Assignment #11

Due Thursday, 7 December at the start of class

Please read sections 7.1–7.4 of the textbook (J. Epperson, *An Intro. to Numerical Methods and Analysis*, 2nd edition).

Section 7.1, page 419:

- **Exercise 2.** (*Hint*. If A is triangular and has nonzero diagonal, can you solve $Ax = b$ for any b?)

Section 7.2, pages 426–427:

- **Exercise 1.** Do **P16** first and then use those codes.
- **Exercise 10.**
- **Exercise 12.**

Section 7.3, page 429:

- **Exercise 2.**
- **Exercise 4.**
- **Exercise 10.**

P15. Do Gaussian elimination and back-substitution *by hand* to solve the linear system

$$\begin{aligned} x_1 + \frac{1}{2}x_2 + \frac{1}{3}x_3 &= \frac{7}{6} \\ \frac{1}{2}x_1 + \frac{1}{3}x_2 + \frac{1}{4}x_3 &= \frac{5}{6} \\ \frac{1}{3}x_1 + \frac{1}{4}x_2 + \frac{1}{5}x_3 &= \frac{13}{20}. \end{aligned}$$

Make the Gauss elimination and back-substitution stages separate, and clear. As was done on the board in class, use "$R_j \leftarrow R_j - m\,R_i$" notation in the elimination stage.

P16. Following the example done in class, and Algorithms 7.1 (page 421) and 7.2 (page 422), write two codes which do the (naive) Gaussian elimination stage,

```
function [U,c] = gausselim(A,b)
```

and the back-substitution stage,

```
function x = backsub(A,b)
```

Note that for a triangular system you just call `backsub`. For a general system do:

```
>> [U,c] = gausselim(A,b)
>> x = backsub(U,c)
```

P17. *This easy problem is an exercise in* MATLAB *notation.*

Consider the fixed 4×4 matrix

$$A = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \end{bmatrix}.$$

Colon notation can be used for row and column indices in MATLAB (and OCTAVE). For example, `A(p:q,r:s)` extracts the submatrix consisting of rows `p` through `q` and columns `r` through `s`. Note that `:` alone corresponds to all rows or all columns. An example is that `A(:,s)` is the sth column of A. You can also give lists of indices as the arguments of "A(·,·)." For example, given the matrix A above,

$$\texttt{A([1 3],[2 4])} \overset{*}{=} \begin{bmatrix} 2 & 4 \\ 10 & 12 \end{bmatrix}.$$

As another example, recall that `n:-1:1` is a list of the numbers n through 1 in reverse order. This list can be used for indexing.

(a) Input A into MATLAB (or OCTAVE) and check claim $*$.

(b) Using colon notation, give an *exactly ten character* expression for the 3×3 submatrix of A found by removing the first row and column. (*I believe there is a unique answer.*)

(c) Using colon notation, construct a new 4×4 matrix B which has the same entries as A but has the rows in reversed order. (*This can also be achieved by* `B = flipud(A)`.)

(d) The expression `A'` finds the *transpose* of A. What do the expressions `sum(A)` and `sum(A')` do? Explain what `sum(sum(A(:,2:3)))` does, and why you get 68. (*State which entries from A are added and in which order.*)

(e) There is also a `reshape` command. First read `help reshape`. Then give a short expression using `reshape` and the transpose operation `'` which creates the original matrix A. Specifically, fill in this expression with one character in each box:
`A = reshape(□□□□,□,□)□`.