Computer Assignment 3

Solutions should be submitted on Blackboard by 11pm on Wednesday, 15 November

1. [15 marks]

a) Write a Matlab function `newton_coef.m` which computes, by divided differences, coefficients of Newton's interpolating polynomial for the points (x_i, y_i), Your m-file should have a header line

```
function [ck, ddif] = newton_coef(xi, yi)
```

where `xi` is the vector of n + 1 nodes, `yi` is the vector of n + 1 values, `ck` is the vector of Newton coefficients, and `ddif` is the divided differences table, such that `ddif(j+1,k+1)` = $f[x_j, \ldots, x_{j+k}]$, $j, k = 0, \ldots, n$. (see pp. 67,68 in Pav's lecture notes). Test your algorithm on Example Problem 5.6 on p. 68 of Pav's notes. You should obtain

```
>> xi = [1; .5; 3]; yi = [3; -10; 2];
>> format rat
>> [ck,ddif] = newton_coef(xi,yi)
ck =
       3
      26
     -53/5
ddif =
       3             26            -53/5
     -10             24/5            0
       2              0              0
```

b) Write a Matlab function `eval_newton.m` which evaluates Newton's interpolating polynomial *in the nested form*. Your m-file should have a header line

```
function y = eval_newton(x, ck, xi)
```

where `x` is a vector of values for which the polynomial needs to be evaluated, `ck` is the vector of Newton coefficients, and `xi` is the vector of nodes.

c) Using your Matlab functions from part (a) and (b), or otherwise, find the interpolating polynomial for the Runge function $f(x) = (1 + x^2)^{-1}$ and equally spaced nodes $x_i = -5, -4, \ldots, 3, 4, 5$. Plot the Runge function and the interpolating polynomial $p_{10}(x)$ in the interval [–5, 5]. Indicate the location of the interpolation nodes with red circles. Calculate the polynomial approximation error $\max_{x\in[-5,5]} |f(x) - p_{10}(x)|$.
Hint: you should evaluate the Runge function and the polynomial at many points in [–5, 5], so that the functions look smooth on the plot (like in the Figures on p. 70 of Pav's notes).

d) Repeat part (c), but using Chebyshev nodes instead of equally spaced nodes. The interpolation should be on the same interval with the polynomial of the same degree.

2. [15 marks] A cubic spline $S(t)$ interpolating data points (t_i, y_i), $i = 0, \ldots, n$, is given by the formula

$$S(t) = S_i(t) = y_i + \left[\frac{1}{h_i}(y_{i+1} - y_i) - \frac{h_i}{6}(y''_{i+1} + 2y''_i)\right](t - t_i) + \frac{1}{2}y''_i(t - t_i)^2 + \frac{1}{6h_i}(y''_{i+1} - y''_i)(t - t_i)^3$$

for $t \in [t_i, t_{i+1}]$, where $h_i = t_{i+1} - t_i$ and the 2nd derivatives of the cubic spline at the nodes, y''_i, $i = 0, \ldots, n$, satisfy the system of linear equations

$$a_i y''_{i-1} + b_i y''_i + c_i y''_{i+1} = d_i, \ \ i = 1, \ldots, n-1,$$

where

$$a_i = \frac{t_i - t_{i-1}}{6}, \quad b_i = \frac{t_{i+1} - t_{i-1}}{3}, \quad c_i = \frac{t_{i+1} - t_i}{6}, \quad d_i = \frac{y_{i+1} - y_i}{t_{i+1} - t_i} - \frac{y_i - y_{i-1}}{t_i - t_{i-1}}.$$

There are n – 1 equations for n + 1 unknowns, so two additional conditions need to be specified in order for the solution to be unique. A cubic spline is called *natural* if the two conditions are $y''_0 = y''_n = 0$.

Note: For the details of the derivation of this formula see lecture notes `Cubic Spline (version 2).pdf` in Course Documents.

a) Write a Matlab function which computes the 2nd derivatives at the nodes for the natural cubic spline. The function inputs are vectors $(t_0, t_1, ..., t_n)^T$, and $(y_0, y_1, ..., y_n)^T$, and the function output is the vector $(y_0'', y_1'', \ldots, y_n'')^T$. Since the system of linear equations is tri-diagonal, it can be efficiently converted into the upper triangular form by naïve Gaussian Elimination using only one `for` loop (rather than the three nested loops of the standard GE). The back-substitution algorithm is also much simpler and involves only one loop as well (instead of the standard two nested loops). You should implement such efficient GE and back-substitution algorithms in your function, instead of the standard ones.

b) Write a Matlab function which calculates the value of the natural cubic spline at a point $t \in [t_0, t_n]$ given $(t_0, t_1, ..., t_n)^T$, $(y_0, y_1, ..., y_n)^T$, and $(y_0'', y_1'', \ldots, y_n'')^T$ as inputs. For a given t, the function should first determine i such that $t \in [t_i, t_{i+1}]$ and then output the value of $S_i(t)$.

c) Using your functions in parts (a) and (b), or otherwise, construct the natural cubic spline interpolation of the Runge function as in Question 1(c). Produce the plot and calculate the error $\max_{x\in[-5,5]} |f(x) - S(x)|$.