CS 322 Prelim 2

Thursday 5 April 2006—90 minutes

Problem 1: Floating Point Representations (20 pts)

We have decided to go to the IEEE with a proposal for a new floating point format that we want to have ratified as a standard. This format will be 16 bits long and will consist of a sign bit, 8 exponent bits, and 7 mantissa bits. Otherwise, it will behave just like all other IEEE floating point numbers do – it will have an implicit leading 1 bit, normalized numbers except for the 0 exponent, and a bias on the exponent of 127.

1. How would the number 4.375 be represented in this format?
2. What floating point operation is simplified when using biased exponents versus using 2's complement exponents?
3. What is the value of epsilon for 4.375?
4. What is the value of epsilon for 2^{-130}?
5. What is the smallest positive number in this representation such that the value of epsilon is 1?
6. Recall that half precision is another proposed IEEE 16-bit floating point spec, with 5 exponent bits and 10 mantissa bits. Give a reason why we might prefer each data format; this can be a reason, application, or advantage of each format.

Problem 2: Linear algebra computations (30 pts)

Three matrix factorizations we've studied are LU, QR, and SVD.

Part A

1. Which of these methods do and don't require pivoting to be stable for nonsingular problems?
2. Order the three methods by the amount of arithmetic (using flops as a rough count) required to compute them, from least arithmetic to most arithmetic

Part B

For the following questions, the basic tools you are allowed to consider are

- matrix and vector addition
- matrix multiplication (multiplication of matrices with matrices or vectors),
- forward and back substitution
- vector scaling (i.e. multiplying vectors with .* in matlab)
- outer and inner vector products

1. Using the basic tools above, what sequence of operations is required to solve a linear system using (a) QR and (b) SVD, assuming that the relevant factorization has already been performed? For example, with LU you need a back substitution followed by a forward substitution.
2. Using the basic tools above, what sequence of operations is required to solve a full-rank $m \times n$ linear least squares system using (a) QR and (b) SVD, again assuming the relevant factorization has already been performed? Indicate what size vector or matrix is involved in each step. For example, with LU you would do an $n \times m$ matrix-vector multiply followed by an $n \times n$ back substitution and an $n \times n$ forward substitution (assuming you factored $\mathbf{A}^T\mathbf{A}$).
3. Using the basic tools above, what sequence of operations is required to compute the k-th step in the QR factorization using Householder transforms (assuming you are just computing $\mathbf{R}$)? Indicate what size vector or matrix is involved in each step.
4. Which factorizations can be used to solve a rank-r $m \times n$ linear system, with $r < n$? Again using the tools above, what sequence and sizes of operations are required?

Problem 3: Numerical Rank (25 pts)

A matrix is found to have singular values $10, 1, 0.1, 0.1, 0.1, 10^{-9}, 10^{-9}$.

Give a one sentence explanation of your reasoning in the questions below.

1. What is the rank of this matrix, according to a mathematician?
2. What is likely the true rank if it's represented in double-precision floating point?
3. What is likely the true rank if it's represented in single-precision floating point?
4. What is likely the true rank if the matrix is 20×7 and corrupted by noise at most 0.02 in each entry?

Problem 4: Root Finding (25 pts)

Consider the root finding methods (a) bisection, (b) Newton's method, and (c) the secant method.

1. Which methods require code to compute the derivatives of the function?
2. Which methods are guaranteed to converge?
3. Which method has the fastest convergence once near a root?
4. Which method has the slowest convergence?

You can express your answer as a 4×3 binary table like so:

	bisection	Newton's	secant
requires derivative			
guaranteed convergence			
fastest convergence around root			
slowest convergence			