MOTIVATIONAL EXAMPLE

The linear system:

$$\begin{array}{rcrcrcr} 6x_1 & + & 2x_2 & + & 2x_3 & = & -2 \\ 2x_1 & + & \frac{2}{3}x_2 & + & \frac{1}{3}x_3 & = & 1 \\ x_1 & + & 2x_2 & - & x_3 & = & 0 \end{array}$$

The solution:

$$x = [2.6, -3.8, -5.0]]^T$$

Method of solution: Use Gaussian elimination with rounded 4 digit decimal arithmetic.

$$\left[\begin{array}{ccc|c} 6.000 & 2.000 & 2.000 & -2.000 \\ 2.000 & .6667 & .3333 & 1.000 \\ 1.000 & 2.000 & -1.000 & 0.000 \end{array}\right] \begin{array}{l} m_{2,1} = .3333 \\ m_{3,1} = .1667 \end{array}$$

$$\left[\begin{array}{ccc|c} 6.000 & 2.000 & 2.000 & -2.000 \\ 0.000 & .0001 & -.3333 & 1.667 \\ 0.000 & 1.667 & -1.333 & .3334 \end{array}\right] \quad m_{3,2} = 16670$$

$$\left[\begin{array}{rrr|r} 6.000 & 2.000 & 2.000 & -2.000 \\ 0.000 & .0001 & -.3333 & 1.667 \\ 0.000 & 0.000 & 5555 & -27790 \end{array}\right]$$

The numerical solution:

$$x = [1.335, 0.000, -5.003]^T$$

Switch rows 2 and 3 preceding elimination of x_2 from equation #3:

$$\left[\begin{array}{rrr|r} 6.000 & 2.000 & 2.000 & -2.000 \\ 0.000 & 1.667 & -1.333 & .3334 \\ 0.000 & .0001 & -.3333 & 1.667 \end{array}\right] \quad m_{3,2} = .00005999$$

$$\left[\begin{array}{rrr|r} 6.000 & 2.000 & 2.000 & -2.000 \\ 0.000 & 1.667 & -1.333 & .3334 \\ 0.000 & 0.000 & -.3333 & 1.667 \end{array}\right]$$

The computed solution is now:

$$x = [2.602, -3.801, -5.003]^T$$

Having a nonzero pivot element is not sufficient. It also must be accurate.

PIVOTING

We want to
(a) make sure the pivot element is nonzero at each stage of the Gaussian elimination process;
(b) decrease propagation of rounding errors.

Ensuring (a) is quite simple. If the pivot element is zero, simply switch some later row for the pivot row, chosen to make sure the pivot element is nonzero. However, just choosing a nonzero pivot element is not sufficient, as the example shows. To ensure a safe choice, we do the following at each state of the process.

At stage k, recall the linear system $A^{(k)}x = b^{(k)}$:

$$\begin{bmatrix} a_{1,1}^{(1)} & a_{1,2}^{(1)} & & \cdots & & a_{1,n}^{(1)} \\ 0 & a_{2,2}^{(2)} & & \cdots & & a_{2,n}^{(2)} \\ & \ddots & \ddots & & & \vdots \\ \vdots & & 0 & a_{k,k}^{(k)} & \cdots & a_{k,n}^{(k)} \\ & & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & a_{n,k}^{(k)} & \cdots & a_{n,n}^{(k)} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \\ \vdots \\ \\ x_n \end{bmatrix} = \begin{bmatrix} b_1^{(1)} \\ b_2^{(2)} \\ \vdots \\ b_k^{(k)} \\ \vdots \\ b_n^{(k)} \end{bmatrix}$$

In rows k thru n, choose the row which has the largest leading element $a_{i,k}^{(k)}$. Say it is row i_*. Then switch equations k and i_*. This will ensure the pivot element is as far from zero as is possible. This scheme is called *partial pivoting* or *row pivoting*.

An additional benefit of such pivoting is that for the resulting multipliers

$$m_{i,k} = \frac{a_{i,k}^{(k)}}{a_{k,k}^{(k)}}, \quad i = k+1, ..., n$$

we gave

$$\left| m_{i,k} \right| \leq 1$$

For changing from $A^{(k)}$ to $A^{(k+1)}$, recall the formula

$$a_{i,j}^{(k+1)} = a_{i,j}^{(k)} - m_{i,k} a_{k,j}^{(k)}, \quad j = k+1, ..., n$$

for $i = k+1, ..., n$. Then these matrix elements cannot increase rapidly in size, because of the bound of the multipliers $m_{i,k}$. This turns out to help minimize the effects of the rounding errors which occur when doing Gaussian elimination on a computer.

COMPLETE PIVOTING

In the development of the error analysis of Gaussian elimination, it appeared that a more thorough version of pivoting was needed, one called *complete pivoting*. Recall the system

$$\begin{bmatrix} a_{1,1}^{(1)} & a_{1,2}^{(1)} & & \cdots & & a_{1,n}^{(1)} \\ 0 & a_{2,2}^{(2)} & & \cdots & & a_{2,n}^{(2)} \\ & \ddots & \ddots & & & \vdots \\ \vdots & & 0 & a_{k,k}^{(k)} & \cdots & a_{k,n}^{(k)} \\ & & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & a_{n,k}^{(k)} & \cdots & a_{n,n}^{(k)} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \\ \vdots \\ \\ x_n \end{bmatrix} = \begin{bmatrix} b_1^{(1)} \\ b_2^{(2)} \\ \vdots \\ b_k^{(k)} \\ \vdots \\ b_n^{(k)} \end{bmatrix}$$

With complete pivoting, we look at

$$\max_{k \le i,j \le n} \left| a_{i,j}^{(k)} \right|$$

and then we switch both rows and columns to have this position appear in the pivot position.

This requires a re-ordering of the unknowns, which must be undone when we are finished with the solution process. In practice, partial pivoting has been found to be sufficient; and it is much more efficient to implement.

MODIFYING THE LU FACTORIZATION

How does the use of pivoting fit in with the LU factorization of A? We begin by introducing the idea of a *permutation matrix*.

PERMUTATION MATRICES

Multiplying by a permutation matrix will permute the order of the rows or columns of a matrix. We obtain permutation matrices by re-ordering the rows (or columns) of the identity matrix I. As an example, consider

$$P = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$$

For a general matrix A of order $3 \times n$, multiply on the right by P. It will rearrange the order of the rows in A. In particular,

$$\begin{aligned} PA &= \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} a_{1,1} & \cdots & a_{1,n} \\ a_{2,1} & & a_{2,n} \\ a_{3,1} & \cdots & a_{3,n} \end{bmatrix} \\ &= \begin{bmatrix} a_{2,1} & \cdots & a_{2,n} \\ a_{3,1} & & a_{3,n} \\ a_{1,1} & \cdots & a_{1,n} \end{bmatrix} \end{aligned}$$

MODIFYING THE LU FACTORIZATION

With this, we can show that when partial pivoting is combined with Gaussian elimination, this is equivalent to the factorization result

$$LU = PA$$

for some permutation matrix P.

In words, if the equations in the system $Ax = b$ had originally been suitably ordered, then Gaussian elimination would not have been needed.

The Matlab function lu produces the matrices L, U, P. In particular, use

$$[L, U, P] = lu(A)$$

EXAMPLE

Consider

$$A = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 2 & 3 \\ -1 & -3 & 0 \end{bmatrix}$$

Using the Matlab function lu yields

$$\begin{bmatrix} 1 & 0 & 0 \\ -.5 & 1 & 0 \\ .5 & -.5 & 1 \end{bmatrix} \begin{bmatrix} 2 & 2 & 3 \\ 0 & -2 & 1.5 \\ 0 & 0 & .25 \end{bmatrix}$$

$$= \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 2 & 3 \\ -1 & -3 & 0 \end{bmatrix}$$

SCALING

In the system $Ax = b$, it has been found useful to make the rows of A of approximately 'equal size', and likewise, to make the columns of A of approximately equal size. Usually this means the maximum element in each row is of approximately equal magnitude, and similarly for the columns. This often improves the behaviour of the Gaussian elimination as it pertains to the propagation of rounding error. It is still not completely understood as to how to best scale the matrix A, but something of the kind just described is often carried out, especially for the rows of A. More details are given in the text.