Solving Upper/Lower Triangular Systems

Toby Sanders (sandertl@mailbox.sc.edu)
Department of Mathematics

Overview

We develop MATLAB functions to solve upper and lower triangular systems and then solve a more general system with LU factorization.

Activities

1. Suppose we have a 4×4 upper triangular system $A\mathbf{x} = \mathbf{b}$ as below:

$$A = \begin{bmatrix} -5 & 5 & 0 & -2 \\ 0 & 6 & 5 & 7 \\ 0 & 0 & 9 & -7 \\ 0 & 0 & 0 & -4 \end{bmatrix}, \qquad \mathbf{b} = \begin{bmatrix} -2 \\ -9 \\ 3 \\ -7 \end{bmatrix}.$$

   In the command window, define these matrices.

2. To solve for $\mathbf{x}$ using backward substitution, type

```
>> x = zeros(4,1)     % Initialize the solution.
>> x(4) = b(4)/A(4,4) % Solve A(4,4)*x(4) = b(4).
>> % Solve A(3,3)*x(3) + A(3,4)*x(4) = b(3).
>> x(3) = (b(3) - A(3,4)*x(4))/A(3,3)
>> % Solve A(2,2)*x(2) + A(2,3)*x(3) + A(2,4)*x(4) = b(2).
>> x(2) = (b(2) - A(2,3)*x(3) - A(2,4)*x(4))/A(2,2)
>> % Solve A(1,1)*x(1) + A(1,2)*x(2)
>> %                    + A(1,3)*x(3) + A(1,4)*x(4) = b(1).
>> x(1) = (b(1) - A(1,2)*x(2) - A(1,3)*x(3) - A(1,4)*x(4))/A(1,1)
```

3. To perform the same task using matrix multiplication, type

```
>> clear x
>> x = zeros(4,1)
>> x(4) = b(4)/A(4,4)
>> x(3) = (b(3) - A(3,4:4)*x(4:4))/A(3,3)
>> x(2) = (b(2) - A(2,3:4)*x(3:4))/A(2,2)
>> x(1) = (b(1) - A(1,2:4)*x(2:4))/A(1,1)
```

   Remark: When performing matrix multiplication, make sure the matrices have proper dimension.

4. Finish the following M-file.

```
% This function performs backward substitution for 4 x 4 upper
% triangular systems.
% Written by [Your name] on [Today's date].
% Input: matrix A, column vector b; Output: column vector x.
function x = backward4(U,b)
    x = zeros(4,1);
    x(4) = b(4)/U(4,4);
    for i = 3:-1:1
        x(i) = ...
    end
end
```

5. Run your function on the same upper triangular system as given above.

6. Write a function **forward4** to solve 4×4 (or $n \times n$ if you want) lower triangular systems (using forward substitution).

7. We now have *backward4.m* and *forward4.m*, as well as *LU4.m* from last week's lab. Combine these to solve the 4×4 system $A\mathbf{x} = \mathbf{b}$ as below:

$$A = \begin{bmatrix} -8 & 1 & 5 & 9 \\ -6 & 9 & 3 & -4 \\ -5 & -2 & 9 & -9 \\ 8 & -4 & 3 & -3 \end{bmatrix}, \qquad \mathbf{b} = \begin{bmatrix} -2 \\ -7 \\ 9 \\ -5 \end{bmatrix}.$$

Type

```
>> % Perform LU factorization of A. We then have L*U*x = b.
>> [L,U] = LU4(A)
>> % Set y = U*x. Solve L*y = b using forward4.m.
>> y = forward4(L,b)
>> x = backward4(U,y) % Solve U*x = y using backward4.m.
```

Remark on MATLAB

Inside of a function in MATLAB, you can call and use any other function that you have. You just have to make sure that the function that you are calling is in the same folder as the function that is using it. For example, if you have some function transform.m, where the input is a scalar x and the output is a scalar y, then the following function makes use of transform.m:

```
function [W,f] = made_up_function(z,p)
v = sin(z)*p/2;
x = sin(cos(v));
y = transform(x);
f = y^(1/3);
W = f*x;
end
```