Computational methods and applications (AMS 147)

Homework 6 - Due Saturday, March 9th

Please submit to CANVAS a .zip file that includes the following Matlab functions:

`backward_sub.m`

`tridiag_solver.m`

`matrix_inverse.m`

For Exercise 3, scan your notes into one PDF file `scan.pdf`, and attach it to your submission (as a separate file, not as part of the .zip file)

Exercise 1 Write a Matlab program that returns the numerical solution to the upper triangular system of equations

$$\begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1n} \\ 0 & u_{22} & \ddots & \vdots \\ \vdots & \ddots & \ddots & u_{(n-1)n} \\ 0 & \cdots & 0 & u_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix} \tag{1}$$

by using the backward substitution algorithm. The function name, inputs and output are given below

```
function [x] = backward_sub(U,b)
```

Input:

`U`: upper triangular matrix

`b`: column vector representing the right hand side of (1)

Output:

`x`: solution to the linear system (1) (column vector)

Exercise 2 Write a Matlab/Octave function that implements the Thomas algorithm to solve tridiagonal linear systems of equations of the form

$$\begin{bmatrix} a_1 & c_1 & 0 & \cdots & 0 \\ e_1 & a_2 & c_2 & \ddots & 0 \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & a_{n-1} & c_{n-1} \\ 0 & \cdots & 0 & e_{n-1} & a_n \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_{n-1} \\ b_n \end{bmatrix} \tag{2}$$

The function name, inputs and output are given below

```
function x = tridiag_solver(e,a,c,b)
```

Input:
`e`=$[e_1\ e_2\ \cdots\ e_{n-1}]$, `a`=$[a_1\ a_2\ \cdots\ a_n]$, `c`=$[c_1\ c_2\ \cdots\ c_{n-1}]$, `b`=$[b_1\ b_2\ \cdots\ b_n]^T$.

Output:
`x`: solution to the linear system (2) (column vector)

Hint: You can debug the output of your functions `backward_sub(U,b)` and `tridiag_solver(e,a,c,b)` by comparing it with the output of the Matlab/Octave built-in linear solver `linsolve(A,b)` (see the documentation), for suitable prototype upper-triangular and tri-diagonal matrices, respectively.

Exercise 3 By using the Gauss elimination method with pivoting by row, compute the LU factorization (including the permutation matrix) of the following singular matrix

$$A = \begin{bmatrix} 1 & -4 & 3 \\ 1 & -1 & 1 \\ -3 & 12 & -9 \end{bmatrix}$$

Verify the correctness of your results by checking that that $PA = LU$, where P is the permutation matrix. Is it true that the the eigenvalues of A are the diagonal entries of the upper triangular matrix U?

Extra Credit The inverse of an $n \times n$ matrix A can be computed by using the Leverrier algorithm in $(n-1)n^3$ flops

$$\alpha_k = \frac{1}{k}\text{trace}(AB_{k-1}), \qquad B_k = -AB_{k-1} + \alpha_k I \qquad B_0 = I, \qquad k = 1, ..., n, \tag{3}$$

$$A^{-1} = \frac{1}{\alpha_n} B_{n-1}. \tag{4}$$

Write a Matlab function that implements the algorithm (3)-(4). The function should be of the form

```
function [Ai] = matrix_inverse(A)
```

Input:
`A`: invertible matrix

Output:
`Ai`: inverse of the matrix `A`

Hint: You can debug the output of your function by comparing it with the Matlab/Octave built-in function `inv(A)`.