Lecture 12
LU Decomposition

In many applications where linear systems appear, one needs to solve $A\mathbf{x} = \mathbf{b}$ for many different vectors $\mathbf{b}$. For instance, a structure must be tested under several different loads, not just one. As in the example of a truss (9.2), the loading in such a problem is usually represented by the vector $\mathbf{b}$. Gaussian elimination with pivoting is the most efficient and accurate way to solve a linear system. Most of the work in this method is spent on the matrix A itself. If we need to solve several different systems with the same A, and A is big, then we would like to avoid repeating the steps of Gaussian elimination on A for every different $\mathbf{b}$. This can be accomplished by the *LU decomposition*, which in effect records the steps of Gaussian elimination.

LU decomposition

The main idea of the LU decomposition is to record the steps used in Gaussian elimination on A in the places where the zero is produced. Consider the matrix

$$A = \begin{pmatrix} 1 & -2 & 3 \\ 2 & -5 & 12 \\ 0 & 2 & -10 \end{pmatrix}.$$

The first step of Gaussian elimination is to subtract 2 times the first row from the second row. In order to record what we have done, we will put the multiplier, 2, into the place it was used to make a zero, i.e. the second row, first column. In order to make it clear that it is a record of the step and not an element of A, we will put it in parentheses. This leads to

$$\begin{pmatrix} 1 & -2 & 3 \\ (2) & -1 & 6 \\ 0 & 2 & -10 \end{pmatrix}.$$

There is already a zero in the lower left corner, so we don't need to eliminate anything there. We record this fact with a (0). To eliminate the third row, second column, we need to subtract -2 times the second row from the third row. Recording the -2 in the spot it was used we have

$$\begin{pmatrix} 1 & -2 & 3 \\ (2) & -1 & 6 \\ (0) & (-2) & 2 \end{pmatrix}.$$

Let U be the upper triangular matrix produced, and let L be the lower triangular matrix with the records and ones on the diagonal, i.e.

$$L = \begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 0 & -2 & 1 \end{pmatrix} \quad \text{and} \quad U = \begin{pmatrix} 1 & -2 & 3 \\ 0 & -1 & 6 \\ 0 & 0 & 2 \end{pmatrix},$$

then we have the following wonderful property:

$$LU = \begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 0 & -2 & 1 \end{pmatrix} \begin{pmatrix} 1 & -2 & 3 \\ 0 & -1 & 6 \\ 0 & 0 & 2 \end{pmatrix} = \begin{pmatrix} 1 & -2 & 3 \\ 2 & -5 & 12 \\ 0 & 2 & -10 \end{pmatrix} = A.$$

Thus we see that A is actually the product of L and U. Here L is lower triangular and U is upper triangular. When a matrix can be written as a product of simpler matrices, we call that a *decomposition* of A and this one we call the LU decomposition.

Using LU to solve equations

If we also include pivoting, then an LU decomposition for A consists of three matrices P, L and U such that

$$PA = LU. \tag{12.1}$$

The pivot matrix P is the identity matrix, with the same rows switched as the rows of A are switched in the pivoting. For instance,

$$P = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix},$$

would be the pivot matrix if the second and third rows of A are switched by pivoting. MATLAB will produce an LU decomposition with pivoting for a matrix A with the command

```
> [L U P] = lu(A)
```

where `P` is the pivot matrix. To use this information to solve $A\mathbf{x} = \mathbf{b}$ we first pivot both sides by multiplying by the pivot matrix:

$$PA\mathbf{x} = P\mathbf{b} \equiv \mathbf{d}.$$

Substituting LU for PA we get

$$LU\mathbf{x} = \mathbf{d}.$$

Then we need only to solve two back substitution problems:

$$L\mathbf{y} = \mathbf{d}$$

and

$$U\mathbf{x} = \mathbf{y}.$$

In MATLAB this would work as follows:

```
> A = rand(5,5)
> [L U P] = lu(A)
> b = rand(5,1)
> d = P*b
> y = L\d
> x = U\y
> rnorm = norm(A*x - b)    ........................Check the result.
```

We can then solve for any other $\mathbf{b}$ without redoing the LU step. Repeat the sequence for a new right hand side: `c = randn(5,1)`; you can start at the third line. While this may not seem like a big savings, it would be if A were a large matrix from an actual application.

Exercises

12.1 Solve the systems below by hand using the LU decomposition. Pivot if appropriate. In each of the two problems, check by hand that $LU = PA$ and $A\mathbf{x} = \mathbf{b}$.

(a) $A = \begin{pmatrix} 2 & 4 \\ .5 & 4 \end{pmatrix}$, $\quad \mathbf{b} = \begin{pmatrix} 0 \\ -3 \end{pmatrix}$

(b) $A = \begin{pmatrix} 1 & 4 \\ 3 & 5 \end{pmatrix}$, $\quad \mathbf{b} = \begin{pmatrix} 3 \\ 2 \end{pmatrix}$

12.2 Finish the following MATLAB function program:

```
function [x1, e1, x2, e2] = mysolve(A,b)
% Solves linear systems using the LU decomposition with pivoting
% and also with the built-in solve function A\b.
% Inputs: A -- the matrix
%         b -- the right-hand vector
% Outputs: x1 -- the solution using the LU method
%          e1 -- the norm of the residual using the LU method
%          x2 -- the solution using the built-in method
%          e2 -- the norm of the residual using the
%                built-in method
```

Using `format long`, test the program on both random matrices (`randn(n,n)`) and Hilbert matrices (`hilb(n)`) with n large (as big as you can make it and the program still run). Print your program and summarize your observations. (Do not print any random matrices or vectors.)