

# AMS 526 Homework 3, Fall 2012

Due: Friday 10/12 (extended from 10/10) in class

1. (10 points) Each of the following problems describes an algorithm implemented on a computer satisfying the two axioms of floating point numbers (axioms (13.5) and (13.7) in the textbook). For each problem, answer whether the algorithm is backward stable, stable but not backward stable, or unstable. Prove your assertion or give a reasonably convincing argument.

- (a) Data:  $x \in \mathbb{C}$ . Solution:  $2x$ , computed as  $x \oplus x$ , where  $\oplus$  denotes floating-point addition.
- (b) Data: none. Solution:  $e$ , computed by summing  $\sum_{k=0}^{\infty} 1/k!$  from right to left using floating-point multiplication  $\otimes$  and floating-point addition  $\oplus$ , stopping when a summand is reached of magnitude  $< \epsilon_{\text{machine}}$ .

2. (10 points) Exercise 2.5.9 on p.147
3. (10 points) Exercise 2.7.24 on p. 168
4. (10 points) Exercise 2.8.8 on p. 175
5. (20 points) Exercise 2.9.15 on p. 180
6. (10 points) Exercise 3.2.8 on p. 190
7. (30 points) Write a routine in MATLAB or Octave to estimate the condition number of a real matrix  $A$  using 1-norm. You will need to compute  $\|A\|_1$ , which is easy, and estimate  $\|A^{-1}\|_1$ , which is more challenging. One way to estimate  $\|A^{-1}\|_1$  is to take it as the ratio  $\|z\|_1/\|y\|_1$ , where  $z$  is the solution to  $Az = y$  and  $y$  is picked by some heuristic to maximize the ratio.

We choose  $y$  as the solution to the system  $A^T y = c$ , where  $c$  is a vector each of whose components is  $\pm 1$ , with the sign for each component chosen by the following heuristic. Using the factorization  $PA = LU$  (you may use MATLAB's routine `lu`), the system  $A^T y = c$  is solved in two stages, successively solving the triangular systems  $U^T v = c$  and  $L^T P y = v$ . At each step of the first triangular solution, choose the corresponding component of  $c$  to be 1 or  $-1$ , depending on which will make the resulting component of  $v$  larger in magnitude. You will need to write a custom triangular solution routine to implement the selection of  $c$ . Then solve the second triangular system  $L^T P y = v$  in the usual way for  $y$ , for which you can use MATLAB's backslash "`\`" operator. The idea here is that any ill-conditioning in  $A$  will be reflected in  $U$ , resulting in a relatively large  $v$ . The relative well-conditioning unit triangular matrix  $L$  will then preserve this relationship, resulting in relatively large  $y$ .

Test your program on the Hilbert matrix of order  $n = 2, 3, \dots, 12$ . To check the quality of your estimates, compute  $A^{-1}$  explicitly using the LU factorization that was already computed and then compute the condition number  $\|A\|_1 \|A^{-1}\|_1$ . Plot the estimated condition numbers and the explicitly computed condition numbers for the Hilbert matrix of order  $n = 2, 3, \dots, 12$  (using the horizontal axis for  $n$  and the vertical axis for the condition numbers). Compare the required flops of the two approaches (i.e., estimation and explicit computation), and also plot their running times for different  $n$ . To obtain reliable timing of a procedure, you may need to run it repeatedly for hundreds of iterations and then take the average (use the `tic()` and `toc()` functions outside the iterations).

Submit your programs, the plots, and a brief discussion of your results.

Remarks: The condition-number estimator in Q7 is from [1]. Modern software (e.g., MATLAB) estimates condition numbers using more advanced methods in [2, 3]. (You are not required to read these articles.)

## References

- [1] A. K. Cline, C. B. Moler, G. W. Stewart, and J. H. Wilkinson, "An Estimate for the Condition Number of a Matrix," *SIAM J. Numer. Anal.*, Vol 16, pp. 368–375, 1979.
- [2] W.W. Hager, "Condition Estimates," *SIAM J. Sci. Stat. Comput.* Vol 5, pp. 311-316, 1984.
- [3] N.J. Higham and F. Tisseur, "A Block Algorithm for Matrix 1-Norm Estimation with an Application to 1-Norm Pseudospectra," *SIAM J. Matrix Anal. Appl.*, Vol. 21, pp. 1185-1201, 2000.