2019 09 16
陈清逸

Statistics and Numerical Method — Problem Set #1 (due 9/30/2019)

1. Machine epsilon (6 pts)

Write a computer program in C (or C++) that experimentally determines the machine epsilon ϵ_m, i.e., the smallest number ϵ_m such that $1 + \epsilon_m$ still evaluates to something different from 1.

(1). What are the ϵ_ms for data types float and double, respectively? (2pts)

(2). For each data type, what is the smallest positive normalized number $f_{\min}$ allowed? (2pts)

(3). Is $f_{\min}$ the same as ϵ_m? Why? (2pts)

2. Pitfalls of floating point arithmetic (5 pts)

Consider the following numbers: $a = 1.0 \times 10^{17}$, $b = -1.0 \times 10^{17}$, $c = 1.0$. They are all double precision numbers.

(1). Calculate the results for $x = (a + b) + c$ and $y = a + (b + c)$. (2pts)

(2). Which one is correct, if any? Explain why the law of associativity is here broken. (3pts)

3. Packing of numbers (4 pts)

Estimate how many numbers there are in the interval between 1.0 and 2.0, and in between the interval of 511.0 to 512.0, for IEEE-754 for (a) single precision and (b) double precision data types.

4. Hilbert Matrix (20 pts)

Hilbert matrix is a well-known example of ill-conditioned matrices, which looks like this:

$$A = \begin{bmatrix} 1 & 1/2 & 1/3 & \dots & 1/n \\ 1/2 & 1/3 & 1/4 & \dots & 1/(n+1) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1/n & 1/(n+1) & 1/(n+2) & \dots & 1/(2n-1) \end{bmatrix}. \tag{1}$$

In this problem, we ask you to write your own piece of code, in your favorite programming language, to solve the problem of $A\boldsymbol{x} = \boldsymbol{b}$. You should make your code sufficiently flexible so that you can switch between single and double precisions, and compare the results in problems (2)-(3) below. **You are required to attach your code in original format** for (1), (2) and (4), and your code must be properly documented/commented (e.g., explain the meaning of variables, function of each loop, and important sentences), which are important references for grading.

(1). Write a direct matrix solver **on your own**, using either of the two methods. (6pts)

- LU decomposition, with partial/row pivoting.
- QR decomposition using Householder transformation.

(2). From your decomposition, solve $A\boldsymbol{x} = \boldsymbol{b}$ where $b_i = \sum_{j=1}^{n}(i+j-1)^{-1}$ so that the exact solution should be $\boldsymbol{x} = (1, 1, \ldots, 1)^T$. Take $n = 5$ for this question, and show results for both single and double precisions. (5pts)

(3). At what n will your method become unstable for single and double precisions (in this problem, let us say you find a more than $\sim 50\%$ relative error in terms of a vector norm)? (3pts)

(4). Calculate the $\infty-$norm of A for $n = 3, 6, 9, 12$. How are the results related to your observations in (3)? (6pts)

(5). Can you suggest and/or demonstrate ways to improve the situation? (Bonus, up to 3 additional pts; but the total score still caps at 20 pts)