CS 322: Prelim

Monday, July 15, 2002

Print Name______________________________

Signature______________________________

Student ID Number______________________

The rules for this prelim are as follows:

- **Write your name and student ID number on the front page of the exam booklet. Initial each of the remaining pages in the upper-right hand corner. Sign the front of the exam booklet. Your exam will not be graded if you have not signed the front page of the booklet.**
- The exam has 5 questions plus 1 extra-credit question. The exam is 7 pages long (including this page). Be sure you have all of the pages before beginning the exam.
- This exam will last for 75 minutes.
- Show **ALL** work for partial/full credit. This includes any definitions, mathematics, figures, etc.
- The exam is closed book and closed notes, and no calculators are allowed on the exam.
- No collaboration of any kind is allowed on the exam.

1. ________ (20 pts)

2. ________ (20 pts)

3. ________ (20 pts)

4. ________ (15 pts)

5. ________ (25 pts)

________ (100 pts)

1. (20 points) Complete the following Matlab function so that it performs as described.
Efficiency matters.

```
function [y,z] = SolveMe(A,C,g,h)
%
% Input:
% A and C are n-by-n matrices with A nonsingular
% g and h are column vectors of length n
%
% Output:
% y and z are column vectors of length n satisfying:
%       A'*y+C*z = g
%            A*z = h
```

2. (20 points) Recall the following theorem regarding piecewise cubic Hermite interpolation:

   If q is the cubic Hermite interpolant of f at x_L and x_R, then

   $$|f(z) - q(z)| \leq \frac{M_4}{384} h^4,$$

   where $h = x_R - x_L$, assuming that $f(z)$ and its first four derivatives are continuous and its fourth derivative is bounded above by M_4 on $[x_L, x_R]$.

   Make use of this theorem to complete the following Matlab function:

```
function [R,fR] = FindEndpoint(L,fL,tol)
%
% Input:
% L,fL are scalars that satisfy fL = exp(-L)
% tol is a positive real number
%
% Output:
% R,fR are scalars that satisfy fR = exp(-R) with the
%     property that if  q(z) is the cubic hermite interpolant
%     of exp(-z) at z=L and z=R, then |q(z) - exp(-z)| <= tol on [L,R]
```

3. (20 points)

(a) (12 points) Our study of spline interpolation was motivated by the desire to construct smooth interpolants using piecewise polynomials. In particular, we wanted to find differentiable piecewise polynomials to serve as local interpolants. In class, we studied cubic spline interpolation. However, the simplest type of differentiable piecewise polynomial function on an entire interval $[x_1, x_n]$ happens to be a function obtained by fitting a quadratic polynomial between each pair of adjacent nodes. As it turns out, quadratic splines are not often used in practice. Why is this? In order to answer this question, you will need to think about the equations that we used to specify the various types of cubic splines. Hint: Let f be defined on $[a, b]$ and let the nodes $a = x_1 < x_2 < x_3 = b$ be given and think about deriving the equations necessary to create an interpolant using quadratic splines.

(b) (3 points) When are Bézier curves used (in the context of interpolation)? (We are interested in the mathematical context rather than a specific engineering application.)

(c) (5 points) Outline the general procedure for computing and using Bézier curves. (Note: We're just interested in the general procedure, not the finer details.)

4. (15 points) If 127 is the nearest floating point number to 128 on a base-2 computer, how long is the mantissa?

5. (25 points) Let n be a positive even integer. Define the $n \times n$ matrix A as follows:

$$A_{ij} = \begin{cases} n - |i-j| & \text{if } i \le n/2 \text{ and } j \le n/2 \\ n - |i-j| & \text{if } i > n/2 \text{ and } j > n/2 \\ 1 & \text{if } i \le n/2, j > n/2, \text{ and } j = i + n/2 \\ 1 & \text{if } i > n/2, j \le n/2, \text{ and } i = j + n/2 \\ 0 & \text{otherwise.} \end{cases}$$

Describe an efficient procedure for setting-up A. You do NOT have to write Matlab code to answer this question, but you should be very specific in what you would do to create A. Break your procedure into very small steps. Use the case with $n = 6$ to illustrate your procedure at each step.

EXTRA-CREDIT (5 points):

Suppose that v is a column vector of length n. Show how to efficiently compute Av, where A is the $n \times n$ matrix given in the preceding problem. You do NOT need to write Matlab code to answer this question, but it should be clear from your answer what happens at each step in your procedure. Do this for a general positive even integer n (i.e., not just the case with $n = 6$).