CS 4220: Review Probs

1.

(a) The unit roundoff EPS is approximately the smallest floating point number x such that $1+x$ is greater than 1 in floating point arithmetic. How is EPS related to the floating point mantissa length?

(b) Assuming that a and b is are initialized floating point numbers with positive value, complete the following conditional so that the message is printed if b is (roughly) less than EPS*a. Do *not* reference the built-in constant EPS.

```
if ____________________________________________

    disp('b is small compared to a')
end
```

(c) Why is the SVD a reliable method for exploring how near a matrix is to being rank-deficient?

2.

(a) We say $A \in \mathbb{R}^{n\times n}$ is *stochastic* if it has positive entries and it has unit column sums. It is known that if A is stochastic then its dominant eigenvalue λ_{max} equals 1 and there is a unique $x \in \mathbb{R}^n$ with positive entries so $Ax = x$. Explain why small changes in A induce corresponding small changes in λ_{max}.

(b) Complete the following function so that it performs as specified:

```
  function x = LowerHessSolve(H,b)
% H is a nonsingular n-by-n matrix with H(i,j) = 0 whenever j>i+1.
% b is a column n-vector
% x satisfies Hx = b.
```

Make effective use of the LU-with pivoting factorization.

3.

If $C \in \mathbb{R}^{n\times n}$ is nonsingular and $u, v \in \mathbb{R}^n$, then the Sherman-Morrison formula gives a recipe for the inverse of $(C + uv^T)$ if this matrix is nonsingular. In particular,

$$(C + uv^T)^{-1} = C^{-1} + \alpha w z^T$$

where $w = C^{-1}u$, $z^T = v^T C^{-1}$, and $\alpha = -1/(1 + v^T C^{-1} u)$.

By making effective use of the QR factorization, complete the following function so that it performs as specified:

```
  function [x,z] = DoubleSolve(A,f,g,b)
% A is a nonsingular n-by-n matrix.
% f, g, and b are column n-vectors.
% x and z are column n-vectors with the property that Ax = b and
% (A + f*g')z = b. (Assume the latter system is nonsingular.)
```

You may use the backslash operator to solve triangular systems.

4.

The Jacobi method for computing the Schur decomposition of a real symmetric matrix $A \in \mathbb{R}^{n\times n}$ is based on solving a sequence 2-by-2 subproblems.

(a) Explain why each subproblem involves $O(n)$ flops.

(b) What makes the method attractive for parallel computation?

5.

For each of the following methods, draw a picture that communicates the main idea behind a step. No formulas are necessary. Just a labeled sketch that graphically indicates how the next iterate is obtained. (Such a picture for Newton's method would show the linear model and label its zero.)

(a) The Secant method for finding a zero of $f{:}\mathbb{R} \rightarrow \mathbb{R}$.

(b) The Golden Section search method for finding a minimum of $f{:}\mathbb{R} \rightarrow \mathbb{R}$ on $[L, R]$ assuming that f'' is always positive.

(c) The steepest descent method with exact line search for finding a minimum of $f{:}\mathbb{R}^2 \rightarrow \mathbb{R}$. (Draw contours.)

6.

Consider the objective function

$$\phi(x) = \frac{1}{2}F(x)^T F(x) \;=\; \frac{1}{2}\sum_{i=1}^{m} F_i(x)^2$$

where $F(x) = [F_1, \ldots, F_m(x)]^T$ and F_i: $\mathbb{R}^n \rightarrow \mathbb{R}$. If $J_c = J(x_c) \in \mathbb{R}^{m\times n}$ is the Jacobian of F at x_c then the gradient of ϕ at x_c is given by

$$g_c = J(x_c)^T F_c \qquad F_c = F(x_c$$

and its Hessian by

$$H_c = J_c^T J_c \;+\; \sum_{i=1}^{m} F_i(x_c)\nabla^2 F_i(x_c)$$

Refer to the summation as the "messy part of the Hessian." From calculus we know that

$$\phi(x_c + s) \;=\; \phi(x_c) + s^T g_c + \frac{1}{2}s^T H_c s \;+\; \ldots$$

(a) The Gauss-Newton method chooses s by minimizing the quadratic model $q_c(s) = \phi(x_c) + s^T g_c + \frac{1}{2}s^T J_c^T J_c s$. Explain why this leads to a linear least squares problem.

(b) Even though the Gauss-Newton method ignores the messy part of the Hessian, it may still converge quadratically for certain types of nonlinear least squares problems. Explain.

(c) The Levenberg-Marquardt method replaces the messy part of the Hessian with $\mu^2 I_n$. In particular, the step s is determined by minimizing $q_c(s) = \phi(x_c) + s^T g_c + \frac{1}{2}s^T(J_c^T J_c + \mu^2 I)s$. How does the least squares problem in (a) change and why does the SVD make it easy to characterize the minimizing s as a function of μ?

7.

Suppose $t, x, y \in \mathbb{R}^m$ are given and that we wish to determine $\rho^T = [\rho_1, \ldots, \rho_k]$ and $r^T = [r_1, \ldots, r_k]$ so that

$$\phi(r,\rho) = \sum_{i=1}^{m}\left(x_i - \sum_{j=1}^{k} r_j \cos\left(\frac{2\pi t_i}{\rho_j}\right)\right)^2 + \sum_{i=1}^{m}\left(y_i - \sum_{j=1}^{k} r_j \sin\left(\frac{2\pi t_i}{\rho_j}\right)\right)^2$$

is minimized.

(a) Define ϕ in the "language" of matrices, vectors, and norms instead of the language of summations and subscripts.

(b) Explain how the problem of minimizing ϕ can be approached as a k-parameter nonlinear least squares problem. Be sure to include in your explanation a definition of the objective function. (You do not have to discuss its implementation.)

8.

(a) A 3-by-3 linear system with infinity-norm condition 10^8 is solved via the MATLAB backslash operator \ on a computer with unit roundoff 10^{-17}. Here is the computed solution:

```
x(1) = 1234.5678901234567

x(2) =    1.2345678901234567

x(3) =     .00012345678901234567
```

Underline the digits that are most likely correct and explain why. Recall that $\| v \|_\infty = \max |v_i|$.

(b) We wish to do a least squares fitting of a function of the form $f(t) = \alpha + \beta e^{\lambda t}$ to the data $(t_1, y_1), \ldots, (t_m, y_m)$. Assume that $y_1 > y_2 > \cdots > y_m$ and $0 < t_1 < t_2 < \cdots < t_m$ with $m > 2$. Explain why small relative changes in the data *might* induce large relative changes in the optimal fitting function. Be brief.

9.

(a) Suppose $A \in \mathbb{R}^{n\times n}$ is symmetric and positive definite, $b \in \mathbb{R}^n$, and $h > 0$. We wish to evaluate

$$f(z) = b^T(A + zI)^{-1}b$$

for $z = h,\ 2h, \ldots, mh$ where $m >> n$. Complete the following function so that it performs as specified.

```
  function fVals = ManySolve(A,b,h,m)
% A is an n-by-n symmetric positive definite matrix and b is n-by-1.
% h>0 and m is a positive integer.
% fVals is a column m-vector with fVals(k) = b'*inv(A+khI)*b, k=1:m
```

Hint: Picking the right factorization can make each f-evaluation $O(n)$.

(b) Suppose $A \in \mathbb{R}^{n\times n}$ is symmetric and positive definite and $c, d \in \mathbb{R}^n$. We wish to solve the following linear system for $y, z \in \mathbb{R}^n$:

$$\begin{bmatrix} A & A \\ A & -A \end{bmatrix} \begin{bmatrix} y \\ z \end{bmatrix} = \begin{bmatrix} c \\ d \end{bmatrix}$$

Write a Matlab function `[y,z] = BigSolve(A,c,d)` that does this. Do not use the backslash operator \ except to solve triangular systems.

10.

For each of the following methods, draw a picture that communicates the main idea behind a step. No formulas are necessary. Just a labeled sketch that graphically indicates how the next iterate is obtained. (Such a picture for Newton's method would show the linear model and label its zero.)

(a) The Secant method for finding a zero of $f:\mathbb{R} \rightarrow \mathbb{R}$.

(b) The Golden Section search method for finding a minimum of $f:\mathbb{R} \rightarrow \mathbb{R}$ on $[L, R]$ assuming that f'' is always positive.

(c) The steepest descent method with exact line search for finding a minimum of $f:\mathbb{R}^2 \rightarrow \mathbb{R}$. (Draw contours.)

11.

(a) Assume that δ is greater than the unit roundoff and that each entry in a data matrix $A \in \mathbb{R}^{m\times n}$ has relative error $\approx \delta$. How would you estimate the rank of A from the QR-with-column-pivoting factorization? Recall that in that factorization r_{kk} is the largest entry in the submatrix $R(k{:}m, k{:}n)$. Justify your answer by explaining the amount of error that we can expect in the computed R. Order-of-magnitude reasoning is absolutely fine.

(b) When solving a rank-deficient least squares problem, why might one prefer QR-with-column-pivoting method to the singular value decomposition method? Be brief.

12.

To find a zero x_* of a function f we can apply Newton's method:

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)} \qquad k = 0,\ 1, \ldots$$

It can be shown that

$$|x_{k+1} - x_*| \;=\; \left|\frac{f''(\eta)}{f'(x_k)}\right| \, |x_k - x_*|^2$$

where η is in between x_* and x_k.

(a) This means that the order of convergence for Newton's method is 2. The secant method has order ≈ 1.6. Does this mean that the Newton's method will reqire fewer iterations? Explain.

(b) Assume that $|f'(x)| \geq \delta > 0$ for all x and that $|f''(x)| \leq M_2$ for all x. Show that the Newton iteration converges to x_* if x_0 is "close enough" to x_*. Be precise about "close enough".

Some MATLAB Functions

LU Factorization

```
[L,U,P] = LU(X) returns unit lower triangular matrix L, upper
triangular matrix U, and permutation matrix P so that
P*X = L*U.
```

Cholesky Factorization

```
R = CHOL(X)  returns an upper triangular R so that R'*R = X
where X is symmetric and positive definite.
```

QR Factorization

```
[Q,R,E] = QR(A) produces unitary Q, upper triangular R and a
permutation matrix E so that A*E = Q*R. The column permutation E is
chosen so that ABS(DIAG(R)) is decreasing.
```

Singular Value Decomposition

```
[U,S,V] = SVD(X) produces a diagonal matrix S, of the same
dimension as X and with nonnegative diagonal elements in
decreasing order, and unitary matrices U and V so that
X = U*S*V'.
```

Schur Decomposition

```
[U,D] = SCHUR(X) produces a diagonal matrix D and
an orthogonal matrix U so that X = U*D*U' assuming that X
is real and symmetric.
```