CS 322: Final Exam Solution and Course Grading

Spring 2004

Final Exam Scores

90-100	x	
85-89	xxxxxxxxx	
80-84	xxxxxxxxxxx	
75-79	xxxxxxxxxxxx	
70-74	xxxxxxxxxx	
65-69	xxxxxxxxxxxx	Median = 71
60-64	xxxxxxxxxxx	
55-59	xxxxx	
50-54	xxxxx	
45-49	xx	
< 45	xxxxxx	

Total Scores (= .3(A1 + A2 + A3 + A4 + A5 + A6) + .2(Pre1 + Pre2) + .3 Final)

90-100	xx
85-89	xxxxxxxxxxxxx
80-84	xxxxxxxxxxxxxxxxxxxxxx
75-79	xxxxxxxxxxxxxx
70-74	xxxxxxx
65-69	xxxxxxxxxxxxxxx
60-64	xxx
55-59	xx
50-54	xx
< 50	x

Approximate grade distribution $(A, B, C) = (30\%, 45\%, 25\%)$

Problem 1 (10 points)

Suppose x is a positive floating point number in a base-2 system that represents mantissas (excluding the sign) with 50 bits. What can you say about the value of k after the following script is executed? Assume that $1/2^{100}$ can be represented exactly in the given floating point system.

```
m = 100;
k = 0;
y = 1;
z = x + y;
while z > x  & k < m
   y = y/2;
   z = x + y;
   k = k + 1;
end
```

Solution

Notice that y is repeatedly halved. Thus, we don't expect $x + y$ to be different from x if y is sufficiently small. In particular, when y is in the vicinity of the spacing of floating point spacing at x, then we expect the floating point sum of x and y to equal x.

Floating point numbers are of the form $m \times 2^e$ and to say that 2^{-100} is a floating point number is to say that the system can handle negative exponets up to -100. This was part of the problem because we didn't want exponent underflow to be part of the problem.

Six points for identifying the floating point spacing as a factor and telling what it is. Suppose $x = m \times 2^e$ where $1/2$ □ $m < 1$. The next biggest floating point number is $x + 2^{-50} \times 2^e$. Thus, 2^{e-50} is the spacing.

Four points for talking about the value of k when the loop terminates. For $x + y = x + 1/2^k$ to be bigger than x, we must have

$$1/2^k \geq 2^{-50} \times 2^e \Rightarrow 1 \geq 2^{(k+e-50)} \Rightarrow 0 \geq k + e - 50$$

Thus, the loop will terminate as soon as $k > 50 - e$ or as soon as $k = 100$, whichever comes first.

Problem 2 (15 points)

(a) (5 points) Suppose we are given two points (x_1, y_1) and (x_2, y_2) with the property that $x_1 < x_2$. Under what conditions can we find unique scalars a and b so that if

$$p(x) = a\cos(x) + b\sin(x)$$

then $p(x_1) = y_1$ and $p(x_2) = y_2$?

Solution

Since a and b solve the linear system

$$\begin{bmatrix} \cos(x_1) & \sin(x_1) \\ \\ \cos(x_2) & \sin(x_2) \end{bmatrix} \begin{bmatrix} a \\ \\ b \end{bmatrix} = \begin{bmatrix} y_1 \\ \\ y_2 \end{bmatrix}$$

we must have a nonsingular matrix of coefficients, i.e., $0 \neq \cos(x_1)\sin(x_2) - \cos(x_2)\sin(x_1) = \cos(x_1 - x_2)$. This means that $x_1 - x_2$ cannot be multiple of π.

Two points for displaying the linear system and three points for saying that the matrix must be nonsingular.

(b) (10 points) Complete the following MATLAB function so that it performs as specified.

```
  function pVal = HornerN(c,x,z)
% c is column n-vector, x is a column (n-1)-vector,
% and z is a column m-vector.
%
% pVal is a vector the same size as z with the property that if
%
%     p(t) = c(1) +
%              c(2)*(t-x(1)) +
%              c(3)*(t-x(1))*(t-x(2)) + ... +
%              c(n)*(t-x(1))*(t-x(2))*...*(t-x(n-1))
%
% then pVal(i) = p(z(i)) , i=1:m.
```

Your solution should be vectorized and flop-efficient.

Solution

The idea is to implement the nested multiplication idea, fully illustrated by the $n = 4$ case:

$$p(t) = (((c_4(t - x_3) + c_3)(t - x_2) + c_2)(t - x_1) + c_1$$

```
% Two points for initialization. No points off if just pVal = c(n)
% or forgeting to establish m and n.
n = length(c); m = length(z)
pVal = c(n)*ones(m,1)

% Two points for loop range
for k=n-1:-1:1
   % Six points to the nested multiplication update
   pVal = pVal.*(z - x(k)) + c(k)
end
```

Correct but not vectorized: -2

Correct but not nested: -3

Problem 3 (15 points)

The simple Simpson's rule and its error is given by

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

(a) (10 points) Assume that a function G that is defined everywhere has been implemented and that the M-file `G.m` is available. Complete the following MATLAB function so that it performs as specified. Assume that if `x` is a column vector and `y = G(x)`, then `y` is a column vector with $y_i = G(x_i)$, $i = 1\text{:length}(x)$.

```
  function I = SimpsonForG(a,b,N)
% I is an estimate of the integral of G from a to b based on the composite
% Simpson rule with N equal subintervals.
```

Solution

Take a look at the $N = 3$ case:

$$I \approx \frac{h}{6}\left((y_1+4y_2+y_3) \;+\; (y_3+4y_4+y_5) \;+\; (y_5+4y_6+y_7)\right) \;=\; \frac{h}{6}(y_1+4y_2+2y_3+4y_4+2y_5+4y_6+y_7)$$

where $h = (b-a)/N$ and $y_i = f(x_i)$ with $x_i = a + (i-1)h/2$.

```
% 4 points for a single correct call to G
  m = 2*N+1;
  x = linspace(a,b,m)
  y = G(x);

% 4 points for getting the correct weights, i.e., the 1-4-2-4-2-4-2-4-1 vector:
  w = ones(m,1);
  w(2:2:m-1) = 4;
  w(3:2:m-2) = 2;

% 2 points for final assembly:
  I = ((b-a)/(6*N))*(w'*y);
```

(b) (5 points) If we know that $|G^{(4)(x)}|$ is never bigger than a given constant M, how would you choose N when invoking `SimpsonForG` so that the absolute error is no bigger than 10^{-6} ?

Solution

One point for noting that

$$\text{Error in single subinterval} <= \left(\frac{(b-a)}{N}\right)^5 \frac{M}{2880}$$

Two points for the overall constraint:

$$\text{Total Error} <= N\left(\frac{(b-a)}{N}\right)^5 \frac{M}{2880} \;<=\; 10^{-6}$$

Two points for specifying N_{opt}:

$$N_{opt} \;=\; \text{ceil}\left\{\left(\frac{(b-a)^5 10^6 M}{2880}\right)^{1/4}\right\}$$

Problem 4 (15 points)

Suppose we are given data $(x_1, y_1), \ldots, (x_n, y_n)$ with $x_1 < \cdots < x_n$.

(a) (5 points) What properties are possessed by a cubic spline interpolant of this data?

Solution

- S is a piecewise cubic, i.e., on $[x_i, x_{i+1}]$, $S = q_i(x)$ where q_i is a cubic polynomial
- $S(x_i) = y_i$, $i = 1{:}n$.
- S is continuous
- S' is continuous
- S'' are continuous

(b) (10 points) Suppose $n = 4$ and $y_i = 5x_i^3 - 3x_i$, $i = 1{:}4$. What is $S''((x_1 + x_4)/2)$ given that S is the not-a-knot cubic spline interpolant?

Solution

There are three local cubics, q_1, q_2, and q_3. Since x_2 is not a knot, $q_1 = q_2$. Since x_2 is not a knot, $q_2 = q_3$.

It follows that $S = q_1$, i.e., S is a *single* cubic polynomial.

Since S interpolates the cubic $5x^3 - 3x$ at four points, $S(x) = 5x^3 - 3x$. (The cubic interpolant of four points is unique.)

Since $S''(x) = 30x$, it follows that $S''((x_1 + x_4)/2) = 15(x_1 + x_4)$

Problem 5 (20 points)

(a)(10 points) Assume that

$$\begin{bmatrix} a_1 & e_1 & 0 \\ e_1 & a_2 & e_2 \\ 0 & e_2 & a_3 \end{bmatrix}$$

is positive definite. By comparing matrix entries in the equation

$$\begin{bmatrix} a_1 & e_1 & 0 \\ e_1 & a_2 & e_2 \\ 0 & e_2 & a_3 \end{bmatrix} = \begin{bmatrix} g_1 & 0 & 0 \\ f_1 & g_2 & 0 \\ 0 & f_2 & g_3 \end{bmatrix} \begin{bmatrix} g_1 & 0 & 0 \\ f_1 & g_2 & 0 \\ 0 & f_2 & g_3 \end{bmatrix}^T$$

develop an algorithm that determines g_1, g_2, g_3, f_1, and f_2.

Solution

Two points for each line:

$$\begin{array}{lcl} a_1 = g_1^2 & \Rightarrow & q_1 = \sqrt{a_1} \\ e_1 = f_1 g_1 & \Rightarrow & f_1 = e_1/g_1 \\ a_2 = f_1^2 + g_2^2 & \Rightarrow & g_2 = \sqrt{a_2 - f_1^2} \\ e_2 = f_2 g_2 & \Rightarrow & f_2 = e_2/g_2 \\ a_3 = f_2^2 + g_3^2 & \Rightarrow & g_3 = \sqrt{a_3 - f_2^2} \end{array}$$

(b) (10 points) Assume that A is n-by-n, tridiagonal, symmetric, and positive definite and that we have a matrix G such that $A = GG^T$. If B is a given n-by-n matrix, how would you solve the equation $A^{-1}XA = B$ for X assuming that the n-by-n matrix B is given? Briefly discuss the error in the computed X. Does your solution procedure require $O(n)$, $O(n^2)$, or $O(n^3)$ flops? Why?

Solution

- Equivalent to solving for X in $XA = C$ where $C = AB$. Because A is tridiagonal, C requires $O(n^2)$ flops.
- Since $XA = C$ we have $(XA)^T = C^T$, i.e., $AX^T = C^T$.
- Thus, must solve $AX(i,:)^T = C(i,:)^T$, $i = 1{:}n$. Solve $Gz = C(i,:)^T$ for z and $G^T y = z$. Set $X(i,:) = y^T$.

Two points for C. Two points for trying to use back-solving with G and G^T instead of forming inverse. Two points for correctly back-solving. Two points for $O(n^2$ since solving a linear system with G or G^T is $O(n)$. Two points for mentioning the condition of the matrix A regarding the error.

Problem 6 (15 points)

Suppose we are given data $(t_1, y_1), \ldots, (t_n, y_n)$ that satisfies $0 = t_1 < t_2 < \cdots < t_n$ and $y_1 > y_2 > \cdots > y_n > 0$. Assume also that

$$\frac{y_i - y_{i-1}}{t_i - t_{i-1}} < \frac{y_{i+1} - y_i}{t_{i+1} - t_i} \qquad i = 2{:}n-1.$$

Our goal is to determine α and λ so that

$$\phi(\alpha, \lambda) = \sum_{i=1}^{n} \left(\alpha e^{\lambda t_i} - y_i\right)^2$$

is minimized

(a) (10 points) Assume that λ is fixed. Using the \ operator which can be used to solve least squares problems, show how to determine a scalar a_λ that minimizes $\phi(\alpha, \lambda)$.

Solution

$$\phi(\alpha, \lambda) = \| A\alpha - y \|_2^2$$

where

$$A = \left[\begin{array}{c} e^{\lambda t_1} \\ \vdots \\ e^{\lambda t_n} \end{array}\right] \qquad y = \left[\begin{array}{c} y_1 \\ \vdots \\ y_n \end{array}\right]$$

Thus, $\alpha = A\backslash y$.

(b) (5 points) With α_λ defined by part **(a)** we want to use `fmin` to minimize $\tilde{\phi}(\lambda) = \phi(a_\lambda, \lambda)$. What would be a good search interval $[L, R]$ to pass to `fmin`? Briefly explain your reasoning.

Solution

A graph of the data shows that it is monotone decreasing with slopes always negative but increasing, like e^{-t}. The time constant λ must clearly be negative. Thus, $R = 0$ is appropriate. Three points for getting this far.

Then two points for any reasoned approach to a heuristic for L. For example, if $n = 2$, then we can exactly interpolate by manipulating of $\alpha e^{t_1 \lambda} = y_1$ and $\alpha e^{t_2 \lambda} = y_2$:

$$\lambda = \frac{\log(y_1) - \log(y_2)}{t_1 - t_2}$$

So one possibility would be to set L to be the minimum of the divided differences $(\log(y_i) - \log(y_{i+1})/(t_i - t_{i+1})$, $i = 1{:}n-1$

Problem 7 (10 points)

(a) (5 points) Complete the following MATLAB function so that it performs as specified

```
   function [c,s] = CS(x,y)
 % x and y are scalars. c and s satisfy c^2 + s^2 = 1 and c*x + s*y = 0
```

Solution

Since

$$cx + sy = \begin{bmatrix} c \\ s \end{bmatrix}^T \begin{bmatrix} x \\ y \end{bmatrix}$$

we see that we need a unit vector that is orthogonal to $\begin{bmatrix} x \\ y \end{bmatrix}$. The vector $\begin{bmatrix} -y \\ x \end{bmatrix}$ is orthogonal. Thus $\begin{bmatrix} c \\ s \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} / \sqrt{x^2 + y^2}$

```
  d = sqrt(x^2 + y^2);
  if d ==0
     c = 1; s = 0;      % One point. In this case, c and s can be anything
  else
     c = -y/r; s = x/r  % Four points
  end
```

(b) (5 points) Complete the following MATLAB function so that it performs as specified. You may assume the availability of the function CS from part (a).

```
   function [c,s] = Symmetrize(A)
 % A is a real 2-by-2 matrix. c and s are real scalars such that c^2 + s^2 = 1
 % and [c s ; -s c]*A is symmetric.
```

Solution

$$\begin{bmatrix} c & s \\ -s & c \end{bmatrix} \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} = \begin{bmatrix} ca_{11} + sa_{21} & ca_{12} + sa_{22} \\ -sa_{11} + ca_{21} & -sa_{12} + ca_{22} \end{bmatrix}$$

Two points for realizing that Symmetric means $ca_{12} + sa_{22} = -sa_{11} + ca_{21}$, i.e., $c(a_{12} - a_{21}) + s(a_{22} + a_{11}) = 0$. And three points for this:

```
 [c,s] = CS(A(1,2)-A(2,1), A(1,1)+A(2,2))
```