Math 6316, Spring 2016
Due 26 February 2016

Homework 3 – Interpolation Part II

1. Let $f \in C^4([a,b])$, and consider a partition $\mathcal{T}_h$ of $[a,b]$ into K subintervals, $I_j = [x_j, x_{j+1}]$, where $a = x_0 < x_1 < \ldots < x_K = b$, $h_j = x_{j+1} - x_j$ and $h = \max_{j=0,\ldots,K-1} h_j$. Suppose that you have the data values $\{(x_j, f(x_j))\}_{j=0}^K$ and $\{(x_j, f'(x_j))\}_{j=0}^K$, and on each subinterval I_j you construct the cubic Hermite interpolant $H_{3,j}$ such that

$$H_{3,j}(x_j) = f(x_j), \quad H'_{3,j}(x_j) = f'(x_j), \quad H_{3,j}(x_{j+1}) = f(x_{j+1}), \quad H'_{3,j}(x_{j+1}) = f'(x_{j+1}).$$

Let $H_h^3 \in C^1([a,b])$ be the piecewise polynomial function constructed using these parts, i.e.

$$H_h^3(x)\Big|_{x \in I_j} = H_{3,j}(x).$$

Prove the error bound

$$\|f - H_h^3\|_\infty \le C h^4 \|f^{(4)}\|_\infty,$$

where the norms are taken over all $x \in [a,b]$.

2. Let $f \in C^2([a,b])$ and assume that $f(a) = f(b)$ and $f'(a) = f'(b)$. Let s_3 be the periodic cubic spline interpolant of f at the knots $a = x_0 < x_1 < \ldots < x_n = b$. Prove that

$$\int_a^b \left[s_3''(x)\right]^2 \,\mathrm{d}x \le \int_a^b \left[f''(x)\right]^2 \,\mathrm{d}x.$$

3. Determine the equations to enforce the periodic cubic spline boundary conditions,

$$s_3'(a) = s_3'(b), \qquad s_3''(a) = s_3''(b)$$

in terms of the second-derivative coefficient values $M_j = s_3''(x_j)$, $j = 0, \ldots, n$. Use these, along with the standard equations for the cubic spline coefficients, to write the full linear system of equations that you would solve to find the coefficients $\{M_j\}_{j=0}^n$ of the periodic cubic spline interpolant for the data $\{(x_j, f(x_j))\}_{j=0}^n$.

4. Matlab: Using your result from problem 3, construct two functions

```
function M = cubic_spline_coefficients(x, f)
function s = cubic_spline_evaluate(x, f, M, z)
```

The first of these takes as input two arrays `x` and `f`, containing the data values $\{x_j\}_{j=1}^{n+1}$ and $\{f(x_j)\}_{j=1}^{n+1}$, and computes/returns the coefficients $\{M_j\}_{j=1}^{n+1}$ for the periodic cubic spline interpolant of the data. In this function, be certain to check that `x` and `f` have the same size, and that the data for the function values is indeed periodic (i.e. $f_1 = f_{n+1}$). You may assume that the nodes are sorted (i.e. $a = x_0 < x_1 < \ldots < x_n = b$).

The second function should take as inputs the vectors of knots, values and spline coefficients, $x, f, M \in \mathbb{R}^{n+1}$, and a point $z \in \mathbb{R}$; it should evaluate and return the value of $s_3(x)$.

Using the above two functions, write a third function with the form

```
function v = parametric_spline(x, m)
```

Here, `x` $\in \mathbb{R}^{2\times(n+1)}$, where each column corresponds to a (x, y) location in $\mathbb{R}^2$, and `m` is an integer at least 1. The output `v` $\in \mathbb{R}^{2\times m}$ contains coordinates for a uniformly-spaced curve that interpolates the data in `x`, i.e. `v` has `m` columns, each an (x, y) location in $\mathbb{R}^2$.

To approach this, think `x` as consisting of two distinct sets of data:

$$\{(t_k, x_k)\}_{k=0}^n \quad \text{and} \quad \{(t_k, y_k)\}_{k=0}^n,$$

where the t_k "knots" are simply the cumulative Cartesian distance between the successive (x, y) locations, i.e. if we define the line segment lengths $l_0 = 0$ and

$$l_j = \sqrt{(x_j - x_{j-1})^2 + (y_j - y_{j-1})^2}, \quad j = 1, \ldots, n,$$

we can then set the cumulative length of all line segments up to the node (x_k, y_k) using the formula

$$t_k = \sum_{j=1}^{k} l_j, \quad k = 0, \ldots, n.$$

Once you have computed these $\{t_k\}_{k=0}^n$ values, you then compute two separate sets of cubic spline coefficients, one for the data $\{(t_k, x_k)\}_{k=0}^n$, and the second for the data $\{(t_k, y_k)\}_{k=0}^n$. Finally, you should create a set of linearly-space data "times",

```
z = linspace(0,t(n),m);
```

you can then evaluate your separate spline interpolants at each z_k value to determine your output coordinates, $\{(x(z_k), y(z_k))\}_{k=1}^m$.

Write a script named `hw3.m` that creates three plots. In each plot, select at least 10 points in the $\mathbb{R}^2$ plane to use for your input data `x` to `parametric_spline`. Ensure that your last point and first point in each test is identical (so that the data is periodic). For each plot:

- Plot the data points using large, solid, red circles, e.g.

  ```
  plot(x(1,:),x(2,:),'ro','MarkerFaceColor','r','MarkerSize',10)
  ```

- Overlaid on these points, plot the interpolated periodic parametric spline curve using a solid blue line; this line should have at least 2000 interpolation points.

When grading, I plan to run both your script and one of my own, so ensure that your `parametric_spline` function takes the correct input/output arguments, in the same order, as specified above.