MA/CS 321:001
MWF 11:00–11:50
FB 213
Fall 2004

Instructor: Russell Brown
Office: POT741
Phone: 257-3951
russell.brown@uky.edu

Announcements.

- As announced in class on Monday, there will be a curve on the first exam.
- Extra credit will continue to be awarded for locating erors in homework assignments.
- Homework solutions should be written in complete sentences. Please be sure that you answer the question.
- We first discuss polynomial interpolation. We will not do much programming with this topic. I hope that we can have some fun with integration.

Homework 5, Due Wednesday, 14 October 2004.

1. (programming exercise, 5 points) Write a short function which implements the Horner method for evaluating polynomials. See page 3 of the text. You should test your function by comparing the result with matlab function `polyval`.

   The function `mypolyval(a,x)` should take as input a vector `a` $= (a_1, a_2, \ldots a_n)$, and return the value of the polynomial

   $$a_1 x^{n-1} + a_2 x^{n-2} + \ldots a_{n-1} x + a_n.$$

   - Hand in a printout of your m-file and an example illustrating the use of this m-file. For example, compute the expression $1 + 2 + 4 + 8 + 16 + 32$ which can be though of as a polynomial whose coefficients are all 1 and is evaluated at $x = 2$. Compare the output of your function with `polyval`.
   - Your m-file should include a comment on the second line of the file describes its function. If you type `help mypolyval` at a matlab prompt, this information will be printed.
   - You may want to use the statement `[rows, cols] = size(a);` to determine the size of the vector `a`. Thus, you do not need to pass the size of `a`.

     You may create row vectors by writing `a = [1,2,3]`. You may access the jth element by writing `a(j)` or `a(1,j)`. If you prefer to use column vectors, the assignment statement `b=a.'` assigns to `b` the transpose of `a`.
   - (optional, 2 points extra credit) Use the `.*` operator so that your function will accept vectors as the second argument. Test this by creating a plot of a polynomial.
   - (comment) Note that matlab indexes arrays from 1 rather than 0. This will lead to some small changes in your code compared to the examples in the text.

- (comment) This method requires only n multiplications and additions to evaluate a polynomial of degree $n-1$.

2. (5 points) Find the Lagrange polynomial which satisfies $p(1) = 1$ and $p(j) = 0$ for $j = 2, 5, 6$. You do not need to simplify the polynomial.

3. (5 points) Can you find a polynomial for which $p(1) = 1$, $p'(1) = 2$ and $p''(1) = 3$?

4. (5 points) Section 4.1, page 162, #4.

5. (5 points) Section 4.1, page 162, #5.

6. (5 points) Section 4.1, page 162, #6.

7. (5 points) Section 4.1, page 163, #10. Do not use divided differences. Instead, I would like you to folllow the procedure in example 3, page 144.

8. (5 points) Section 4.1, page 165, #24.

October 4, 2004