CS 4220: Final Exam Solutions

May 14, 2013

1. (10 points)

(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:

```
xHat(1) = 1234.5678901234567

xHat(2) =    1.2345678901234567

xHat(3) =     .00012345678901234567
```

Underline the digits that are most likely correct and justify your answer by explaining how the relative error $\| \hat{x} - x \|_\infty / \| x \|_\infty$ depends on both the condition and the unit roundoff. Recall that $\| v \|_\infty = \max |v_i|$.

Solution (5 points)

Since

$$\frac{\| \hat{x} - x \|_\infty}{\| x \|_\infty} \approx \text{cond}_\infty(A) \text{ eps}$$

we have for this example that

$$|\hat{x}_i \; - \; x_i| \leq \| \hat{x} - x \|_\infty \; \leq \; \text{cond}_\infty(A) \text{ eps} \| x \|_\infty \approx 10^8)(10^{-17})(10^3) \approx 10^{-6}$$

So underline each component value through the sixth decimal place.

-2 if you underline 6 significant digits in each component. The basic heuristic $\| \hat{x} - x \| / normx \approx \text{cond}(A) eps = 10^{-d}$ says the *vector* $\hat{x}$ has about d correct digits. That does not translate into d correct digits for every component, except the largest one.

(b) Assume that `a` and `b` is are initialized floating point numbers with positive value and that the message "`b is small compared to a`" is displayed when the following code is executed:

```
if a + b == a

    disp('b is small compared to a')

end
```

What can you say about the actual magnitude of `b`?

Solution (5 points)

b must be less than the spacing of the floating point numbers at a, so roughly $|b| < \text{eps } a$ where eps is the unit roundoff.

-2 if you say $b \approx eps$. For example $\text{fl}(1 + 2^{100}) = 2^{100}$.

2. (20 points)

(a) Show how the SVD can be used to solve the linear system

$$(A^T A + \mu I)x \ = \ A^T b$$

where $A \in \mathbb{R}^{m \times n}$, $n \leq m$, $\mu > 0$, and $b \in \mathbb{R}^m$. Answer by completing the following code in MATLAB:

```
% A, b, mu defined
[U,S,V] = svd(A);
[m,n] = size(A);
```

Solution (10 points)

```
% (USV')'(USV') + muI)x = (USV')b   ---->

% (VS'SV' + mu*I)x = VS'U'b  --->

%  V(S'S + mu*I)V'x = VS'U'b

% (S'S + mu*I)y = btilde   where btilde = S'(U'*b) and  y = V'x

d = diag(S);


btilde  = d.*(U(:,1:n)'*b);    % 3 points for transformed rhs

y = btilde ./ (d.^2 + mu);     % 4 points for solution of transformed system

x = V*y                        % 3 points for transforming back to get x
```

S is not square so -3 for things that involve S^2.
-3 for dimension incompatibility, e.g., `diag(S)*(U'*b)`
-5 for correct but with an $O(n^3)$ computation, e.g. `V'*A'*b` instead of `V'*(A'*b)`

(b) Suppose $A = QR$ is the QR factorization of a matrix $A \in \mathbb{R}^{m\times n}$. Assume that $R(k,k) = 0$ for some $k > 1$ and that all other entries along R's diagonal are nonzero. Show how to compute a nonzero vector $x \in \mathbb{R}^n$ so that $Ax = 0$. You may use the \ operator to solve triangular systems. Answer by completing the following code in MATLAB:

```
% A defined
[Q,R] = qr(A);
[m,n] = size(A);
```

Solution (10 points)

```
% Ax = 0 means QRx = 0 means Rx = 0
% 3 points for recognizing that that need null vector for R


% R(1:k,1:k) is singular with zero in its lower right corner.
% 2 points for this observation

y = R(1:k-1,1:k-1)\R(1:k-1,k);    % [y;-1] is in the nullspace of R(1:k,1:k)
% 3 points for this

x = [y;-1;zeros(n-k,1)]
% 2 points
```

-3 if correct but you set $k = n$.

3. (15 points) If $C \in \mathbb{R}^{n \times n}$ is symmetric and positive definite and $u \in \mathbb{R}^n$, then

$$(C + uu^T)^{-1} = C^{-1} + \alpha vv^T$$

where $v = C^{-1}u$ and $\alpha = -1/(1 + u^T C^{-1} u)$. By making effective use of the Cholesky factorization and the above math fact, complete the following function so that it performs as specified:

```
  function [x,z] = DoubleSolve(A,u,b)
% A is a symmetric positive definite n-by-n matrix.
% u and b are column n-vectors.
% x and z are column n-vectors with the property that Ax = b and
% (A + u*u')z = b.
```

You may use the $\backslash$ operator to solve triangular systems.

Solution (15 points)

```
%  Solve Ax = b using Cholesky for 5 points..
G = chol(A,'lower');
x = G'\(G\b);

%  Solve Av = u for 4 points
v = G'\(G\u);


% 6 points for using the math fact without inverse computation...
alfa = -1/(1+u'*v);
% z = (inv(A) + alfa*v*v')b = inv(A)*b + alfa*(v'*b)*v
z = x + alfa*(v'*b)*v
```

-4 if you have unnecessary solves with G
-2 for `(v*v')*x` instead of `(v'*x)*v`
-10 for any kind of inverse computation, e.g., `inv(A)` or `A\ eye(n,n)`.

4. (15 points) Short answer.

(a) Assume that $A \in \mathbb{R}^{n\times n}$ is symmetric with distinct eigenvalues. Describe in English what the power method would try to compute when applied to the matrix $(A-\mu I)^{-1}$.

Solution (5 points)

If λ is the closest eigenvalue of A to μ and $Ax = \lambda x$, then $\lambda_* = 1/(\lambda-\mu)$ is the dominant eigenvalue of $(A-\mu I)^{-1}$ and $x_* = x$ is the corresponding eigenvector. The power method will try to find λ_* and x_*. If μ is equidistant to two eigenvalues then a problem.

(b) After k exact-arithmetic steps with starting unit 2-norm vector q_1 we assume that the Lanczos method computes part of the decomposition $Q^T AQ = T$ where Q is orthogonal with $Q(:,1) = q_1$ and T is tridiagonal. Explain. What makes the method a "sparse matrix friendly"? What is the method typically used for?

Solution (5 points)

Will have $Q(:,1{:}k)$ and $T(1{:}k,1{:}k)$. (2 points)

Uses only matrix-vector products and the last two Lanczos vectors (2 points)

The extremal eigenvalues of T_k are good approximations for the extremal eigenvalues of A. (1 point)

(c) Any symmetric positive definite matrix A has a Cholesky factorization $A = GG^T$ where G is lower triangular. Why does MATLAB sparse Cholesky software compute the factorization $PAP^T = GG^T$ where P is a permutation matrix?

Solution (5 points)

P is chosen to minimize fill-in in the Cholesky factor

5. (15 points) 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 $f\colon \mathbb{R} \to \mathbb{R}$.

The picture should show show the secant line that goes through $(x_k, f(x_k)$ and $(x_{k-1}, f(x_{k-1}))$. (3 points). The intersection of that line with the x-axis defones x_{k+1}.

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

The picture should show a function $f(x)$ with $f''(x) > 0$ across the search interval $[L, R]$. (2 points). It should show two sample points c and d in the interval and a comparison based on comparison of $f(c)$ and $f(d)$. (1 point). It should show reduction of the search interval and the reuse of either $f(c)$ of $f(d)$ in th enext step (2 points)

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

Should show the negative gradient direction (2 points) and give a clue that one tries to minimize f in that direction.

6. (15 points)

(a) Consider the following MATLAB script for approximating the square root of a positive real number A:

```
L = A;
W = 1;
for k=1:10
   L = (L+W)/2;
   W = A/L;
end
```

Explain why this is essentially an instance of Newton's method.

Solution (10 points)

Equivalent to

```
L = A;
for k=1:10
   L = (L+A/L)/2;
end
```

This is Newton's method applied to $f(L) = L^2 - A$:

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)} = x_k - \frac{x_k^2 - A}{2x_k} = \frac{1}{2}\left(x_k + \frac{A}{x_k}\right)$$

At least -5 if you thing that $f(x) = \sqrt{x}$.

(b) Newton's method is applied to a function f:$\mathbb{R} \to \mathbb{R}$ that has a single root x_*. Here is a heuristic result that relates the error at step $k+1$ to the error at step k:

$$|x_{k+1} - x_*| \approx \frac{|f''(x_k)|}{2|f'(x_k)|}|x_k - x_*|^2.$$

What additional information is required before one can be confident that the iteration converges quadratically for a particular initial guess x_0?

Solution

The derivative must be bounded away from zero.
x_0 must be close enough. How close depends on the second derivative (the bigger the closer) and the first derivative (the small the closer) .
Can also say that $|f''(x)/f'x)|$ bounded by M and $|x_0 - x_*| \cdot M < 1$

7. (15 points) Suppose $t, x, y \in \mathbb{R}^m$ are given and that we wish to use `lsqnonlin` to determine $a^T = [a_1, \ldots, a_k]$ and $\rho^T = [\rho_1, \ldots, \rho_k]$ so that

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

is minimized. To use `lsqnonlin` we must design a vector-valued function $R(a,\rho)$ with the property that

$$\phi(a,\rho) \;=\; \frac{1}{2}R(a,\rho)^T R(a,\rho).$$

Express $R(a,\rho)$ in the form

$$R(a,\rho) \;=\; \text{Matrix} \,\times\, \{\text{vector}\,\} \;-\; \{\,\text{vector}\,\}$$

Clearly define the matrix and the two vectors.

Solution

Since

$$\sum_{i=1}^{m}\left(x_i - \sum_{j=1}^{k} a_j \cos\left(\frac{2\pi t_i}{\rho_j}\right)\right)^2 \;=\; \|\, Ca - x \,\|_2^2$$

where $C = (c_{ij}) \in \mathbb{R}^{m\times k}$ and $c_{ij} = \cos(2\pi t_i/\rho_j)$ and

$$\sum_{i=1}^{m}\left(y_i - \sum_{j=1}^{k} a_j \sin\left(\frac{2\pi t_i}{\rho_j}\right)\right)^2 \;=\; \|\, Sa - y \,\|_2^2$$

where $C = (c_{ij}) \in \mathbb{R}^{m\times k}$ and $c_{ij} = \cos(2\pi t_i/\rho_j)$ we have

$$\phi(a,\rho) \;=\; \frac{1}{2}\|\, Ca - x \,\|_2^2 \;+\; \frac{1}{2}\|\, Sa - y \,\|_2^2 \;=\; \frac{1}{2}\left\|\, \left[\begin{array}{c} C \\ S \end{array}\right] a \;-\; \left[\begin{array}{c} x \\ y \end{array}\right] \,\right\|_2^2$$

Thus,

$$R \;=\; \left[\begin{array}{c} C \\ S \end{array}\right] a \;-\; \left[\begin{array}{c} x \\ y \end{array}\right]$$

5 points for the matrix
3 points for the rhs
2 points for the vector that is multiplied by the matrix.
Up to -7 for stuff like $[C\ S][a\,;\ a] = x + y$