AMSC/CMSC460 Computational Methods **Fall 2014**

Homework 5-6, Due on Tuesday, October 21, 2014

1. *(Mixed-type polynomial interpolation)* Let f be a function with the following point values:

x_i	1	3	4
$f(x_i)$	-3	2	4
$f'(x_i)$	1		2
$f''(x_i)$			-1

a). Find a polynomial P_5 of degree 5 that interpolates f as well as its derivatives at corresponding nodes in the table. (You do not have to simplify your result)

b). Prove that there exists $\xi = \xi(x)$ in $[1,4]$ such that.

$$f(x) - P_5(x) = \frac{f^{(6)}(\xi)}{6!}(x-1)^2(x-3)(x-4)^3.$$

Hint: use the same idea in Theorem 6.2 and Theorem 6.4 in Suli's book.

2. *(Linear splines)* Let $f(x) = x^3$ in the interval $x \in [0,1]$. $\{x_k\}_{k=0}^m$ are $m+1$ equally distributed nodes (or knots):

$$x_k = \frac{k}{m}, \quad k = 0, \cdots, m.$$

a). $s_L(x)$ is the linear interpolating spline for f. We write $s_L(x) = \sum_{k=0}^m a_k \varphi_k(x)$, where φ_k are basis functions (in this case, hat functions). What are the coefficient a_k? Find $a_0, \cdots, a_5$ for the case $m = 5$.

b). $s(x)$ is the linear spline which minimize $\|f - s\|_2$. We write $s(x) = \sum_{k=0}^m \alpha_k \varphi_k(x)$. How to get the coefficient α_k? To answer this question, please follow exercise 11.4 in Suli's book. Calculate A_{ij} and b_i explicitly with our specific $f(x)$. Solve the linear system $A\alpha = \mathbf{b}$ for $m = 5$ (you can use Matlab backslash operation, or get help from the next problem), and find $\alpha_0, \cdots, \alpha_5$.

3. *(Tridiagonal matrix algorithm)* The goal is to solve a linear system $Ax = b$, where A is a tridiagonal matrix. While traditional Gaussian elimination method requires $\mathcal{O}(n^3)$ operations, and $\mathcal{O}(n^2)$ space storage, the *Thomas algorithm* only requires $\mathcal{O}(n)$ operations, and the matrix can be stored sparsely which requires $\mathcal{O}(n)$ space storage.

a). Read the Wikipedia page for Thomas algorithm and understand how it works.
`http://en.wikipedia.org/wiki/Tridiagonal_matrix_algorithm`

b). Write a Matlab function `x = thomas(A, b)` to implement Thomas algorithm. Note that your input matrix could be sparsely defined. (You might get some help from section 2.10 in Moler's book.)

c). Download the Matlab code `testthomas.m` to the same folder where your function `thomas.m` is located. Run the script to test whether your code works or not.

4. *(Interpolating cubic spline)* The interpolating cubic spline is a $\mathcal{C}^2$ function which is a piecewise cubic polynomial, and it interpolates given function f at given set of knots $\{x_i\}_{i=0}^m$. Since there are 2 extra degrees of freedom, different boundary conditions can be imposed.

a). Write $s(x)$ in the form

$$s(x) = \frac{(x_i - x)^3}{6h_i}\sigma_{i-1} + \frac{(x - x_{i-1})^3}{6h_i}\sigma_i + \alpha_i(x - x_{i-1}) + \beta_i(x_i - x),$$

for $x \in [x_{i-1}, x_i]$, and $h_i := x_i - x_{i-1}$. Check that $s''(x_i-) = s''(x_i+) = \sigma_i$, and express α_i, β_i in terms of $\{\sigma_i\}_i$ and $\{f_i\}_i$.

b). Express $s'(x)$ in terms of $\{\sigma_i\}_i$. What is the condition $s'(x_i-) = s'(x_i+)$ in $\{\sigma_i\}_i$? You should obtain a linear system of $\{\sigma_i\}_i$ with $m+1$ unknowns and $m-1$ equations.

c). One natural way to impose boundary condition is $s''(x_0) = s''(x_m) = 0$. Write the linear system $A\sigma = b$ that consists $m-1$ interior equations and the two boundary conditions. Here, $\sigma = (\sigma_0, \cdots, \sigma_m)^T$ is a column vector with $m+1$ entries. Please specify matrix A and vector b. Is the matrix A sparse or not?

d). Consider another set of boundary condition $s'(x_0) = s'(x_m) = 0$. In this case, what does the linear system look like? Write the matrix A and vector b.

e). There is another so-called periodic boundary condition, where $s^{(k)}(x_0) = s^{(k)}(x_m)$ for $k = 0, 1, 2$. In this case, x_0 and x_m are considered as a same point. Now, we denote $\sigma = (\sigma_1, \cdots, \sigma_m)^T$ be a column vector with m entries. (Note here we omit σ_0 as it is the same as σ_m.) Write out the system $A\sigma = b$.

f). Read section 3.5 on numerical implementations for cubic spline in Moler's book.
(The following is *optional*. No need to submit) Write a Matlab code to find interpolating cubic spline, under different setups on boundary conditions.