Computer Science CSC260S — Spring 2000
St. George Campus — University of Toronto

Homework Assignment #2
Due: Monday, March 13, 2000
Partners: not allowed for this assignment
Mark: 10%

1. (15 points) This question is to be done by hand calculations.

Write down all conditions on a_i, b_i and c_i, $i = 1, 2$, which guarantee that the following function $f(x)$ is continuous in the knots $0, 1, 2$. Describe what each condition means.

$$f(x) = \begin{cases} a_1 x^2 + b_1 x + c_1 & \text{for } 0 \le x \le 1 \\ a_2 x^2 + b_2 x + c_2 & \text{for } 1 \le x \le 2 \\ 0 & \text{elsewhere} \end{cases}$$

With the additional conditions $f(1) = 1$, $f'(0) = -1$, and $f'(2) = 1$ determine a_i, b_i and c_i, $i = 1, 2$. Show your calculations and the numerical values for a_i, b_i and c_i, $i = 1, 2$.

2. (20 points) Write a MAPLE procedure called `quadspline` that finds a piecewise quadratic spline based on data $(x_0, y_0), (x_1, y_1), \ldots, (x_n, y_n)$, with $x_0 < x_1 < \cdots < x_n$. The boundaries of the n pieces are at the x_i locations. Since we have 3 unknowns per piece (n pieces in total), we have $3n$ unknowns. The values of each piece at its two endpoints should match the given y values. This gives us $2n$ equations. Furthermore, the first derivative of the two pieces that end at location x_i, $i = 1 \ldots, n-1$, should match. This gives us $n-1$ other equations. Obtain a final equation by enforcing that the second derivative at x_n be zero.

The lists of x values and y values should be given as the first and second arguments of `quadspline`. The third argument should be the unknown variable for the quadratic polynomials. The result returned should be expressed using the `piecewise` function.

To help you get started, a MAPLE procedure that finds first-order splines is given in the accompanying handout. You should follow the general pattern of this procedure — in particular, for this exercise, you should not work out solutions to equations by hand, but should instead get MAPLE to solve them, as in the example provided.

3. (45 points)

(a) (30 points) MAPLE provides a function `spline` that computes cubic spline interpolation among other spline interpolations (see the manual page in MAPLE). In this exercise, you will write a similar function `clampspline` that computes cubic spline interpolation with the clamped (derivative) boundary conditions. [Read about the clamped cubic spline interpolation in the accompanying handout.]

The lists of x values and y values should be given as the first and second arguments of `quadspline`. The third argument should be a list of length 2 containing the derivative data at the endpoints. The fourth argument is the unknown variable for the clamped cubic spline polynomials. The result returned should be expressed using the `piecewise` function.

Implement the above function in MAPLE. Your code should follow these steps.

- Calculate the step-sizes from `x`.
- Form the cubic spline interpolation matrix for clamped boundary conditions (see Equation (6) of the notes) in **sparse** format.
- Form the right side vector of the cubic spline interpolation system for clamped boundary conditions (see Equation (6) of the notes).
- Solve the linear system to obtain the coefficients c_j, $j = 0, \ldots, n$.
- Recover the rest of the coefficients of the cubic spline (see Equations (3), (2) and (4) of the notes).

(b) (15 points) Consider three sets of data given by the following piece of MAPLE code:

```
x1  = [1,2,5,6,7,8,10,13,17];                    % Curve 1
y1  = [3.0,3.7,3.9,4.2,5.7,6.6,7.1,6.7,4.5];
dy1 = [1.0,-0.67];
x2  = [17,20,23,24,25,27,27.7];                  % Curve 2
y2  = [4.5,7.0,6.1,5.6,5.8,5.2,4.1];
dy2 = [2.0,-4.0];
x3  = [27.7,28,29,30];                           % Curve 3
y3  = [4.1,4.3,4.1,3.0];
dy3 = [0.33,-1.5];
```

Each set of data corresponds to a curve. For each curve the x and y values on several points are given, as well as the y' values on the two endpoints of the curve.

Consider also the set of equidistant evaluation points $\{t_i, i = 1, \ldots, 290\}$ in the interval $[1, 30]$, with $t_1 = 1$ and $t_{290} = 30$. Write a MAPLE program that

- constructs the clamped cubic spline interpolant of each of the above sets of data, evaluates it on the evaluation points and plots the resulting (three) vectors versus the (appropriate) evaluation points in one graph (Figure 1);
- constructs the cubic spline interpolant of each of the above sets of data, evaluates it on the evaluation points and plots the resulting (three) vectors versus the (appropriate) evaluation points in one graph (Figure 2);
- constructs the linear spline interpolant of each of the above sets of data, evaluates it on the evaluation points and plots the resulting (three) vectors versus the (appropriate) evaluation points in one graph (Figure 3);

- constructs the (appropriate degree) polynomial interpolant of each of the above sets of data, evaluates it on the evaluation points and plots the resulting (three) vectors versus the (appropriate) evaluation points in one graph (Figure 4);

Comment on what you observe. Notes:

- Use your own `clampspline` function (developed in Part (a)) to get the values of the clamped cubic spline interpolant on the evaluation points.
- Use MAPLE's `spline` and `interp` to get the values of the cubic and linear spline interpolant, and of the polynomial interpolant on the evaluation points.
- For each set of data above use only that values of t that are within the x range.
- You can find an electronic version of the above curves in the webpage of the course (hence, no excuses for typographical errors...).
- Hand in a hard-copy of your code and plots, as well as your comments.

4. (20 points) Consider the function $g(x) = \ln(x)$ in the interval $I = \left[\frac{1}{16}, \frac{17}{16}\right]$. Write a MAPLE script that, for $n = 4, 8, 16, 32, 64$,

- constructs a set of equidistant points x_i, $i = 0, \ldots, n$, with $x_0 = \frac{1}{16}$ and $x_n = \frac{17}{16}$, the respective set of function values $y_i = g(x_i)$, $i = 0, \ldots, n$ and the 2 derivative values $y_0^{'} = g^{'}(x_0)$ and $y_n^{'} = g^{'}(x_n)$;

- constructs the clamped cubic spline interpolant of the data (x_i, y_i), $i = 0, \ldots, n$, $y_0^{'}$ and $y_n^{'}$, and evaluates it on a set of 100 equidistant points in I (including the endpoints) and computes the maximum in absolute value error of the interpolant on the set of 100 points;

- constructs the cubic spline interpolant of the data (x_i, y_i), $i = 0, \ldots, n$, evaluates it on a set of 100 equidistant points in I (including the endpoints) and computes the maximum in absolute value error of the interpolant on the set of 100 points;

- constructs the linear spline interpolant of the data (x_i, y_i), $i = 0, \ldots, n$, evaluates it on a set of 100 equidistant points in I (including the endpoints) and computes the maximum in absolute value error of the interpolant on the set of 100 points;

- constructs the polynomial interpolant of the data (x_i, y_i), $i = 0, \ldots, n$, evaluates it on a set of 100 equidistant points in I (including the endpoints) and computes the maximum in absolute value error of the interpolant on the set of 100 points;

- saves the (maximum in absolute value) errors computed above in 4 vectors, say `errcs, errs, errl, errp`, each of length 5.

After completing the computations for each value of n, **output** the 4 vectors `errcs`, `errs`, `errl`, `errp`. Comment on the results obtained.

Notes:

- To get the values of the 4 interpolants on the set of 100 equidistant points in I follow the notes of question 3.
- Do **not** output more results than requested. Indicate which output corresponds to which interpolant. Hand in a hard-copy of your code and output, as well as your comments.