Numerical Analysis 1 – Class 4

Thursday, February 1st, 2018

Subjects covered

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

Reading

- Kutz, Chapters 2.1 and 15.1 – 15.2.
- "A brief overview of the condition number", by Aaron C Acton (linked on Blackboard).
- "The Extraordinary SVD", by C. Martin and M. Porter (linked on Blackboard).
- Optional: C. Moler, "Numerical Computing with MATLAB", Chapter 2. http://de.mathworks.com/moler/lu.pdf
- Optional: "Sparse matrices in Matlab: design and implementation", J. Gilbert et al. https://www.mathworks.com/help/pdf_doc/otherdocs/simax.pdf

Problems

Most of the following problems require you to write a program. As always, please make sure you also write a test program which validates your program's implementation. You will be graded on both your program as well as on your test. E-mail your answers to our TA: Mohamed Elbehiry, elbehiry.m@husky.neu.edu.

1. Recall the condition number for a matrix, defined by

$$\kappa = \|A\| \|A^{-1}\|$$

One way to comprehend the condition number is by considering solving the equation $A x = b$. When solving $A x = b$, the known quantities are the matrix *A* and the vector *b*. The unknown quantity is the vector *x*. Accordingly, you can consider the process of solving this system as taking an input *b* and getting an output *x* using the computation *x* = *A**b*.

Now consider a perturbation to the input: $b + \delta b$. A perturbation (error) on the input causes a perturbation (error) on the output, $x + \delta x$. The relationship between the perturbed input and output is $A(x + \delta x) = (b + \delta b)$.

The condition number can then seen as governing the relationship between the normalized input and output perturbations. That is, the condition number is the "amplification factor" for any perturbations on the input. The relationship between input and output perturbations is:

$$\frac{\|\delta x\|}{\|x\|} \le \kappa \frac{\|\delta b\|}{\|b\|}$$

Please prove this statement. This is a hand derivation. Please turn in your written derivation.

2. The goal of this problem is to explore experimentally the relationship between matrix condition number and errors incurred when solving the linear system $Ax = b$ for x.

   Consider the so-called Frank matrix F_n . This is a matrix family (of order n) whose members are Hessenberg matrices with ill conditioned eigenvalues and whose determinant is 1. They follow a regular pattern for increasing order n. Here are a couple of examples:

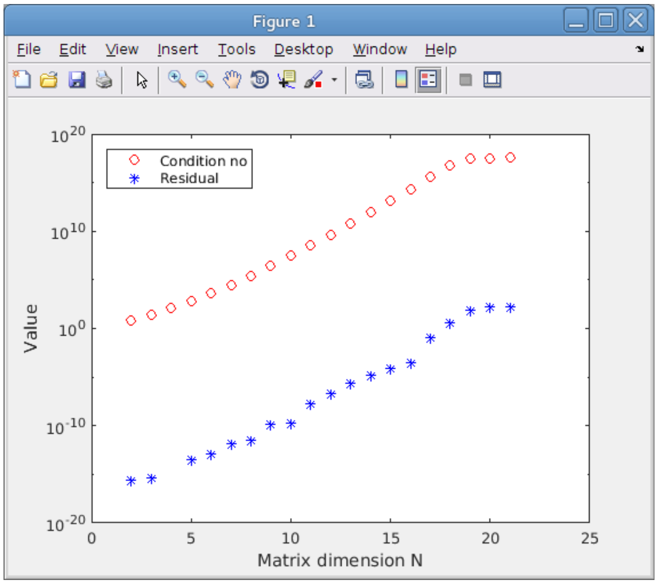
$$F_4 = \begin{pmatrix} 4 & 3 & 2 & 1 \\ 3 & 3 & 2 & 1 \\ 0 & 2 & 2 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix} \qquad F_6 = \begin{pmatrix} 6 & 5 & 4 & 3 & 2 & 1 \\ 5 & 5 & 4 & 3 & 2 & 1 \\ 0 & 4 & 4 & 3 & 2 & 1 \\ 0 & 0 & 3 & 3 & 2 & 1 \\ 0 & 0 & 0 & 2 & 2 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix}$$

   Based on these examples, the pattern used to create a Frank matrix is easy to pick out.

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

   1. Start with a known vector $x_0 = [1, 1, 1, ..., 1]^T$ where x_0 is of length n.
   2. Create the nth degree Frank matrix. (This is best done as a sub-function.) Matlab has a function "gallery" which will return a Frank matrix when give the correct inputs. Moreover, you can find functions which build this matrix online. Nonetheless, please write your own implementation since it's a good exercise. Feel free to test your implementation against the one in Matlab's "gallery".
   3. Compute the matrix condition number $c_n = \text{cond}(F_n)$.
   4. Compute the matrix vector product $b = F_n x$.
   5. Now perform the linear solve operation $x_c = F_n \backslash b$.
   6. Compute the norm of the difference between the computed and the starting x:
      $r_n = \|x_0 - x_c\|$ In theory, what should be the value of r_n?
   7. Loop through values of $n = 2, 3, ..., 21$ and make a plot of r_n and c_n vs. matrix order n. My plot is shown below.

   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? Can you change something in your program to demonstrate why this is true? You can turn in your answers in a .txt file.



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 things. As an example, consider the so-called “King post truss” shown in the photo below 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 truss bridge shown in the drawing on the next page below. Its joints and beams are lettered/numbered for analysis. Joint a is considered fixed to its anchor both horizontally and vertically. Joint d 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.)

King post truss

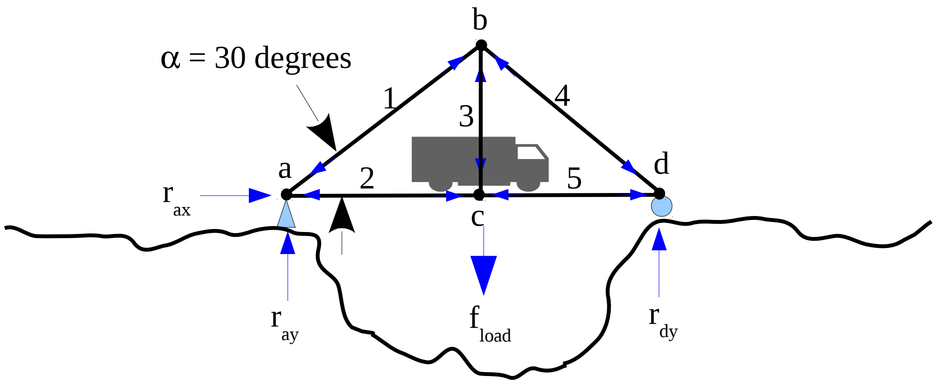
Take the angle α = 30 degrees. Also, imagine there is a large weight (a 12 ton truck) pulling down at the center of the bridge (joint c). In opposition, the ground pushes up on both ends of the bridge (joints a and d). Finally, a reaction force pushes on the bridge at joint a in the horizontal direction.

A simple analysis of this truss is performed by creating its so-called “equilibrium matrix”. This is 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

$$\boldsymbol{A} f_{\text{int}} = -f_{\text{load}}$$

where f_{int} is a vector of the unknown internal forces, and f_{load} is the vector of known external (load) forces. The internal forces correspond to compression or stretching of each of the individual beams. The matrix A is the bridge's “equilibrium matrix”. A detailed description of this method is available on Blackboard.

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, or to the externally applied force.



The equilibrium equations are:

Joint a, x: $-f_1\cos(30)-f_2+r_{ax}=0$

Joint a, y: $-f_1\sin(30)+r_{ay}=0$

Joint b, x: $f_1\cos(30)-f_4\cos(30)=0$

Joint b, y: $f_1\sin(30)+f_3+f_4\sin(30)=0$

Joint c, x: $f_2-f_5=0$

Joint c, y: $f_3=f_{load}=-12$

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. Please do the following:

1. Complete the above analysis, and write down the entire matrix equation describing the forces at each joint.
2. Solve the system (using Matlab, or your preferred solver), and find the vector of member forces, f_1, f_2, f_3, etc. Please create a function which sets up the internal and external force matrices, then performs the solve operation and prints the results.
3. The strength of each beam is: compression 10 tons, tension 15 tons. Will the bridge collapse because of weight of the truck? (Recall that the truck weighs 12 tons.)
4. Now suppose the bridge designer makes the angle $\alpha=45$ degrees. Will the bridge support the weight of the truck?

The goal of this problem is to create and compute with a sparse matrix used in real-world applications.