University of Waterloo
Department of Computer Science

CS370 Midterm Examination: Fall 2005

Tuesday, November 1, 2005 Instructors: Profs K. O. Geddes and Y. Li
Duration = 2 hours

Name .... Student ID ....

Section: 8:30 11:30 1:30

The aids allowed are:

- Printed Course Notes
- Lecture notes
- Your assignments
- One text book
- Hand Calculators

There are 8 questions - do all 8.

Question	Mark	Max	Init.
1		6	
2		8	
3		6	
4		6	
5		5	
6		4	
7		5	
8		8	
Total		50	

1. (6 marks)

Consider the function

$$f(x) = \frac{5}{1 + x^2}$$

(a) Write the interpolating polynomial $p_1(x)$ for the function $f(x)$ in the Lagrange form for data $x_1 = 0$, $x_2 = 1$ and $x_3 = 2$.

(b) Let $p_2(x) = c_0 + c_1 x + c_2 x^2$ be the polynomial in simple monomial form that interpolates $f(x)$ at the same data points x_1, x_2, x_3 as in (a). Compute the coefficients c_0, c_1, and c_2 of $p_2(x)$.

(c) What is the relation between $p_1(x)$ and $p_2(x)$? Justify your answer.

2. (8 marks)

(a) Let $S(x)$ be the following piecewise cubic function

$$S(x) = \begin{cases} S_1(x) & \text{for } 0 \leq x < 1 \\ S_2(x) & \text{for } 1 \leq x \leq 3 \end{cases},$$

for some polynomials $S_1(x)$ and $S_2(x)$ of degree not exceeding 3. Do not assume any representation of $S_1(x)$ and $S_2(x)$ for this part of the question.

(i) Write down the condition on $\frac{dS_1(1)}{dx}$ and $\frac{dS_2(1)}{dx}$ that would have to be satisfied for $S(x)$ to be a cubic spline on interval $0 \leq x \leq 3$.

(ii) What condition on the second derivatives of S_1 and S_2 would have to be satisfied for $S(x)$ to be a cubic spline on interval $0 \leq x \leq 3$?

Continued ...

(b) We now specify the following representations for $S_1(x)$ and $S_2(x)$:

$$S(x) = \begin{cases} S_1(x) = 0.917x^3 - 5.917x + a_1 & \text{for } 0 \le x < 1 \\ S_2(x) = b_3x^3 + b_2x^2 + b_1x + 5.375 & \text{for } 1 \le x \le 3 \end{cases},$$

where a_1, b_1, b_2, and b_3 are unknown coefficients. We now also require that $\frac{d^2S(x)}{dx^2} = 0$ at the end points of the interval, i.e. for $x = 0$ and $x = 3$.

Write down four equations that could be used to solve for the four unknowns a_1, b_1, b_2 and b_3. For each equation, indicate what requirement for $S(x)$, or its derivatives, the equation is based on. You should express the equations as linear equations in a_1, b_1, b_2 and b_3, but you do not need to solve them.

3. (6 marks)

Suppose you are fitting a straight line

$$y = ax + b$$

to the three data points $(0, 1)$, $(1, 2)$, and $(3, 3)$.

(a) Set up the 3×2 overdetermined linear system for the least squares problem.

(b) Write down the coefficient matrix and right hand side vector of the corresponding normal equations.

(c) Compute numerical values a and b of the least squares solution.

4. (6 marks)

Consider the following data (t_k, x_k), $k = 1, \ldots, 5$, given in the following table:

t_k	0	1	2	3	4
x_k	x_1	x_2	x_3	x_4	x_5

Let $S(t)$ be the interpolating cubic spline for data (t_k, x_k) $k = 1, \ldots, 5$. Use $S(t)$ to construct two larger data tables by adding some interpolated points. The first larger table has data $(t_k^{(1)}, x_k^{(1)})$ for $k = 1, \ldots, 9$, where $t_k^{(1)} = (k-1)/2$:

$t_k^{(1)}$	0	.5	1	1.5	2	2.5	3	3.5	4
$x_k^{(1)}$	$x_1^{(1)}$	$x_2^{(1)}$	$x_3^{(1)}$	$x_4^{(1)}$	$x_5^{(1)}$	$x_6^{(1)}$	$x_7^{(1)}$	$x_8^{(1)}$	$x_9^{(1)}$

The second larger table has data $(t_k^{(2)}, x_k^{(2)})$ with about 10 times as many columns as (t_k, x_k), where $t_k^{(2)} = (k-1)/10$, $k = 1, \ldots, 41$.

Note: help command descriptions for some Matlab spline commands are provided in the appendix.

(a) Write the Matlab commands to compute the representation of $S(t)$. You may assume that two vectors named `t` and `x` have been initialized with the data (t_k, x_k), $k = 1, \ldots, 5$, from the first table above. Your commands should assign the representation to a Matlab variable `Srep`.

(b) Show how `Srep` as described in part (a) can be used to compute the table values $x^{(1)}$ and $x^{(2)}$ as specified above.

5. (5 marks)

Consider the system FPNS of floating point numbers of the form

$$\pm\ 0.d_1d_2...d_t \times \beta^p \quad \text{for} \quad L \le p \le U\ ,\ d_1 \ge 1\ ,\ d_i \ge 0 \ \text{ for } \ 1 < i \le t$$
$$\text{or} \quad 0 \quad \text{(a special floating point number)}$$

with parameters $t = 5$, $\beta = 10$, $L = -5$, $U = 5$.

(a) What is the smallest positive value that does not result in underflow?

(b) What number is $fl(2578.9523)$ in the floating point number system defined above?

(c) What number is $5912.32 \oplus 4087.50$? ($\oplus$ is addition in the FPNS.)

(d) How many numbers are there in the interval $[1, 2)$ in the FPNS defined above?

6. (4 marks)
Matlab uses the IEEE standard double precision number system $\mathcal{F}(\beta = 2, t = 53, L = -1022, U = 1023)$ for its floating point arithmetic. Its machine epsilon value is $\texttt{eps} = 2^{-52}$.

Let $a = 16 - 2^{-23}$; $b = 32 + 2^{-22}$. Both these numbers are in the FPNS ; note that $16 = 2^4$.

(a) Compute $a \otimes b \ = fl(a \times b)$

(b) Compute the relative error

$$\text{RelativeError} = \frac{|a \otimes b - a\, b|}{|a\, b|}$$

and compare it to machine eps by computing the ratio RelativeError/eps.

7. (5 marks)

(a) Describe each step of Gaussian elimination with partial pivoting for the coefficient matrix A of the linear system $Ax = b$ where :

$$A = \begin{pmatrix} 0 & 0.5 & -1 \\ 1 & 0 & 0 \\ 0 & 2 & 0.1 \end{pmatrix}.$$

(b) Give the corresponding permutation matrix P, the lower triangular matrix with unit diagonal L and the upper triangular matrix U such that $PA = LU$ for matrix A in part (a).

8. (8 marks)
Let

$$p_1(x) = c_1\, x^2 \; + \; c_2\, x \; + \; c_3$$

be the monomial form of the interpolating polynomial for 3 data pairs $(1,1), (a,-1), (0,0)$ for any value of a. The vector of coefficients, C_1, satisfies the Vandermode system of linear equations, $V\, C_1 = D_1$, where:

$$V = \begin{pmatrix} 1 & 1 & 1 \\ a^2 & a & 1 \\ 0 & 0 & 1 \end{pmatrix}, \quad D_1 = \begin{pmatrix} 1 \\ -1 \\ 0 \end{pmatrix}$$

(a) Determine the entries $u_{2,2}, u_{2,3}, u_{3,3}$ in the upper triangular factor U of $V = LU$ where

$$L = \begin{pmatrix} 1 & 0 & 0 \\ a^2 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad U = \begin{pmatrix} 1 & 1 & 1 \\ 0 & u_{2,2} & u_{2,3} \\ 0 & 0 & u_{3,3} \end{pmatrix}$$

(b) Suppose that we want to compute two additional quadratic polynomials $p_2(x)$ and $p_3(x)$ to interpolate $(1,2)$, $(a,-1/2)$, $(0,0)$ and $(1,4)$, $(a,-1/4)$, $(0,0)$ respectively.

Write down the Vandermode systems satisfied by the column vector C_2 and C_3 which are the coefficients of $p_2(x)$ and $p_3(x)$ respectively.

Continued ...

(c) Assuming that the Matlab matrices L and U satisfying $V = LU$ have been initialized for you according to part (a), write a Matlab program that computes matrix $C = [C_1, C_2, C_3]$ by re-using the factors L and U.

Appendix

```
help spline

 SPLINE Cubic spline data interpolation.
    YY = SPLINE(X,Y,XX) uses cubic spline interpolation to find YY, the values
    of the underlying function Y at the points in the vector XX.  The vector X
    specifies the points at which the data Y is given.  If Y is a matrix, then
    the data is taken to be vector-valued and interpolation is performed for
    each column of Y and YY will be length(XX)-by-size(Y,2).

    PP = SPLINE(X,Y) returns the piecewise polynomial form of the cubic spline
    interpolant for later use with PPVAL and the spline utility UNMKPP.

    Ordinarily, the not-a-knot end conditions are used. However, if Y contains
    two more values than X has entries, then the first and last value in Y are
    used as the endslopes for the cubic spline.  Namely:
       f(X) = Y(:,2:end-1),   df(min(X)) = Y(:,1),   df(max(X)) = Y(:,end)

    Example:
    This generates a sine curve, then samples the spline over a finer mesh:
        x = 0:10;  y = sin(x);
        xx = 0:.25:10;
        yy = spline(x,y,xx);
        plot(x,y,'o',xx,yy)

    Example:
    This illustrates the use of clamped or complete spline interpolation where
    end slopes are prescribed. Zero slopes at the ends of an interpolant to the
    values of a certain distribution are enforced:
       x = -4:4; y = [0 .15 1.12 2.36 2.36 1.46 .49 .06 0];
       cs = spline(x,[0 y 0]);
       xx = linspace(-4,4,101);
       plot(x,y,'o',xx,ppval(cs,xx),'-');

    See also INTERP1, PPVAL, SPLINES (The Spline Toolbox).


help ppval

 PPVAL  Evaluate piecewise polynomial.
    V = PPVAL(PP,XX) returns the value at the points XX of the piecewise
    polynomial contained in PP, as constructed by SPLINE or the spline utility
    MKPP.
```

```
V = PPVAL(XX,PP) is also acceptable, and of use in conjunction with
FMINBND, FZERO, QUAD, and other function functions.

Example:
Compare the results of integrating the function cos and this spline:
  a = 0; b = 10;
  int1 = quad(@cos,a,b,[],[]);
  x = a : b; y = cos(x); pp = spline(x,y);
  int2 = quad(@ppval,a,b,[],[],pp);

int1 provides the integral of the cosine function over the interval [a,b]
while int2 provides the integral over the same interval of the piecewise
polynomial pp which approximates the cosine function by interpolating the
computed x,y values.

See also SPLINE, MKPP, UNMKPP.
```

```
help csape

 CSAPE Cubic spline interpolation with various end-conditions.

    PP  = CSAPE(X,Y)

    returns the cubic spline interpolant (in ppform) to the given
    data (X,Y) using Lagrange end-conditions (see default in table below).

    PP  = CSAPE(X,Y,CONDS) uses the end-conditions specified in CONDS, with
    default values (which depend on the particular conditions).

    PP  = CSAPE(X,Y,CONDS,VALCONDS) uses the end-conditions specified
    in CONDS, with particular values as specified in VALCONDS.

    CONDS may be a *string* whose first character matches one of the
    following: 'complete' or 'clamped', 'not-a-knot', 'periodic',
    'second', 'variational', with the following meanings:

    'complete'     : match endslopes (as given in VALCONDS, with
                     default as under *default*)
    'not-a-knot'   : make spline C^3 across first and last interior
                     break (ignoring VALCONDS if given)
    'periodic'     : match first and second derivatives at first data
                     point with those at last data point
                     (ignoring VALCONDS if given)
    'second'       : match end second derivatives (as given in VALCONDS,
```

```
                with default [0 0], i.e., as in variational)
'variational' : set end second derivatives equal to zero
                (ignoring VALCONDS if given)
The *default* : match endslopes to the slope of the cubic that
                matches the first four data at the respective end.
```

By giving CONDS as a 1-by-2 matrix instead, it is possible to specify *different* conditions at the two endpoints, namely CONDS(i) with value VALCONDS(:,i), with i=1 (i=2) referring to the left (right) endpoint.

CONDS(i)=j means that the j-th derivative is being specified to be VALCONDS(:,i) , j=1,2. CONDS(1)=0=CONDS(2) means periodic end conditions.

If CONDS(i) is not specified or is different from 0, 1 or 2, then the default value for CONDS(i) is 1 and the default value of VALCONDS(:,i) is taken. If VALCONDS is not specified, then the default value for VALCONDS(:,i) is taken to be

```
 deriv. of cubic interpolant to nearest four points, if   CONDS(i)=1;
                 0                                   if   CONDS(i)=2.
```

It is possible (and, in the case of gridded data required) to specify VALCONDS as part of Y. Specifically, if size(Y) == [d,ny] and ny == length(X)+2, then VALCONDS is taken to be Y(:,[1 end]), and Y(:,i+1) is matched at X(i), i=1:length(X).

It is also possible to handle gridded data, by having X be a cell array containing m univariate meshes and, correspondingly, having Y be an m-dimensional array (or an m+1-dimensional array if the function is to be vector-valued). Correspondingly, CONDS is a cell array with m entries, but the information normally specified by VALCONDS is now expected to be part of Y.

For example,

```
   fnplt(csape( [0:4], [1 0 -1 0 1;0 1 0 -1 0], 'periodic')), axis equal
```

plots a circle, while

```
   x = linspace(0,2*pi,21);  pp = csape( x, sin(x), [1 2], [1 0] );
```

gives a good approximation to the sine function on the interval [0 .. 2*pi] (matching its slope 1 at the left endpoint, x(1) = 0, and its second derivative 0 at the right endpoint, x(21) = 2*pi, in addition to its value at every x(i), i=1:21).

As a multivariate vector-valued example, here is a sphere, done as a parametric bicubic spline, using prescribed slopes in one direction and periodic side conditions in the other:

```
   x = 0:4; y=-2:2; s2 = 1/sqrt(2);
   clear v
   v(3,:,:) = [0 1 s2 0 -s2 -1 0].'*[1 1 1 1 1];
   v(2,:,:) = [1 0 s2 1 s2 0 -1].'*[0 1 0 -1 0];
   v(1,:,:) = [1 0 s2 1 s2 0 -1].'*[1 0 -1 0 1];
   sph = csape({x,y},v,{'clamped','periodic'});
   values = fnval(sph,{0:.1:4,-2:.1:2});
 surf(squeeze(values(1,:,:)),squeeze(values(2,:,:)),squeeze(values(3,:,:)))
   % the previous two lines could have been replaced by: fnplt(sph)
   axis equal, axis off
```

See also CSAPI, SPAPI, SPLINE.

Scrap paper

Scrap paper