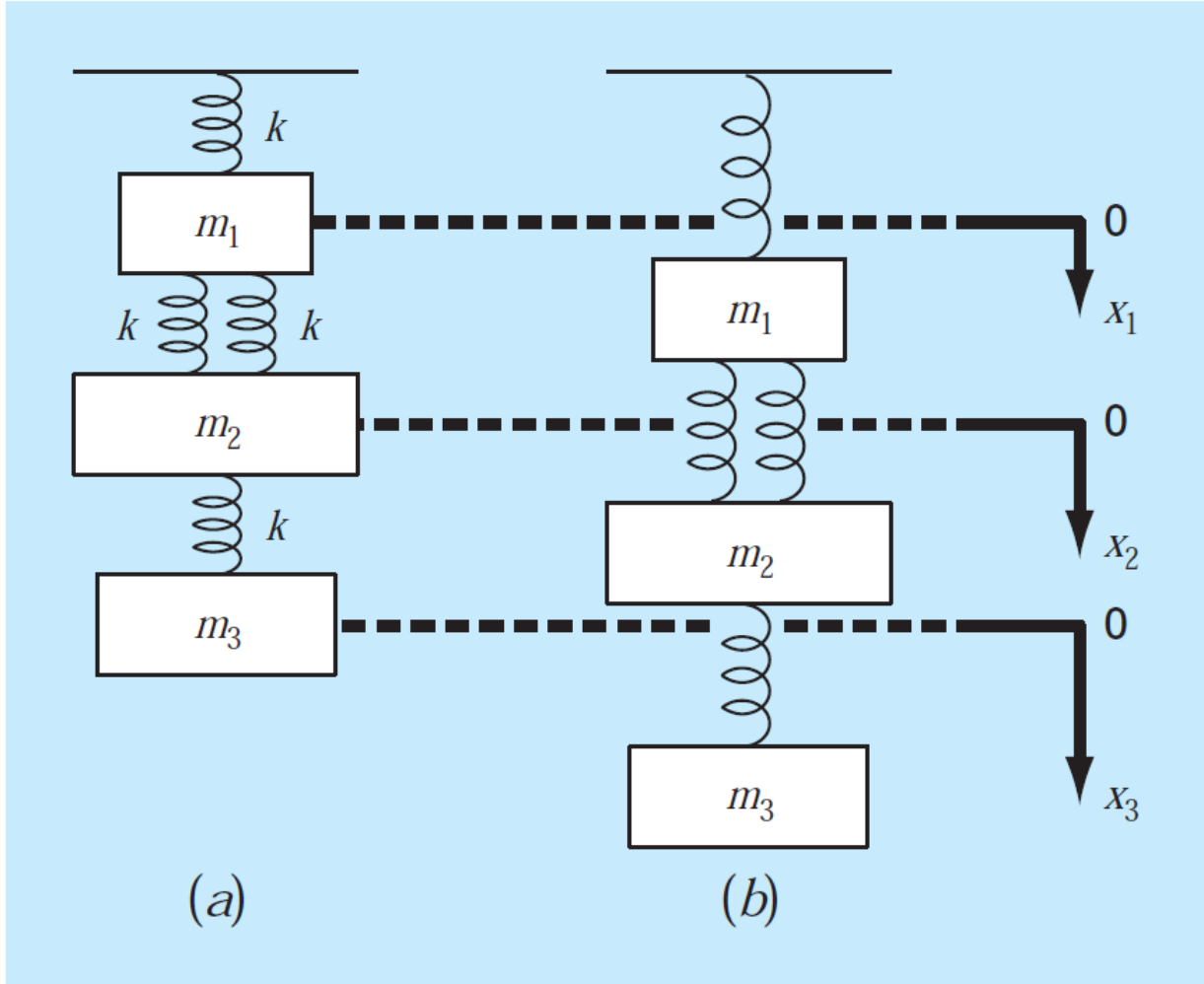


FIGURE 12.11

A system composed of three masses suspended vertically by a series of springs. (*a*) The system before release, that is, prior to extension or compression of the springs. (*b*) The system after release. Note that the positions of the masses are referenced to local coordinates with origins at their position before release.

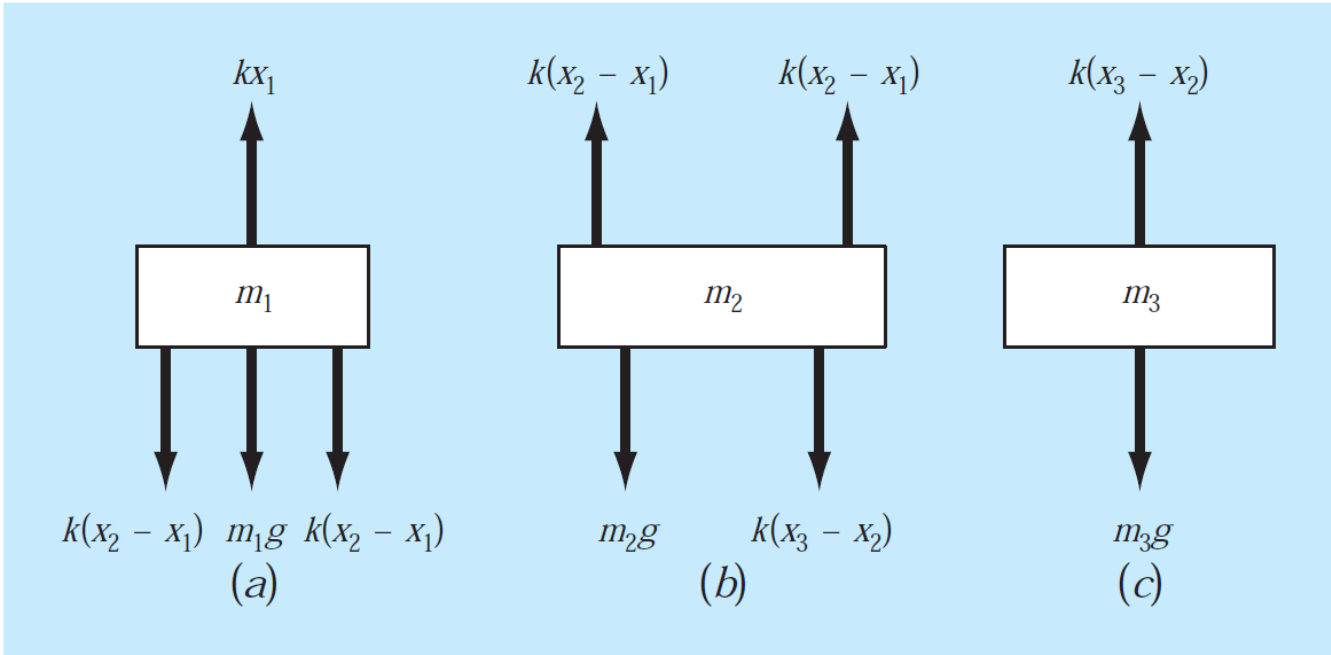


FIGURE 12.12
Free-body diagrams for the three masses from Fig. 12.11.

that can be employed to derive

$$m_2 \frac{d^2 x_2}{dt^2} = k(x_3 - x_2) + m_2 g - 2k(x_2 - x_1) \tag{12.17}$$

and

$$m_3 \frac{d^2 x_3}{dt^2} = m_3 g - k(x_3 - x_2) \tag{12.18}$$

Equations (12.16), (12.17), and (12.18) form a system of three differential equations with three unknowns. With the appropriate initial conditions, they could be used to solve for the displacements of the masses as a function of time (that is, their oscillations). We will discuss numerical methods for obtaining such solutions in Part Seven. For the present, we can obtain the displacements that occur when the system eventually comes to rest, that is, to the steady state. To do this, the derivatives in Eqs. (12.16), (12.17), and (12.18) are set to zero to give

$$\begin{aligned} 3kx_1 &- 2kx_2 &&&= m_1 g \\ -2kx_1 &+ 3kx_2 &- kx_3 &= m_2 g \\ &- kx_2 &+ kx_3 &= m_3 g \end{aligned}$$

or, in matrix form,

$$[K]\{X\} = \{W\}$$

where $[K]$, called the *stiffness matrix,* is

$$[K] = \begin{bmatrix} 3k & -2k & \\ -2k & 3k & -k \\ & -k & k \end{bmatrix}$$

and $\{X\}$ and $\{W\}$ are the column vectors of the unknowns X and the weights mg, respectively.

Solution. At this point, numerical methods can be employed to obtain a solution. If $m_1 = 2$ kg, $m_2 = 3$ kg, $m_3 = 2.5$ kg, and the k's $= 10$ kg/s^2, use LU decomposition to solve for the displacements and generate the inverse of $[K]$.

Substituting the model parameters gives

$$[K] = \begin{bmatrix} 30 & -20 & \\ -20 & 30 & -10 \\ & -10 & 10 \end{bmatrix} \qquad \{W\} = \begin{Bmatrix} 19.6 \\ 29.4 \\ 24.5 \end{Bmatrix}$$

LU decomposition can be employed to solve for $x_1 = 7.35$, $x_2 = 10.045$, and $x_3 = 12.495$. These displacements were used to construct Fig. 12.11b. The inverse of the stiffness matrix is computed as

$$[K]^{-1} = \begin{bmatrix} 0.1 & 0.1 & 0.1 \\ 0.1 & 0.15 & 0.15 \\ 0.1 & 0.15 & 0.25 \end{bmatrix}$$

Each element of this matrix k_{ji}^{-1} tells us the displacement of mass i due to a unit force imposed on mass j. Thus, the values of 0.1 in column 1 tell us that a downward unit load to the first mass will displace all of the masses 0.1 m downward. The other elements can be interpreted in a similar fashion. Therefore, the inverse of the stiffness matrix provides a fundamental summary of how the system's components respond to externally applied forces.

TABLE PT3.2 Comparison of the characteristics of alternative methods for finding solutions of simultaneous linear algebraic equations.

Method	Stability	Precision	Breadth of Application	Programming Effort	Comments
Graphical	—	Poor	Limited	—	May take more time than the numerical method, but can be useful for visualization
Cramer's rule	—	Affected by round-off error	Limited	—	Excessive computational effort required for more than three equations
Gauss elimination (with partial pivoting)	—	Affected by round-off error	General	Moderate	
LU decomposition	—	Affected by round-off error	General	Moderate	Preferred elimination method; allows computation of matrix inverse
Gauss-Seidel	May not converge if not diagonally dominant	Excellent	Appropriate only for diagonally dominant systems	Easy	

Although elimination methods have great utility, their use of the entire matrix of coefficients can be somewhat limiting when dealing with very large, sparse systems. This is due to the fact that large portions of computer memory would be devoted to storage of meaningless zeros. For banded systems, techniques are available to implement elimination methods without having to store the entire coefficient matrix.

The approximate technique described in this book is called the Gauss-Seidel method. It differs from the exact techniques in that it employs an iterative scheme to obtain progressively closer estimates of the solution. Thus, the effect of round-off is a moot point with the Gauss-Seidel method because the iterations can be continued as long as is necessary to obtain the desired precision. In addition, versions of the Gauss-Seidel method can be developed to efficiently utilize computer storage requirements for sparse systems. Consequently, the Gauss-Seidel technique has utility for large systems of equations where storage requirements would pose significant problems for the exact techniques.

The disadvantage of the Gauss-Seidel method is that it does not always converge or sometimes converges slowly on the true solution. It is strictly reliable only for those systems that are diagonally dominant. However, relaxation methods are available that sometimes offset these disadvantages. In addition, because many sets of linear algebraic equations originating from physical systems exhibit diagonal dominance, the Gauss-Seidel method has great utility for engineering problem solving.

In summary, a variety of factors will bear on your choice of a technique for a particular problem involving linear algebraic equations. However, as outlined above, the size and sparseness of the system are particularly important factors in determining your choice.

TABLE PT3.3 Summary of important information presented in Part Three.

Method	Procedure	Potential Problems and Remedies
Gauss elimination	$\left[\begin{array}{ccc\|c} a_{11} & a_{12} & a_{13} & c_1 \\ a_{21} & a_{22} & a_{23} & c_2 \\ a_{31} & a_{32} & a_{33} & c_3 \end{array}\right] \Rightarrow \left[\begin{array}{ccc\|c} a_{11} & a_{12} & a_{13} & c_1 \\ & a'_{22} & a'_{23} & c'_2 \\ & & a''_{33} & c''_3 \end{array}\right] \Rightarrow \begin{array}{l} x_3 = c''_3/a''_{33} \\ x_2 = (c'_2 - a'_{23}x_3)/a'_{22} \\ x_1 = (c_1 - a_{12}x_1 - a_{13}x_3)/a_{11} \end{array}$	Problems: Ill conditioning Round-off Division by zero Remedies: Higher precision Partial pivoting
LU decomposition	Decomposition; Back Substitution $\left[\begin{array}{ccc} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{array}\right] \Rightarrow \left[\begin{array}{ccc} 1 & 0 & 0 \\ l_{21} & 1 & 0 \\ l_{31} & l_{32} & 1 \end{array}\right] \left\{\begin{array}{c} d_1 \\ d_2 \\ d_3 \end{array}\right\} = \left\{\begin{array}{c} c_1 \\ c_2 \\ c_3 \end{array}\right\} \Rightarrow \left[\begin{array}{ccc} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{array}\right] \left\{\begin{array}{c} x_1 \\ x_2 \\ x_3 \end{array}\right\} = \left\{\begin{array}{c} d_1 \\ d_2 \\ d_3 \end{array}\right\} \Rightarrow \left\{\begin{array}{c} x_1 \\ x_2 \\ x_3 \end{array}\right\}$ Forward Substitution	Problems: Ill conditioning Round-off Division by zero Remedies: Higher precision Partial pivoting
Gauss-Seidel method	$\left.\begin{array}{l} x_1^i = (c_1 - a_{12}x_2^{i-1} - a_{13}x_3^{i-1})/a_{11} \\ x_2^i = (c_2 - a_{21}x_1^i - a_{23}x_3^{i-1})/a_{22} \\ x_3^i = (c_3 - a_{31}x_1^i - a_{32}x_2^i)/a_{33} \end{array}\right\}$ continue iteratively until $\left\|\frac{x_i^i - x_i^{i-1}}{x_i^i}\right\| 100\% < \epsilon_s$ for all x_i's	Problems: Divergent or converges slowly Remedies: Diagonal dominance Relaxation