Gauss Elimination

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

Reference:

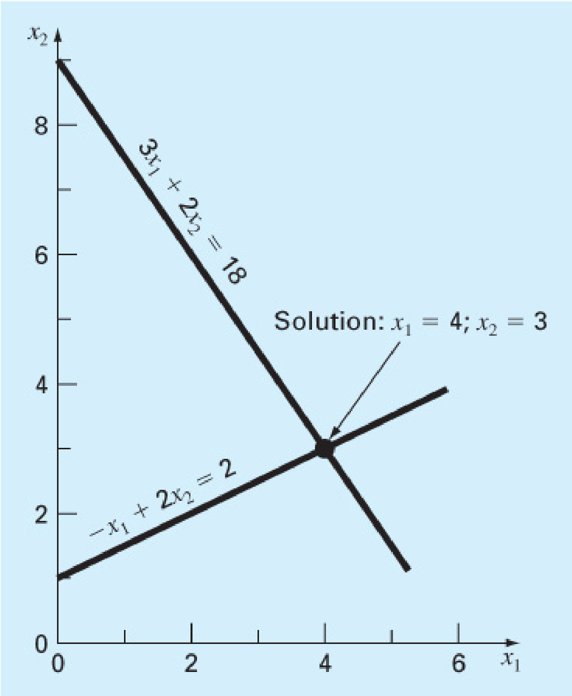
1. *Applied Numerical Methods with MATLAB for Engineers*, Chapter 9 & Teaching material

Chapter Objectives

- Knowing how to solve small sets of linear equations with the graphical method and Cramer's rule
- Understanding how to implement forward elimination and back substitution as in Gauss elimination
- Understanding how to count flops to evaluate the efficiency of an algorithm
- Understanding the concepts of singularity and ill-condition
- Understanding how partial pivoting is implemented and how it differs from complete pivoting
- Recognizing how the banded structure of a tridiagonal system can be exploited to obtain extremely efficient solutions

Graphical Method (1/2)

- For small sets of simultaneous equations, graphing them and determining the location of the intercept provides a solution



A graphical solution is obtainable for two linear equations by plotting them on Cartesian coordinates with one axis corresponding to x_1 and the other to x_2. Because the equations are linear, each equation will plot as a straight line. For example, suppose that we have the following equations:

$$3x_1 + 2x_2 = 18$$
$$-x_1 + 2x_2 = 2$$

If we assume that x_1 is the abscissa, we can solve each of these equations for x_2:

$$x_2 = -\frac{3}{2}x_1 + 9$$

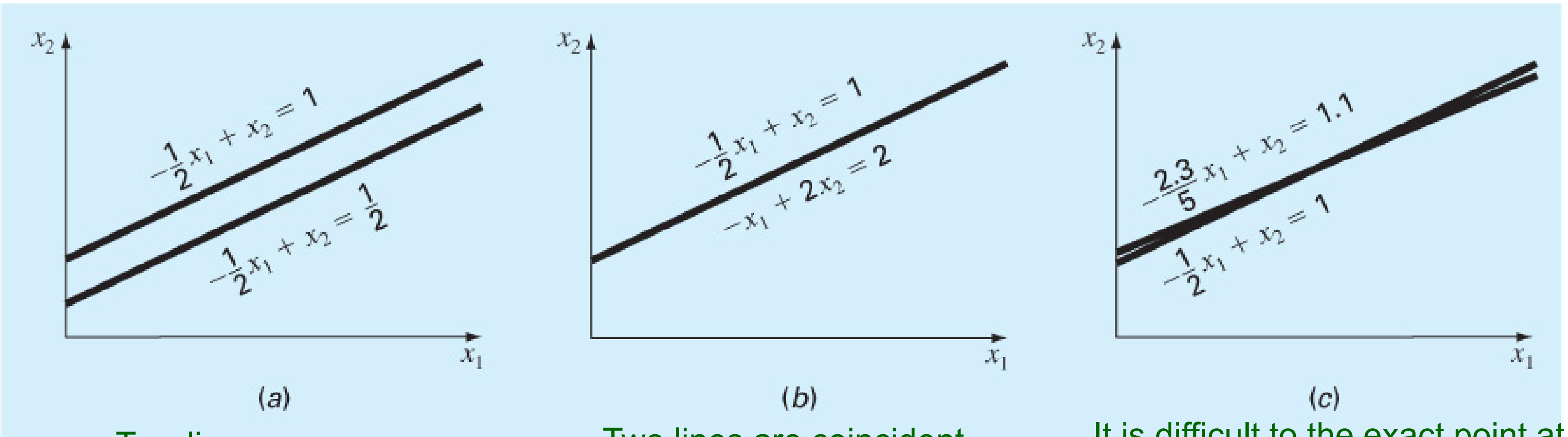
$$x_2 = \frac{1}{2}x_1 + 1$$

The equations are now in the form of straight lines—that is, $x_2 = (\text{slope})\, x_1 + \text{intercept}$. When these equations are graphed, the values of x_1 and x_2 at the intersection of the lines represent the solution (Fig. 9.1). For this case, the solution is $x_1 = 4$ and $x_2 = 3$.

FIGURE 9.1
Graphical solution of a set of two simultaneous linear algebraic equations. The intersection of the lines represents the solution.

Graphical Method (2/2)

- Graphing the equations can also show systems where:
  a) No solution exists
  b) Infinite solutions exist
  } ***singular***
  c) System is ***ill-conditioned*** (sensitive to round-off errors)



Two lines never cross. Two lines are coincident. It is difficult to the exact point at which the lines intersect.

FIGURE 9.2
Graphical depiction of singular and ill-conditioned systems: (*a*) no solution, (*b*) infinite solutions, and (*c*) ill-conditioned system where the slopes are so close that the point of intersection is difficult to detect visually.

Determinants (1/2)

- The ***determinant*** *D*=|*A*| of a matrix is formed from the coefficients of [A]
- Determinants for small matrices are:

1×1	$\lvert a_{11} \rvert = a_{11}$
2×2	$\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} = a_{11}a_{22} - a_{12}a_{21}$
3×3	$\begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} = a_{11}\underbrace{\begin{vmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{vmatrix}}_{\text{minor}} - a_{12}\underbrace{\begin{vmatrix} a_{21} & a_{23} \\ a_{31} & a_{33} \end{vmatrix}}_{\text{minor}} + a_{13}\underbrace{\begin{vmatrix} a_{21} & a_{22} \\ a_{31} & a_{32} \end{vmatrix}}_{\text{minor}}$

- Determinants for matrices larger than 3 x 3 can be very complicated

Determinants (2/2)

Problem Statement. Compute values for the determinants of the systems represented in Figs. 9.1 and 9.2.

Solution. For Fig. 9.1:

$$D = \begin{vmatrix} 3 & 2 \\ -1 & 2 \end{vmatrix} = 3(2) - 2(-1) = 8$$

For Fig. 9.2*a*:

$$D = \begin{vmatrix} -\frac{1}{2} & 1 \\ -\frac{1}{2} & 1 \end{vmatrix} = -\frac{1}{2}(1) - 1\left(\frac{-1}{2}\right) = 0$$

For Fig. 9.2*b*:

$$D = \begin{vmatrix} -\frac{1}{2} & 1 \\ -1 & 2 \end{vmatrix} = -\frac{1}{2}(2) - 1(-1) = 0$$

For Fig. 9.2*c*:

$$D = \begin{vmatrix} -\frac{1}{2} & 1 \\ -\frac{2.3}{5} & 1 \end{vmatrix} = -\frac{1}{2}(1) - 1\left(\frac{-2.3}{5}\right) = -0.04$$

In the foregoing example, the singular systems had zero determinants. Additionally, the results suggest that the system that is almost singular (Fig. 9.2*c*) has a determinant that is close to zero. These ideas will be pursued further in our subsequent discussion of ill-conditioning in Chap. 11.

Cramer's Rule

- ***Cramer's Rule*** states that each unknown in a system of linear algebraic equations may be expressed as a fraction of two determinants with denominator *D* and with the numerator obtained from *D* by replacing the column of coefficients of the unknown in question by the constants b_1, b_2, …, b_n
- For a set of three equations, x_1 will be computed as

$$AX = B$$

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

$$x_1 = \frac{\begin{vmatrix} b_1 & a_{12} & a_{13} \\ b_2 & a_{22} & a_{23} \\ b_3 & a_{32} & a_{33} \end{vmatrix}}{D}$$

Cramer's Rule: An Example

- Find x_2 in the following system of equations:

$$\begin{aligned} 0.3x_1 + 0.52x_2 + \quad x_3 &= -0.01 \\ 0.5x_1 + \quad x_2 + 1.9x_3 &= \quad 0.67 \\ 0.1x_1 + 0.3\ \ x_2 + 0.5x_3 &= -0.44 \end{aligned}$$

- Find the determinant D

$$D = \begin{vmatrix} 0.3 & 0.52 & 1 \\ 0.5 & 1 & 1.9 \\ 0.1 & 0.3 & 0.5 \end{vmatrix} = 0.3\begin{vmatrix} 1 & 1.9 \\ 0.3 & 0.5 \end{vmatrix} - 0.52\begin{vmatrix} 0.5 & 1.9 \\ 0.1 & 0.5 \end{vmatrix} + 1\begin{vmatrix} 0.5 & 1 \\ 0.1 & 0.4 \end{vmatrix} = -0.0022$$

- Find determinant D_2 by replacing D's second column with b

$$D_2 = \begin{vmatrix} 0.3 & -0.01 & 1 \\ 0.5 & 0.67 & 1.9 \\ 0.1 & -0.44 & 0.5 \end{vmatrix} = 0.3\begin{vmatrix} 0.67 & 1.9 \\ -0.44 & 0.5 \end{vmatrix} - 0.01\begin{vmatrix} 0.5 & 1.9 \\ 0.1 & 0.5 \end{vmatrix} + 1\begin{vmatrix} 0.5 & 0.67 \\ 0.1 & -0.44 \end{vmatrix} = 0.0649$$

- Divide

$$x_2 = \frac{D_2}{D} = \frac{0.0649}{-0.0022} = -29.5$$

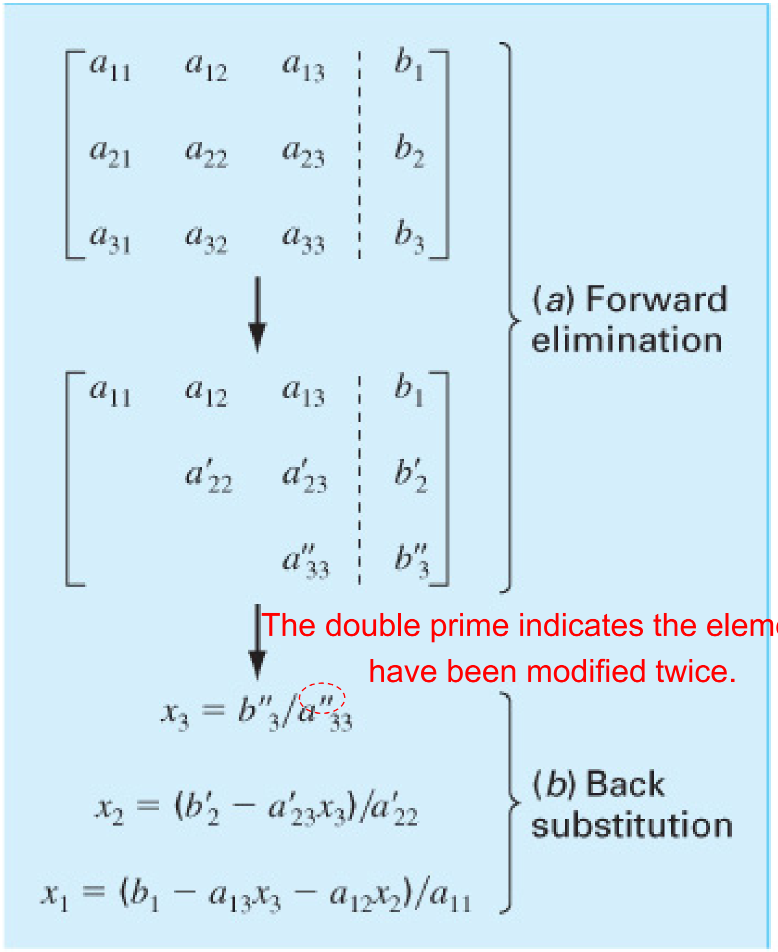
More on Cramer’s Rule

- For more than three equations, Cramer’s rule becomes impractical because, as the number of equations increases, the determinants are time consuming to evaluate by hand (or by computer)

Naïve Gauss Elimination (1/4)

- For larger systems, ***Cramer's Rule*** can become impractical (unwieldy)
- Instead, a sequential process of removing unknowns from equations using *forward elimination* followed by *back substitution* may be used - this is ***Gauss elimination***
- "**Naïve**" Gauss elimination simply means the process does not check for potential problems resulting from division by zero

Naïve Gauss Elimination (2/4)



The double prime indicates the elements have been modified twice.

FIGURE 9.3
The two phases of Gauss elimination: (*a*) forward elimination and (*b*) back substitution.

- **Forward elimination**
  - Starting with the first row, add or subtract multiples of that row to eliminate the first coefficient from the second row and beyond
  - Continue this process with the second row to remove the second coefficient from the third row and beyond
  - Stop when an upper triangular matrix remains
- **Back substitution**
  - Starting with the ***last*** row, solve for the unknown, then substitute that value into the next highest row
  - Because of the upper-triangular nature of the matrix, each row will contain only one more unknown

Naïve Gauss Elimination (3/4)

- Pivot Equations and Pivot Elements

pivot element — pivot equation

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \cdots + a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + \cdots + a_{2n}x_n &= b_2 \\ &\vdots \\ a_{n1}x_1 + a_{n2}x_2 + a_{n3}x_3 + \cdots + a_{nn}x_n &= b_n \end{aligned}$$

First Step of Forward Elimination

pivot element — pivot equation

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \cdots + a_{1n}x_n &= b_1 \\ a'_{22}x_2 + a'_{23}x_3 + \cdots + a'_{2n}x_n &= b'_2 \\ a'_{32}x_2 + a'_{33}x_3 + \cdots + a'_{3n}x_n &= b'_3 \\ &\vdots \\ a'_{n2}x_2 + a'_{n3}x_3 + \cdots + a'_{nn}x_n &= b'_n \end{aligned}$$

Second Step of Forward Elimination

Naïve Gauss Elimination (4/4)

Back Substitution. Equation (9.11d) can now be solved for x_n:

$$x_n = \frac{b_n^{(n-1)}}{a_{nn}^{(n-1)}} \tag{9.12}$$

This result can be back-substituted into the $(n-1)$th equation to solve for x_{n-1}. The procedure, which is repeated to evaluate the remaining x's, can be represented by the following formula:

$$x_i = \frac{b_i^{(i-1)} - \sum_{j=i+1}^{n} a_{ij}^{(i-1)} x_j}{a_{ii}^{(i-1)}} \qquad \text{for } i = n-1, n-2, \ldots, 1 \tag{9.13}$$

Naïve Gauss Elimination: An Example (1/3)

Naive Gauss Elimination

Problem Statement. Use Gauss elimination to solve

$$3x_1 - 0.1x_2 - 0.2x_3 = 7.85 \quad \text{(E9.3.1)}$$

$$0.1x_1 + 7x_2 - 0.3x_3 = -19.3 \quad \text{(E9.3.2)}$$

$$0.3x_1 - 0.2x_2 + 10x_3 = 71.4 \quad \text{(E9.3.3)}$$

Solution. The first part of the procedure is forward elimination. Multiply Eq. (E9.3.1) by 0.1/3 and subtract the result from Eq. (E9.3.2) to give

$$7.00333x_2 - 0.293333x_3 = -19.5617$$

Then multiply Eq. (E9.3.1) by 0.3/3 and subtract it from Eq. (E9.3.3). After these operations, the set of equations is

$$3x_1 - 0.1x_2 - 0.2x_3 = 7.85 \quad \text{(E9.3.4)}$$

$$7.00333x_2 - 0.293333x_3 = -19.5617 \quad \text{(E9.3.5)}$$

$$-0.190000x_2 + 10.0200x_3 = 70.6150 \quad \text{(E9.3.6)}$$

Example 9.3

Naïve Gauss Elimination: An Example (2/3)

To complete the forward elimination, x_2 must be removed from Eq. (E9.3.6). To accomplish this, multiply Eq. (E9.3.5) by $-0.190000/7.00333$ and subtract the result from Eq. (E9.3.6). This eliminates x_2 from the third equation and reduces the system to an upper triangular form, as in

$$3x_1 - 0.1x_2 - 0.2x_3 = 7.85 \tag{E9.3.7}$$

$$7.00333x_2 - 0.293333x_3 = -19.5617 \tag{E9.3.8}$$

$$10.0120x_3 = 70.0843 \tag{E9.3.9}$$

We can now solve these equations by back substitution. First, Eq. (E9.3.9) can be solved for

$$x_3 = \frac{70.0843}{10.0120} = 7.00003$$

This result can be back-substituted into Eq. (E9.3.8), which can then be solved for

$$x_2 = \frac{-19.5617 + 0.293333(7.00003)}{7.00333} = -2.50000$$

Example 9.3

Naïve Gauss Elimination: An Example (3/3)

Finally, $x_3 = 7.00003$ and $x_2 = -2.50000$ can be substituted back into Eq. (E9.3.7), which can be solved for

$$x_1 = \frac{7.85 + 0.1(-2.50000) + 0.2(7.00003)}{3} = 3.00000$$

Although there is a slight round-off error, the results are very close to the exact solution of $x_1 = 3$, $x_2 = -2.5$, and $x_3 = 7$. This can be verified by substituting the results into the original equation set:

$$3(3) - 0.1(-2.5) - 0.2(7.00003) = 7.84999 \cong 7.85$$

$$0.1(3) + 7(-2.5) - 0.3(7.00003) = -19.30000 = -19.3$$

$$0.3(3) - 0.2(-2.5) + 10(7.00003) = 71.4003 \cong 71.4$$

Example 9.3

Naïve Gauss Elimination Program

```
function x = GaussNaive(A,b)
% GaussNaive: naive Gauss elimination
%   x = GaussNaive(A,b): Gauss elimination without pivoting.
% input:
%   A = coefficient matrix
%   b = right hand side vector
% output:
%   x = solution vector

[m,n] = size(A);
if m~=n, error('Matrix A must be square'); end
nb = n+1;
Aug = [A b];
% forward elimination
for k = 1:n-1
  for i = k+1:n
    factor = Aug(i,k)/Aug(k,k);
    Aug(i,k:nb) = Aug(i,k:nb)-factor*Aug(k,k:nb);
  end
end
% back substitution
x = zeros(n,1);
x(n) = Aug(n,nb)/Aug(n,n);
for i = n-1:-1:1
  x(i) = (Aug(i,nb)-Aug(i,i+1:n)*x(i+1:n))/Aug(i,i);
end
```

FIGURE 9.4
An M-file to implement naive Gauss elimination.

Gauss Program Efficiency

- The execution of Gauss elimination depends on the amount of ***floating-point operations*** (or **flops**). The flop count for an *n* x *n* system is:

Forward Elimination	$\frac{2n^3}{3} + O(n^2)$
Back Substitution	$n^2 + O(n)$
Total	$\frac{2n^3}{3} + O(n^2)$

- Conclusions:
  - As the system gets larger, the computation time increases greatly
  - Most of the effort is incurred in the elimination step

Pivoting

- Problems arise with naïve Gauss elimination **if a coefficient along the diagonal is 0** (problem: division by 0) or close to 0 (problem: round-off error)

- One way to combat these issues is to determine the coefficient with the largest absolute value in the column below the **pivot element**. The rows can then be switched so that the largest element is the pivot element. This is called ***partial pivoting***

- If the **columns** to the right of the pivot element are also checked and columns switched, this is called ***complete pivoting***

Pivoting: An Example (1/2)

Problem Statement. Use Gauss elimination to solve

$$0.0003x_1 + 3.0000x_2 = 2.0001$$
$$1.0000x_1 + 1.0000x_2 = 1.0000$$

Note that in this form the first pivot element, $a_{11} = 0.0003$, is very close to zero. Then repeat the computation, but partial pivot by reversing the order of the equations. The exact solution is $x_1 = 1/3$ and $x_2 = 2/3$.

Solution. Multiplying the first equation by $1/(0.0003)$ yields

$$x_1 + 10{,}000x_2 = 6667$$

which can be used to eliminate x_1 from the second equation:

$$-9999x_2 = -6666$$

which can be solved for $x_2 = 2/3$. This result can be substituted back into the first equation to evaluate x_1:

$$x_1 = \frac{2.0001 - 3(2/3)}{0.0003} \tag{E9.4.1}$$

without partial pivoting

Example 9.4

Pivoting: An Example (2/2)

with partial pivoting

On the other hand, if the equations are solved in reverse order, the row with the larger pivot element is normalized. The equations are

$$1.0000x_1 + 1.0000x_2 = 1.0000$$
$$0.0003x_1 + 3.0000x_2 = 2.0001$$

Elimination and substitution again yields $x_2 = 2/3$. For different numbers of significant figures, x_1 can be computed from the first equation, as in

$$x_1 = \frac{1 - (2/3)}{1}$$

- Comparison

without partial pivoting

Significant Figures	x_2	x_1	Absolute Value of Percent Relative Error for x_1
3	0.667	−3.33	1099
4	0.6667	0.0000	100
5	0.66667	0.30000	10
6	0.666667	0.330000	1
7	0.6666667	0.3330000	0.1

with partial pivoting

Significant Figures	x_2	x_1	Absolute Value of Percent Relative Error for x_1
3	0.667	0.333	0.1
4	0.6667	0.3333	0.01
5	0.66667	0.33333	0.001
6	0.666667	0.333333	0.0001
7	0.6666667	0.3333333	0.0000

Determinant Evaluation with Gauss Elimination (1/2)

- The determinant of a triangular matrix can be simply computed as the product of its diagonal elements

$$D = \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ 0 & a_{22} & a_{23} \\ 0 & 0 & a_{33} \end{vmatrix}$$

where the determinant can be evaluated as [recall Eq. (9.1)]:

$$D = a_{11}\begin{vmatrix} a_{22} & a_{23} \\ 0 & a_{33} \end{vmatrix} - a_{12}\begin{vmatrix} 0 & a_{23} \\ 0 & a_{33} \end{vmatrix} + a_{13}\begin{vmatrix} 0 & a_{22} \\ 0 & 0 \end{vmatrix}$$

or, by evaluating the minors:

$$D = a_{11}a_{22}a_{33} - a_{12}(0) + a_{13}(0) = a_{11}a_{22}a_{33}$$

Determinant Evaluation with Gauss Elimination (1/2)

- The forward-elimination step of Gauss elimination results in an upper triangular system (matrix)
- Also, the value of the determinant is not changed by the forward-elimination process, the determinant thus can be evaluated via

$$D = a_{11}a'_{22}a''_{33}\cdots a_{nn}^{(n-1)}$$

where the superscripts signify the number of times that the elements have been modified by the elimination process. Thus, we can capitalize on the effort that has already been expended in reducing the system to triangular form and, in the bargain, come up with a simple estimate of the determinant.

There is a slight modification to the above approach when the program employs partial pivoting. For such cases, the determinant changes sign every time a row is switched. One way to represent this is by modifying the determinant calculation as in

$$D = a_{11}a'_{22}a''_{33}\cdots a_{nn}^{(n-1)}(-1)^p$$

where p represents the number of times that rows are pivoted. This modification can be incorporated simply into a program by merely keeping track of the number of pivots that take place during the course of the computation.

Tridiagonal Systems

- A *tridiagonal* system is a banded system with a bandwidth of 3:

$$\begin{bmatrix} f_1 & g_1 & & & & & \\ e_2 & f_2 & g_2 & & & & \\ & e_3 & f_3 & g_3 & & & \\ & & \cdot & \cdot & \cdot & & \\ & & & \cdot & \cdot & \cdot & \\ & & & & \cdot & \cdot & \cdot \\ & & & & e_{n-1} & f_{n-1} & g_{n-1} \\ & & & & & e_n & f_n \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \\ \cdot \\ \cdot \\ \cdot \\ x_{n-1} \\ x_n \end{Bmatrix} = \begin{Bmatrix} r_1 \\ r_2 \\ r_3 \\ \cdot \\ \cdot \\ \cdot \\ r_{n-1} \\ r_n \end{Bmatrix}$$

- Tridiagonal systems can be solved using the same method as Gauss elimination, but with much less effort because most of the matrix elements are already 0