CS412 Spring Semester 2011

Homework Assignment #2

Due Thursday February 24th 2011, in class

Sum of all problems : 120%, Maximum possible score : 100%.

1. [40%] Create a `MATLAB` function called **NewtonCubic** which:

   (a) Accepts the 5 parameters a_0, a_1, a_2, a_3 and x_k, in that specific order.

   (b) Returns the approximation at the $(k+1)$-th step (i.e x_{k+1}) that would be generated by Newton's method for the nonlinear (cubic) equation

   $$a_0 + a_1 x + a_2 x^2 + a_3 x^3 = 0$$

   when the approximation at the k-th step is the value of the parameter x_k.

   Turn in a printout of the function you created. Use the function you constructed to approximate the solution to the equation $x^3 + x + 1 = 0$ (the real solution is $x \approx -0.6823$). Use $x_0 = 1$ as your initial guess, and apply 5 steps of the Newton procedure you generated (i.e. you will need to call the function **NewtonCubic** 5 times). List the approximations that your function created in those 5 iterations.

2. [20%] Adapt the function you created in the previous problem into a `MATLAB` function called **NewtonPoly** which:

   (a) Accepts the *vector* $\mathbf{P}$ and the real number x_k as parameters.

   (b) Returns the approximation at the $(k+1)$-th step (i.e x_{k+1}) that would be generated by Newton's method for the polynomial equation

   $$P(1) + P(2)x + P(3)x^2 + \cdots + P(N)x^{N-1} = 0$$

   where N is the size of $\mathbf{P}$ and the approximation at the k-th step is the value of the parameter x_k.

   Test your function on the equation $x^5 - x^3 + 1 = 0$ with an initial guess $x_0 = -1$. Provide your code, and the first 5 approximations generated by iterated application of your function.

3. [60%] Consider the following 4 points:

   $$\begin{aligned} (x_0, y_0) &= (-1, 0.5) \\ (x_1, y_1) &= (0, 1) \\ (x_2, y_2) &= (1, 2) \\ (x_3, y_3) &= (2, 4) \end{aligned}$$

(a) [20%] Find a polynomial of degree 3 that interpolates through (x_0, y_0), $\ldots, (x_3, y_3)$ using the Lagrange interpolation method.

(b) [20%] Find a polynomial of degree 3 that interpolates through (x_0, y_0), $\ldots, (x_3, y_3)$ using the Newton interpolation method. Your result should be the same as above.

(c) [20%] The 4 points given above were taken from the "real" function $f(x) = 2^x$. Using `MATLAB` draw $f(x)$ on the same plot as the polynomial interpolant you created, for $x \in [-5, 4]$ (please turn in a printout of the plot). What do you observe?