**Ma221 - Final Exam - Fall 1990 - 3 hours - Open notes, closed book**
**Choose 4 questions out of 6. Clearly indicate which 4 of the 6 you want graded. Write your name on each separate piece of paper you hand in. Brief answers are preferred!**

(1) Let $X$ be the space of problems we want to solve, $Y$ the space of solutions and $f : X \rightarrow Y$ the solution map. In other words $y = f(x)$ is the solution of problem $x \in X$. Let $alg : X \rightarrow Y$ be the algorithm. In other words $alg(x)$ is the computed solution of the problem $x$, including the effects of rounding errors. Using this notation, say what it means for the algorithm $alg$ to be *backward stable*. Again using this notation, and assuming $f$ is differentiable, define a condition number. Show how to use a backward error bound and condition number to bound the error $\|alg(x) - f(x)\|/\|f(x)\|$.

(2) In order to analyze the effects of rounding errors, we have used the following model:

$$fl(a \oplus b) = (a \oplus b)(1 + \delta)$$

where $\oplus$ is one of the four basic operations $+$, $-$, $*$, and $/$, and $|\delta| \leq \epsilon$, where $\epsilon$ is a constant called *machine precision*. To extend all the analyses we did this semester to *complex* data, we need to prove that the above formula holds for the four basic complex operations. Now $\delta$ will be a tiny *complex* number bounded in absolute value by a small multiple of $\epsilon$. Prove this is true for complex addition, subtraction and multiplication. Is it true that both the real and imaginary parts of the complex product are always computed to high relative accuracy? Extra Credit: Prove it for complex division.

(3) Let $A$ and $B$ be symmetric tridiagonal matrices, with $B$ positive definite. Give an algorithm for finding all the eigenvalues of the matrix pencil $A - \lambda B$. Efficiency is important: full credit will be given only for an algorithm whose running time is proportional to $n^2$ (and possibly other factors not depending on $n$). Describe the algorithm at a high level in terms of the algorithms we discussed during the semester. Explain briefly why your algorithm works and why it is numerically stable. Describe how you would develop a high quality subroutine for this problem with as little programming effort as possible.

(4) Describe how you would solve the model problem (the Poisson equation on a square) using only FFTs and divisions, i.e. without solving tridiagonal systems. What is the running time of this algorithm? Big-O notation is adequate.

(5) Suppose you are collecting experimental data, and want to use some of the data to predict other parts of the data. In particular, suppose we do $m$ experiments, and in experiment $i$ we measure values $a_{i1}, \ldots, a_{in}$ and $b_i$. We want to predict $b_i$ as a function $f(a_{i1}, \ldots, a_{in})$, and choose $f$ to minimize $\sum_{i=1}^{m} w_i(b_i - f(a_{i1}, \ldots, a_{in}))^2$, where $w_i > 0$ is a weight measuring the quality of the data in the $i$-th experiment.
(5a) Describe an algorithm to solve this problem when we restrict $f$ to be a linear combination

$$f(a_{i1}, \ldots, a_{in})) = \sum_{j=1}^{n} \beta_j a_{ij}$$

How does your solution depend on the relative sizes of $n$ and $m$?
(5b) Describe an algorithm to solve this problem when we let $f$ be an arbitrary quadratic polynomial

$$\alpha + \sum_{j=1}^{n} \beta_j a_{ij} + \sum_{j,k=1}^{n} \gamma_{jk} a_{ij} a_{ik}$$

Extra Credit: Suppose you wanted $f$ to have a maximum number of terms $t$, with $t$ less than $n$ in part 5a or less than $n^2 + n + 1$ in part 5b. How would you efficiently pick a good set of $t$ terms to use? That is, without trying all possible subsets of $t$ terms?

(6) A matrix $A$ is called strictly column diagonally dominant if $|a_{ii}| > \sum_{j \neq i} |a_{ji}|$ for all $i$. Show that such an $A$ is nonsingular. Show that no pivoting is needed when computing $A = LU$. In other words, if we did do partial pivoting to compute $PA = LU$, $P$ a permutation matrix, then we would get $P = I$. Hint: Show that after one step of Gaussian elimination, the bottom right $n - 1$ by $n - 1$ submatrix is also strictly column diagonally dominant.