Numerical Analysis 1 – Class 3

Thursday, February 4th, 2021

Subjects covered

- Dense and sparse matrices.
- Matrix multiplication algorithms and complexity.
- Matrix norm, condition number.
- Introducing the SVD (singular value decomposition).
- Visualizing a matrix.

Reading

- Kutz, Chapters 2.1 and 15.1 – 15.2.
- C. Moler, “Numerical Computing with MATLAB”, Chapter 2, “Linear Equations” (linked on Canvas).
- "The Extraordinary SVD", by C. Martin and M. Porter (linked on Canvas).

Problems

Most of the following problems require you to write a program. For each program you write, please make sure you also write a test which validates your program. E-mail your answers to our TA: Hiu Ying Man, man.h@northeastern.edu.

Problem 1

The goal of this problem is to explore the relationship between matrix condition number and errors incurred when solving the linear system $A x=b$ for *x*. Consider the so-called Hilbert matrix of order *n*, H_n. This is a matrix whose elements are given by

$$h_{i,j}=\frac{1}{i+j-1}$$

where *i* and *j* are the row and column indices of the matrix, and we assume the Matlab convention where the index count starts at 1. (In C, and some other computer languages the index count starts at 0. Therefore, there are two different ways to index into an array commonly found out in the world. This leads to all kinds of confusion and program bugs – be aware of this fact!)

This matrix is well known to be badly conditioned. The Mathworks’s founder Cleve Moler has a couple of interesting blog posts about this matrix which you can find here:

https://blogs.mathworks.com/cleve/2013/02/02/hilbert-matrices/
https://blogs.mathworks.com/cleve/2017/06/07/hilbert-matrices-2/

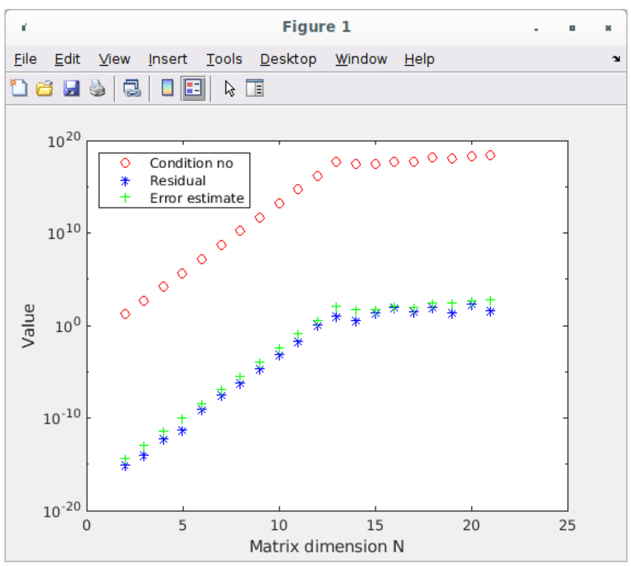
Please read the blog posts for more background about the Hilbert matrix.

Consider the linear system $H_n x = b$. Write a program which does the following:

1. Start with a known vector $x_0 = [1,1,1,1,\cdots]^T$ where x0 is of length n.
2. Create the nth degree Hilbert matrix using the Matlab built-in “hilb()”.
3. Compute the matrix condition number cn = cond(Hn).
4. Compute the matrix vector product b = Hn x0.
5. Perform the linear solve operation xc = Hn \ b.
6. Compute the norm of the difference between the computed and the starting x0: $r_n = \|x_0 - x_c\|$ In theory, what should be the value of rn?
7. Loop through values of n = 2, 3, ..., 21 and make a plot of rn and cn vs. matrix order n. My plot is shown below.
8. For each n, also compute and plot the error estimate given by Moler, errest = cond(Hn)*eps(1).

Now answer the questions:

- How many orders of magnitude separate the value of the residual and the value of the condition number? Why is this the case?
- Why do the condition number and the residual seem to saturate for $n \geq 13$? Hint: Matlab’s cond() actually computes the inverse of the condition number, $1/\kappa$. What effect might round-off error have on $1/\kappa$?



Problem 2

The goal of this problem is to get some additional experience with thinking about symmetric matrices and quadratic forms. Consider the 2x2 matrix A where the parameter *a* takes values $a \in [0,2]$.

$$A = \begin{pmatrix} 1 & a \\ a & 1 \end{pmatrix}$$

Please do the following:

1. Compute the eigenvalues of A as functions of parameter a.
2. As mentioned in class, the singular values and eigenvalues of a *symmetric* matrix are related by $\sigma_i = |\lambda_i|$. Knowing this, please write down an expression for the condition number of matrix A as a function of the parameter a. What value of a causes A to become singular?
3. Consider the quadratic form $f(u) = u^T A u$ where u is the two-element vector describing a point in the [x, y] plane. That is, $u = [x, y]^T$. Write a program which computes and makes plots of $f(u)$ while varying the parameter a.
4. What happens to the quadratic form when the matrix is singular?

Problem 3

A truss is a architectural structure frequently used to create a rigid framework capable of bearing great loads. You will recognize many trusses in the real world acting as bridges, roof braces, and power-line supports amongst other structures. An example truss bridge is shown on the right. Analysis of a truss involves computing the forces on all of its beams to verify that none of the beams are near their breaking limit.

Consider the simple truss bridge shown in the figure below. Its joints and beams are lettered/numbered for analysis. Joint a is considered fixed to its anchor both horizontally and vertically. Joint f is fixed vertically, but is free to move horizontally. (This is common in real-world bridges. If you look at the footings of real bridges, you will see that one end of the bridge is fixed, while the other typically rests on a support which allows for horizontal movement. This is done so the bridge can expand/contract in response to temperature changes without buckling.)

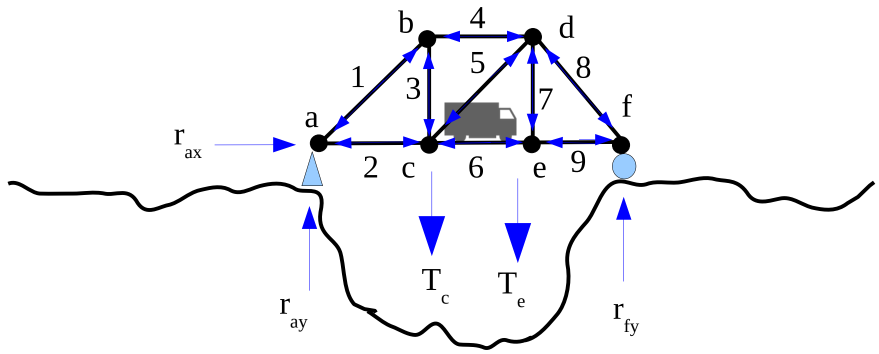
Take the angles as either 45 or 90 degrees. Also, imagine there is a large weight (a 15 ton truck) pulling down at the center of the bridge (between joints c and e). For the purposes of this problem, imagine the truck's weight is divided equally between joints c and e. In opposition, the ground pushes up on both ends of the bridge (joints a and f). This is a reaction force. Finally, a reaction force pushes on the bridge at joint a in the horizontal direction.

A simple analysis of this truss is performed by considering the force balance equations. This may be expressed as a matrix which relates the unknown internal forces on all beams to the known externally applied forces (i.e. the truck). The reaction forces pushing on the bridge are regarded as internal forces since they are unknown. We write this relation as

$$A f_i = -f_e$$

where f_i is a vector (list) of the internal forces on each beam, and f_e is the vector of known external forces. The matrix A holds the information detailing how the forces interact, i.e. the structure of the bridge. The goal is to find the internal forces f_i .

To find the matrix A, we use the fact that the sum of all forces at each joint is zero in equilibrium. The forces at each joint are resolved into *x* and *y* (horizontal and vertical) components. For each force component, we write an equation in which the forces sum to zero.



The equilibrium equations are:

- Joint a, x: $-f_1\cos(45)-f_2+r_{ax}=0$
- Joint a, y: $-f_1\sin(45)+r_{ay}=0$
- Joint b, x: $f_1\cos(45)-f_4=0$
- Joint b, y: $f_1\sin(45)+f_3=0$
- Joint c, x: $f_2-f_5\cos(45)-f_6=0$
- Joint c, y: $-f_3-f_5\sin(45)=-T_c$
- etc.

The forces at each joint are depicted by the blue arrows in the figure; the direction of the arrow shows the direction of the force. We use the convention that force components pointing up or to the right are positive, down or to the left are negative. Your goal is to compute the internal forces on the beams and verify none of them exceed the limit of the beam. We assume the beams have the following limits:

- Compression < 14 tons. If compression is larger than this limit, the beam will bend and break.
- Tension < 11 tons. If tension is larger than this limit, the beam will snap apart.

Note that as drawn, compressive forces on the beams are negative, and tension forces on the beams are positive. (Compression and tension follow the blue arrows in the figure.)

Please do the following:

1. Complete the above analysis and write down the entire matrix equation describing the forces at each joint.
2. Write down the external force vector, f_e . Recall that we take $T_c=T_e=$ truck weight/2.
3. Solve the system (using Matlab, or your preferred solver), and find the vector of member forces, $f_1, f_2, f_3,$ etc.
4. Will the bridge collapse because of weight of the 15 ton truck?
5. What happens if somebody drives a 20 ton truck on this bridge?

The goal of this problem is to create and use a sparse matrix in a in real-world application. If you need help with this problem, I have placed an analysis of a similar problem onto Canvas. It shows the general approach of how to create the equilibrium matrix for a truss, solve it, and then interpret the solution.