CSE Ph.D. Qualifying Exam, Spring 2014: Numerical Analysis

There are four problems below. Please choose three to solve. If you choose to solve all four, only the lowest three scores will count. Show all your work and write in a readable way.

1. In this question, you are asked to write an efficient algorithm for computing the cube root $y^{1/3}$ of any given positive number y. Your algorithm must be designed for a base 2 floating point system, where any floating point number can be represented as $y = a \times 2^e$, where a is a normalized fraction ($0.5 \le a < 1$) and e is an integer exponent. For efficiency, you may store some useful constants ahead of computation time, e.g., $2^{1/3}$, $2/3$, and $a/3$, assuming these are useful.

   (a) Show how $y^{1/3}$ can be obtained once $a^{1/3}$ has been calculated for the corresponding fraction, in as little as five additional flops.

   (b) Derive the corresponding Newton iteration. What is the flop count per iteration?

   (c) How would you choose an initial approximation? Roughly how many iterations are needed if the machine rounding unit is 2^{-52}?

2. Consider the following one-step implicit method for solving the ODE $y' = f(t, y)$

$$y_{k+1} = y_k + \frac{h}{2}(3f(t_k, y_k) - f(t_{k+1}, y_{k+1}))$$

   By applying the above method to the model problem $y' = \lambda y$ with $\lambda < 0$, determine

   (a) what is its order of accuracy?

   (b) what is its stability region? You need to determine an interval $(0, a)$ such that when $h > 0$ belongs to that interval, the method is stable.

3. (a) Suppose that A is a complex $m \times n$ dense, rank-r matrix, where $r \ll m, n$. Propose an algorithm which runs in at most $O(rmn)$ time for computing its *thin* Singular Value Decomposition,

$$A = U \Sigma V^H,$$

   where U is $m \times r$, Σ is $r \times r$, and V is $n \times r$.

   (b) Once you have the factorization, what is the complexity of applying the factored matrix to a vector?

   (c) What is the cost of explicitly forming A^2?

   (d) Suppose that A and B are both $m \times n$ and of rank r. Propose an algorithm (and explain its cost) for forming the *best* (in the two-norm sense) rank-r approximation to $A + B$?

4. (a) Derive a Newton iteration for computing the *matrix sign function*,

$$\mathrm{sign}(A) = A(A^2)^{-1/2},$$

which is the matrix generalization of the scalar function

$$\mathrm{sign}(z) = \begin{cases} 1, & \mathrm{Re}z > 0, \\ -1, & \mathrm{Re}z < 0 \end{cases}$$

and is valid as long as A does not have any pure-imaginary eigenvalues. [Hint: use the fact that $S = \mathrm{sgn}(A)$ satisfies $S^2 = I$ and use A as the initial state].

(b) Prove that the iteration always converges (provided that A does not have any pure imaginary eigenvalues).

(c) Provide the number of operations required for each iteration of the algorithm.

(d) Explain what is numerically worrisome about the iteration and justify a choice of stopping criteria [Note: there is no universally agreed-upon answer for the latter].