ECS130 Scientific Computation Homework Solution Template January 23,2017

Problem 1

(a) Derive, mathematically, the forward substitution method for solving the lower triangular linear system

$$Lx = b$$

using componentwise, row-oriented and column-oriented algorithms, respectively.

(b) Write the three MATLAB functions:

```
x = myfscomponent(L,b)
x = myfsrow(L,b)
x = myfscolumn(L,b)
```

for the componentwise, row-oriented and column-oriented algorithms, respectively.

(c) Test the correctness of your functions and compare the execution time of these functions for a set of different sizes of lower triangular systems.

Problem 1 solution

(a) Derivations:

1. The ith equation of the lower triangular linear system

   $$l_{i,1}x_1 + l_{i,2}x_2 + \cdots + l_{i,i-1}x_{i-1} + l_{i,i}x_i = b_i$$

   can be rewritten as

   $$x_i = \left(b_i - l_{i,1}x_1 - l_{i,2}x_2 - \cdots - l_{i,i-1}x_{i-1}\right)/l_{i,i}$$

   to arrive at the **row-oriented componentwise** form of the forward substitution algorithm.

2. The **row-oriented vectorized** form comes from writing out the dot product

   $$x_i = \left(b_i - \begin{bmatrix} l_{i,1} & l_{i,2} & \ldots & l_{i,i-1} \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ \ldots \\ x_{i-1} \end{bmatrix} \right) / l_{i,i}$$

3. The **column-oriented** forward substitution algorithm is derived as follows (Handout A). By the partition

   $$\begin{array}{c} \\ 1 \\ n-1 \end{array} \begin{array}{c} \begin{array}{cc} 1 & n-1 \end{array} \\ \begin{bmatrix} \ell_{11} & \\ L_{21} & L_{22} \end{bmatrix} \end{array} \begin{bmatrix} x_1 \\ x_{(2:n)} \end{bmatrix} = \begin{bmatrix} b_1 \\ b_{(2:n)} \end{bmatrix}$$

we have

$$\begin{aligned} \ell_{11}x_1 &= b_1 \\ L_{21}x_1 + L_{22}x_{(2:n)} &= b_{(2:n)} \end{aligned}$$

Therefore, $x_1 = b_1/\ell_{11}$, and then after updating $\widehat{b}_{(2:n)} = b_{(2:n)} - L_{21}x_1$, we solve solve the same-kind of lower triangular system with dimension $n-1$:

$$L_{22}x_{(2:n)} = \widehat{b}_{(2:n)}.$$

The procedure is repeated until finding all entries of x.

(b) Code attachments:
myfscomponent implements row-oriented componentwise form of the forward substitution.
myfsrow implements row-oriented vectorized form of the forward substitution.
myfscolumn implements column-oriented form of the forward substitution.
FSbenchmark times the execution of the algorithms.
myfstest tests the correctness of the algorithms.

(c)

1. Performance timing:
   The table below lists the execution times of the three forward substitution algorithms for a range of random matrix sizes from 100 to 5000. The column-oriented forward substitution algorithm performs best in terms of timing.

Table 1: Execution times in second (using MATLAB's **tic-toc** timer)

n	Row-component	Row-vectorized	Col-vectorized
100	0.0033	0.0026	0.0025
1000	0.0199	0.0154	0.0128
2000	0.0815	0.0589	0.0432
3000	0.2012	0.1142	0.0797
4000	0.3619	0.2090	0.1464
5000	0.5869	0.3244	0.2141

2. Accuracy test:
   Starting with a vector of all ones, $\mathbb{1}$, and a random lower triangular matrix L, we can form the right hand side b as $L \cdot \mathbb{1}$ to give us a lower tridiagonal system (L,b) with known solution.

   $error1$, $error2$, and $error3$ are errors of the row-oriented component substitution, row-oriented vectorized, and column-oriented vectorized respectively, measured by taking the 2-norm of the difference between computed and true solution ($\mathbb{1}$) and dividing by square root of n. All three errors are around machine precision.

```
error1 =
     2.381163109968744e-16
```

```
error2 =
     3.510833468576701e-17
error3 =
     3.510833468576701e-17
```