Assignment #2, due Oct. 16

1. Let \( A = \begin{bmatrix} 2 & -4 & 1 & -2 \\ -1 & 2 & 0 & 2 \\ 4 & -4 & 2 & 4 \\ 1 & 3 & 1 & 3 \end{bmatrix} \) and \( b = \begin{bmatrix} -1 \\ 1 \\ 1 \\ 2 \end{bmatrix} \). Do (a) by hand with pencil and paper, use fractions.

(a) Perform Gaussian elimination with pivoting to find \( L, U, p \), using the pivot candidate with largest absolute value. Show \( L, U, p \) after each step, see the example on the web page. Check that \( LU \) gives the correct result. Use \( L, U, p \) to solve the linear system \( Ax = b \).

(b) Use \( lu \) in Matlab to find \( L, U, p \), then use this to solve \( Ax = b \).

2. Investigate the following linear systems: Print the approximate condition number \( c_1 \approx \text{cond}_1(A) \) (use \text{invnormest}). For the given rhs vector \( b \) find the computed solution \( \hat{x} \) using \( lu \) (for (i) use \text{lu} instead).

(iii) \( A = \begin{bmatrix} .1 & .3 & .4 \\ .3 & .9 & .2 \\ .1 & .2 & .3 \end{bmatrix} \) with \( a_{ij} = \frac{1}{i+j-1} \), \( x_j = \begin{cases} 1 & \text{if } j = 7, 8 \\ 1 & \text{otherwise} \end{cases} \), let \( b := Ax \).

If not: perform one step of iterative improvement, go to \( \diamondsuit \).

(ii) \( A = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 3 & 4 & 5 & 6 \\ 7 & 8 & 9 & 10 \\ 11 & 12 & 13 & 14 \end{bmatrix} \) and the right hand side vector \( b \) find the computed solution \( \hat{x} \). Run the program for \( N = 20 \), print the actual error \( \| \hat{x} - x \|_1 / \| x \|_1 \) for comparison (since we know the exact solution vector \( x \) in each case). Was the computation numerically stable?

3. A square plate with \( x \in [0, 1] \) and \( y \in [0, 1] \) is is heated at the edges: assume the bottom edge has temperature 100, the other three edges have temperature 0. We want to find the temperature in the point \( (\frac{1}{4}, \frac{1}{4}) \).

We divide the square into a grid of \( N^2 \) smaller squares of size \( h = 1/N \). We number the interior grid points from 1 to \( M = (N - 1)^2 \) rowwise, left to right, from bottom to top (see picture below). The unknown temperature at the interior grid points are denoted by \( u_1, \ldots, u_M \). The temperature at an interior grid point is equal to the mean of the temperature of the four neighboring grid points.

(a) For \( N = 4 \) we have \( h = \frac{1}{4} \). We obtain one equation for each of the \( M = 9 \) interior grid points: E.g., in point 1 we obtain the equation \( u_1 = (100 + u_2 + u_4 + 0)/4 \), in point 2 we obtain \( u_2 = (100 + u_3 + u_5 + u_1)/4 \)

Write down the 9 equations for \( u_1, \ldots, u_9 \). Find the matrix \( A \in \mathbb{R}^{M \times M} \) and the right hand side vector \( b \) of the linear system \( Au = b \) (the diagonal elements of \( A \) should be 1). Solve the linear system in Matlab using \( lu \) and print \( u \).

(b) Write a Matlab program which works for any \( N \): use for loops to generate the matrix \( A \in \mathbb{R}^{M \times M} \) (the entries are \( 1, -\frac{1}{4} \) or 0) and the right hand side vector \( b \). Run the program for \( N = 20 \), print out the temperature at the point \( (\frac{1}{4}, \frac{1}{4}) \) and make a contour plot of the temperature (see hints on course web page).

(c) Solve the problem for \( N = 20, 40, 80, 160, \ldots \) using “lu”.

(d) Make the code more efficient by using sparse matrices in Matlab. Use \text{spgridmat.m} from the web page to compute \( A \). Solve the linear system using \( u = A \backslash b \). For each \( N \) use \text{fprintf} to print a line like

N=8, M=49, cond=37.2647, temperature=43.1053056827363, time=0.000523

showing \( N, M, \) the estimate of \( \text{cond}_1(A) \), the temperature at the point \( (0.25, 0.25) \), the time for the solution of the linear system. Do NOT print anything else. What is the largest value of \( M \) (the number of unknowns) where you can still solve the problem in Matlab?

(e) Make the code more efficient by using sparse matrices in Matlab. Use \text{spgridmat.m} from the web page to compute \( A \). Solve the linear system using \( u = A \backslash b \). For each \( N \) use \text{fprintf} to print a line like

N=8, M=49, temperature=43.1053056827363, time=0.000523

showing \( N, M, \) the temperature at \( (0.25, 0.25) \) and the time for the command \( u = A \backslash b \). Do NOT print anything else.

Solve the problem for \( N = 80, 160, 320, 640, \ldots \). What is the largest value of \( M \) (the number of unknowns) where you can still solve the problem in Matlab?
4. (a) Write down the divided difference table and find the interpolating polynomial \( p(x) \) for \( f(x) = \sqrt{x} \) and the nodes 1, 4, 9, 16. Use it to approximate \( f(12) \). What bound does the error formula give for \( |f(12) - p(12)| \)?

(b) Now we use for \( f(x) = \sqrt{x} \) the values \( f(1), f'(1), f(4), f'(4) \). Write down the divided difference table and find the interpolating polynomial. Use it to approximate \( f(3) \). What bound does the error formula give for \( |f(3) - p(3)| \)?

5. (a) Consider the function \( f(x) = \cos(x) \) on the interval \([-\pi, \pi]\). For \( n = 5, 11, 17, 23, \ldots, 41 \) find (A) the polynomial interpolation using \( n \) equidistant nodes, (B) the polynomial interpolation using \( n \) Chebyshev nodes, (C) the complete cubic spline using \( n \) equidistant nodes. Evaluate the interpolating function \( p(x) \) at 1000 equidistant nodes \( \tilde{x}_j \) and find the maximal error \( E := \max_{j=1,\ldots,1000} |f(\tilde{x}_j) - p(\tilde{x}_j)| \). For each \( n \) print out a line like

\[ \text{n=5: E_A=0.55987, E_B=.1234, E_C=.13823} \]

Do NOT print anything else. Do NOT plot anything.

(b) Repeat this for the function \( f(x) = 1/(10 - 9 \cos x) \) on \([-\pi, \pi]\).

(c) What do you observe in each case as \( n \) increases? Which method works best for (a),(b)? Where can you see the effect of roundoff error?