

AMSC/CMSC 460
Problem set 2

1. Consider the linear system $Ax = b$ where A is the matrix

$$A = \begin{bmatrix} 6 & 2 & 2 \\ 2 & 2/3 & 1/3 \\ 1 & 2 & -1 \end{bmatrix}$$

and $b = (-2, 1, 0)$.

a) Verify that the exact solution is $x_1 = 2.6$, $x_2 = -3.8$, $x_3 = -5.0$.

b) Using four digit floating-point decimal arithmetic with rounding, solve this system by Gaussian elimination without pivoting. In performing the arithmetic operations, remember to round to four digits after each operation. In particular, use this rounding when entering the elements of the matrix, and in computing the multipliers. Call the computed solution y .

c) Repeat part b) using partial pivoting. Call the computed solution z . Compare the solutions y and z with the exact solution x .

2. The $n \times n$ Hilbert matrix is

$$H_n = [h_{i,j}] \quad \text{where } h_{i,j} = \frac{1}{i+j-1}, \quad 1 \leq i, j \leq n.$$

This matrix is nonsingular and the inverse may be computed explicitly, The inverse $H_n^{-1} = a_{ij}$ where

$$a_{ij} = \frac{(-1)^{i+j}(n+i-1)!(n+j-1)!}{(i+j-1)[(i-1)!(j-1)!]^2(n-i)!(n-j)!}.$$

The MATLAB functions `hilb(n)` and `invhilb(n)` give H_n and H_n^{-1} respectively, using the formulas. If we set $b_n = (1, 0, \dots, 0)$, then the exact solution x_n of $H_n x_n = b_n$ is the first column of H_n^{-1} .

(a) Using the MATLAB back slash command, solve for x_n for the cases $n = 5$ and $n = 10$. Call this computed result y_n .

(b) Let $e_n = x_n - y_n$ where x_n is the exact solution as given by the formula above and let $r_n = b_n - H_n y_n$ be the residual. Find the condition number of H_n in the 2-norm using the MATLAB command `cond` for $n = 5$ and $n = 10$. Verify that

$$\frac{\|e_n\|}{\|x_n\|} \leq \text{cond}(H_n)u$$

where $u = \text{eps}/2$ is the unit roundoff, and $\| \cdot \|$ is the 2-norm. The MATLAB command `norm(v)` computes the 2-norm of a vector v .

(c) How many accurate digits are there in the computed solutions in each case?

(d) Verify in each case that

$$\frac{\|r_n\|}{\|H_n\|\|y_n\|} \leq u.$$

3. a) Use the MATLAB command `chol` to find the Cholesky factorization of the matrix

$$A = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & .25 & -.5 & 1.5 \\ 1 & -.5 & 18 & 1 \\ 0 & 1.5 & 1 & 110 \end{bmatrix}$$

b) Let R be the upper triangular matrix found in part a) such that $A = R^T R$. Compare $\text{cond}(R)$ and $\text{cond}(A)$. Which is smaller?

c) Let B be a random 10×10 matrix, produced with the command `B = rand(10,10)`. Let $A = B^T B$. Let R be the upper triangular matrix such that $A = R^T R$ obtained from the Cholesky factorization, `R = chol(A)`. Compare the condition numbers of A and R .

4. Problem 2.3, page 82 of Moler.

5. Consider the problem of finding a solution of the boundary value problem for given $f(x)$

$$\begin{aligned} -u''(x) &= f(x) & \text{for } 0 < x < 1 \\ u(0) &= u(1) = 0. \end{aligned} \tag{1}$$

The solution u can be found analytically in the form of an integral, but we shall use a finite difference approximation to calculate some numerical solutions of this problem.

First we approximate the second derivative with a difference quotient:

$$u''(x) \approx \frac{u(x + \Delta x) - 2u(x) + u(x - \Delta x)}{(\Delta x)^2}$$

so that the DE becomes

$$-\frac{u(x + \Delta x) - 2u(x) + u(x - \Delta x)}{(\Delta x)^2} \approx f(x).$$

Now introduce grid points in the interval $[0, 1]$. Let the number of subintervals be n , setting $\Delta x = 1/n$ and $x_j = j\Delta x$, $j = 0, \dots, n$. We let u_j be the approximate value we wish to compute for $u(x_j)$. The u_j are chosen to satisfy the system of linear equations

$$-u_{j+1} + 2u_j - u_{j-1} = (\Delta x)^2 f(x_j), \quad j = 1, \dots, J$$

where $J = n - 1$ is the number of interior mesh points. The boundary conditions say that $u_0 = u_{J+1} = 0$ so we get the $J \times J$ system

$$T\mathbf{u} = (\Delta x)^{-2}S\mathbf{u} = \mathbf{f}$$

where S is the the $J \times J$ tridiagonal matrix

$$S = \begin{bmatrix} 2 & -1 & 0 & \dots & 0 \\ -1 & 2 & -1 & \dots & 0 \\ 0 & -1 & 2 & \dots & 0 \\ \cdot & \cdot & \cdot & \dots & 0 \\ \cdot & \cdot & \cdot & \dots & 0 \\ \cdot & \cdot & \dots & 2 & -1 \\ 0 & 0 & \dots & -1 & 2 \end{bmatrix}$$

$\mathbf{u} = (u_1, \dots, u_J)$ and $\mathbf{f} = (f(x_1), \dots, f(x_J))$.

Write a short MATLAB program (script mfile) to solve this system with the given function f as input. You can write f as in inline function. The program should have n as an input parameter and should plot the solution u along with the function f on the interval $[0, 1]$. The program should be written to accomodate values of n as large as 200 ($J = 199$). Remember that in MATLAB an index in a vector always starts with one.

We shall take advantage of the fact that S is a sparse matrix to save on storage space. You will need to enter the matrix S as a sparse matrix using the `sparse` commands of MATLAB. Enter `help sparse` at the MATLAB prompt to get information.

Note that S can be written $S = E + D + E'$ where $D = 2I$ and E has -1 on the super diagonal and zeros elsewhere. D and E can each be entered as sparse matrices.

You can use the MATLAB command `u = T\f` to solve the system. You can also use the sequence of commands

$$\begin{aligned} [L,U] &= \text{lu}(T) \\ v &= L \backslash f; \\ u &= U \backslash v; \end{aligned}$$

For this size problem, the LU factorization does not save much time.

a) For starters take $f(x) = 1$. The exact solution is $u(x) = x(1-x)/2$. Note that the values of u at the points x_j satisfy the difference equation exactly. You can use this solution to see if your program is working correctly.

b) Now use the function $f(x) = \sin(\pi x)$ with exact solution $u(x) = \sin(\pi x)/\pi^2$. Run your program with values of $n = 10, 50, 100, 200$. To see convergence of the numerical solutions to the exact solution, plot the numerical solutions for these values of n together on the same graph using the `hold on` command.

c) For each n of part c) compute the error $e_n = \max |u_j - \sin(\pi x_j)/\pi^2|$. According to what power of $\Delta x = 1/n$ is e_n tending to zero as n increases?

d) Now try a function f which is continuous but not C^1 , such as

$$f(x) = \begin{cases} x/a, & 0 \leq x \leq a \\ (1-x)/(1-a), & a \leq x \leq 1 \end{cases} .$$

How does the solution curve compare with f ? Is it smoother, or does it have the same jump in the derivative at $x = a$? Again try $n = 20, 50, 100, 200$ and observe the convergence of the computed solutions.