Fall 2006 CS 421: Final Exam Solution Guide

Final Exam Results

95-100	x
90-94	
85-89	xxxxxx
80-84	xxx
75-79	x
70-74	x
65-69	xxx
60-64	x
50-59	xxx

	Max	Average
Problem 1	15 points	10.6
Problem 2	15 points	11.8
Problem 3	15 points	10.0
Problem 4	20 points	17.8
Problem 5	15 points	13.3
Problem 6	10 points	6.2
Problem 7	10 points	6.6
	100 points	76.4

Course Score

Recalling that HW = 40%, Pre1 = 15%, Pre2 = 15%, and Final = %30,

$$\begin{aligned}\text{Total Score} \;=\; & 40 \cdot ((A1/25 + A2/15 + A3/15 + A4/15 + A5/20 + A6/10)/6) \;+ \\ & 15 \cdot (\text{Pre1}/100) \;+ \\ & 15 \cdot (\text{Pre2}/100) \;+ \\ & 30 \cdot (\text{Final}/100)\end{aligned}$$

Class Ave = 80.5

1. (15 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:

```
x(1) = 1234.5678901234567

x(2) =    1.2345678901234567

x(3) =     .00012345678901234567
```

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

Solution

4 points for the basic heuristic:

$$\| \hat{x} - x \|_\infty / \| x \|_\infty \;\approx\; \epsilon \kappa_\infty(A) \;=\; 10^{-17} \cdot 10^8 = 10^{-9}$$

.

2 points if you think this means 9 digits correct in each component

```
x(1) = 1234.56789
x(2) =    1.23456789
x(3) =     .000123456789
```

4 points if you see this means absolute error of 10^{-6} in each component:

$$|\hat{x}_i - x_i| \leq \| \hat{x} - x \|_\infty \approx \| x \|_\infty 10^{-9} \approx 10^{-6}$$

So

```
x(1) = 1234.567890
x(2) =    1.234567
x(3) =     .000123
```

(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.

Solution

6 points

You have to same something about where the data is sampled. If the sampling points are too far out, then the data will be very flat and it will be very hard to determine the time constant λ. Small changes in the y_i out in the tail etc. Or if the t_i are very close together. Or you can look at the $m-by-2$ matrix $[ones(m,1)exp(\lambda t)]$ and talk about the columns "looking alike".

2. (15 points)

Consider the Simpson rule and its error

$$\int_c^d f(x)dx \;=\; \frac{d-c}{6}\left(f(c) + 4f\left(\frac{c+d}{2}\right) + f(d)\right) \;+\; \frac{(d-c)^5}{2880} f^{(4)}(\eta) \qquad c \leq \eta \leq d$$

Let S_N denote the composite Simpson rule estimate of

$$I = \int_a^b f(x)dx$$

with N equally spaced intervals.

(a) Give an upper bound for $|I - S_N|$ assuming that $|f^{(4)}(x)| \leq M_4$ on the interval $[a, b]$.

Solution

7 points

If $h = (b-a)/N$, then the error on one subinterval $\leq M_4 h^5/2880$. The sum of these errors is $\leq M_4(b-a)h^4/2880$

(b) Assume that `a`, `b`, and `N` are available. Write a vectorized MATLAB fragment that assigns to `Q` the value of S_N for the function $f(x) = x^3/(1 + |\sin(x)| \exp(-x))$.

Solution

8 points

```
% 2 points Getting the sampling points right...
h = (b-a)/N;
x = linspace(a,b,2*N+1);

% 3 points for vectorized evaluation of f...  (-2 if not vectorized)
f = x.^3 ./ (1 + abs(sin(x)).*exp(-x));

% 3 points for correct summation with weights... (-1 if not vectorized
Q = (h/6)(f(1) + f(2*N+1) + 4*f(2:2:2*N) + 2*f(3:2:2*N-1));
```

3. (15 points)

Both parts of this problem are concerned with the initial value problem $\dot{y} = Ay$ with $y(0) = y_0$. Assume that all the eigenvalues of $A \in \mathbb{R}^{n\times n}$ are in the left half plane.

(a) The second order Adams-Moulton method for the problem $\dot{y} = f(t, y(t))$ is given by

$$y_{n+1} = y_n + \frac{h}{2}\left(f(t_n, y_n) + f(t_{n+1}, y_{n+1})\right)$$

How would you use this method to compute estimates of $x(h)$, $x(2h), \ldots, x(mh)$ for a given steplength h? A complete MATLAB implementation is not necessary. Just explain the calculation associated with each step.

Solution

10 points

Since

$$y_{n+1} = y_n + (h/2)(Ay_n + Ay_{n+1}) \Rightarrow (I - (h/2)A)y_{n+1} = (I + (h/2)A)y_n$$

we can compute $P((I - (h/2)A)) = LU$ once and then each y_n costs $O(n^2)$.

Also full credit for reasonable predictor-corrector approach since AM2 is implicit. We need a way of estimating y_{n+1} before invoking the AM2 formula. E.g.,

$$\tilde{y}_{n+1} = \text{via some explicit method}$$

then

$$y_{n+1} = y_n + \frac{h}{2}\left(Ay_n + A\tilde{y}_{n+1}\right)$$

(b) If $\tilde{A}$ has an eigenvalue with nonnegative real part, then the solution to $\dot{x} = \tilde{A}x$ may not decay which is a "bad" situation. Why does this prompt us to consider the function $f(\mu) = \sigma_{min}(A + \mu iI)$? Here, $\sigma_{min}(\cdot)$ designates the minimum singular value.

Solution

(5 points)

There is a matrix $\tilde{A}$ with eigenvalues on the imaginary axis with the property that $\| A - \tilde{A} \| = f_{min}$. So if this number is small, a small change in A can put us in a bad situation. Need to talk about small changes in A that move one of its eigenvalues to the rh plane.

4. (20 points)

(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 $O(n)$.

Solution

10 points

```
% 5 points for using schur and transforming the problem..
% If A = QDQ' then f(z) = b'(QDQ' +khI)^-1 b  = c' inv(D + khI) c  where c = Q'b
[Q,D] = schur(A);
c = Q'b;

% 5 points for Solving the transformed system
fvals = zeros(m,1);
for k=1:m
   fVals(k) = sum((c.^2)./(diag(D)+k*h));
end
```

(b)

10 points

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. Your implementation must explicitly use `lu`. Do not use the backslash operator \ except to solve triangular systems.

Solution

```
function [y,z] = BigSolve(A,c,d)
% A(y+z) = c  and A(y-z) = d.
R = chol(A);
f = R\(R'\c);    % f = y+z
g = R\(R'\d);    % g = y-z
y = (f+g)/2
z = (f-g)/2
```

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:$\mathbb{R} \to \mathbb{R}$.

See page 232 in book. (5 points)

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

See page 271 in book. (5 points) -1 or -2 if you dont indicate what the next interval is.

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

See page 278 in book. (5 points) -1 if your step picture doesn't indicate how far along you go in the neg grad direction.

6. (10 points)

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.

Solution

5 points

The R-matrix will have errors of order $\delta \| A \| \approx \delta |r_{11}|$. Set the rank to be the largest value of k so $|r_{kk}| \geq \delta \| A \|$.

-3 if you declare anything $\leq \delta$ to be zero.

(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.

5 points

More efficient (2 points) and you can subsequently predict b with just r columns of A (3 points)

7. (10 points)

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.

Solution

5 points

No. The converence rate only takes over when you are close enough. Even if you are "close enough" it can go either way.

(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".

Solution

5 points

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

so if

$$\frac{M_2}{\delta} |x_0 - x_*| \leq \rho < 1$$

then $|x_k - x_*| \leq \rho^k$.