

# MATH 335: Numerical Analysis

## Problem Set 12, Final version

Due Date: Mon., Mar. 30, 2009

Read Section 6.3 in Kharab & Guenther.

★20 Suppose  $x_1, \dots, x_n$ , and  $c_{10}, c_{11}, c_{20}, c_{21}, \dots, c_{n0}, c_{n1}$  are given real numbers.

(a) Consider the functions

$$A_i(x) := [1-2(x-x_i)L'_i(x_i)][L_i(x)]^2 \quad \text{and} \quad B_i(x) := (x-x_i)[L_i(x)]^2, \quad i = 1, 2, \dots, n,$$

where the  $L_i(x)$ ,  $i = 1, \dots, n$  are the usual cardinal functions (of Lagrange interpolation). Show that these functions satisfy the relationships

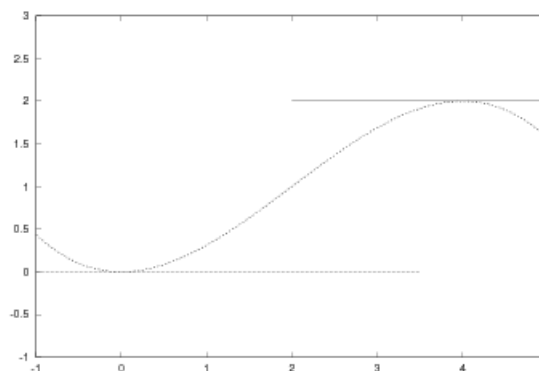
$$\left. \begin{array}{l} A_i(x_j) = \delta_{ij} \\ A'_i(x_j) = 0 \\ B_i(x_j) = 0 \\ B'_i(x_j) = \delta_{ij} \end{array} \right\} \text{ for } 1 \leq i, j \leq n, \text{ where } \delta_{ij} := \begin{cases} 0, & \text{if } i \neq j, \\ 1, & \text{if } i = j. \end{cases}$$

(b) Explain how one may use the functions  $A_i$ ,  $B_i$  defined above to construct a polynomial  $p$  of degree at most  $(2n - 1)$  which solves the Hermite interpolation problem

$$p(x_i) = c_{i0}, \quad p'(x_i) = c_{i1}, \quad \text{for } i = 1, 2, \dots, n.$$

(c) Write a function (if in OCTAVE, call it `hermite.m`) which implements part (b). In particular, your routine should accept arguments  $\mathbf{x}$  (the nodes),  $\mathbf{y}$  (the vector of function values at the nodes),  $\mathbf{yp}$  (the vector of derivative values at the nodes), and  $\mathbf{xi}$ , a vector of points at which to evaluate your resulting interpolating polynomial. You may wish to make calls to other routines in the process—for instance, once you have found appropriate coefficients for each  $L_i$ , you may want a routine that operates on those coefficients in such a way as to produce the coefficients of the corresponding derivative  $L'_i$ . I believe any subsidiary routines can be included write in the same *.m*-file with `hermite()`; but should you not find this to be the case, make sure you send me all of the necessary files.

- (d) A “switching path” between two parallel railroad tracks is to be a cubic polynomial joining positions  $(0, 0)$  and  $(4, 2)$ , tangent to the lines  $y = 0$  and  $y = 2$ . (See the figure.) Find (analytically) the coefficients of the appropriate polynomial. Then use your Hermite interpolation program to produce its graph, and check that it is working properly.



- ★21 In our discussion of cubic splines, we assumed that each cubic “piece”  $S_i(x)$  could be written in the form

$$S_i(x) = d_i(x - x_i)^3 + c_i(x - x_i)^2 + b_i(x - x_i) + a_i .$$

- (a) Let  $p(x) = x^3 + x - 1$ . Take  $x_0 = 2$ , and rewrite  $p$  (accurately) in the form

$$p(x) = d(x - x_0)^3 + c(x - x_0)^2 + b(x - x_0) + a .$$

- (b) Explain how this may be done for a general polynomial  $p$  and general “offset”  $x_0 \in \mathbb{R}$ . (Hint: Consider Taylor’s theorem.)

- ★22 The book-supplied cubic spline program `sp13.m` could be improved a great deal. Write programs that carry out the following modifications. (Read them all before you begin. You may prefer to write your own algorithm from scratch—one that performs according to the final set of specifications—based on information from class, rather than make modifications to `sp13.m`.)

- (a) Instead of printing values of the cubic spline function to the screen, values should be returned to the calling program, as suggested by this modified function declaration

```
function [abscissas, ordinates] = sp13(x, y, m)
```

- (b) Unlike most OCTAVE/MATLAB interpolating routines, `sp13.m` does not allow a user to specify the points at which to evaluate the natural cubic spline  $S(x)$ . The `m` argument it accepts has to be an integer, and is used to subdivide each interval  $[x_i, x_{i+1}]$  so that there are `m` equally-spaced  $x$ -values inserted inside the interval; these are the  $x$ -values at which  $S$  is evaluated.

Make changes to your routine so that the new function declaration

```
function vals = myspl3(x, y, xi)
```

is used. Here  $\mathbf{xi}$  is a vector of  $x$ -values (specified by the user) at which the natural cubic spline  $S(x)$  should be evaluated. The corresponding values of  $S(x)$  are returned in the vector  $\mathbf{vals}$ .

- (c) The natural cubic spline is the one which employs the constraints  $S''(x_1) = S''(x_n) = 0$ . Modify your routine yet again so its declaration looks like

```
function vals = myspl3(x, y, xi, z1=0, zn=0)
```

Note that when this routine is called with three arguments, the values of  $z_1$  and  $z_n$  will default to 0. The routine should take whatever values are passed for  $z_1$  and  $z_n$  and use them in determining the corresponding cubic spline  $S(x)$  (*natural* only if  $z_1 = 0$  and  $z_n = 0$ ) that satisfies  $S''(x_1) = z_1$  and  $S''(x_n) = z_n$ , which should then be evaluated at the user-specified values found in  $\mathbf{xi}$ .

- (d) Write a program with declaration

```
function vals = clampedSpl3(x, y, xi, yprime1, yprimen)
```

that determines the appropriate clamped cubic spline (i.e., one whose additional constraints are  $S'(x_1) = \mathbf{yprime1}$  and  $S'(x_n) = \mathbf{yprimen}$ ) and then evaluates it at the user-specified values found in  $\mathbf{xi}$ .

What is likely going to be new (different from part (c)) in your routine is the inclusion of commands translating the given  $\mathbf{yprime1}$ ,  $\mathbf{yprimen}$  values into two more equations (equations to replace the two equations involving  $z_1$  and  $z_n$  in part (c)); after that you can (depending on how you coded part (c)) find the interpolating cubic spline as before. Work out these two replacement equations on paper first.

As a way to check your work, a call to your function should produce the same result as the following call to the OCTAVE `spline()` function:

```
> spline(x, [yprime1 y yprimen], xi)
```

That is (according to the help for `spline()`), when the 2nd argument  $\mathbf{y}$  has 2 more components than the first one  $\mathbf{x}$ , “then the first and last values of the vector  $\mathbf{y}$  are the first derivative of the cubic spline at the end-points.