Assignment #6

Due Thursday, 19 October at the start of class

Please read sections 2.4 and 4.1–4.3 of the textbook (J. Epperson, *An Intro. to Numerical Methods and Analysis*, 2nd edition).

Section 2.4, pages 66–68:

- **Exercise 1.** Do parts (a), (b), and (c) only.
- **Exercise 7.** This can be answered with formula (2.13).

Section 4.3, pages 190–191:

- **Exercise 14.** Do parts (a) and (b) only.
- **Exercise 1.** Do this two ways. First use formula (4.12) to get an upper bound on the error. Then actually compute the interpolant using `polyfit`, and sample using `polyval`, to get an accurate value for $\|f - p_2\|_\infty$.
- **Exercise 4.**

P7. On page 179 of the textbook there are pseudocodes Algorithm 4.1 and 4.2. The first Algorithm computes coefficients $a_0, \dots, a_n$ for the Newton-form interpolating polynomial $p_n(x) = a_0 + a_1(x - x_0) + \cdots + a_n(x - x_0) \cdots (x - x_{n-1})$ for given data $(x_i, f(x_i))$. The second Algorithm is Horner's rule for evaluating $p_n(x)$. This pair of algorithms is just like the MATLAB pair `polyfit` and `polyval`, but now via the coefficients of the Newton form.

Implement Algorithms 4.1 and 4.2 as a pair of functions

```
function a = npolyfit(x,y,n)
```

and

```
function px = npolyval(a,x,xx)
```

respectively. You will need to be quite careful with the indexing arrays! (*Because* MATLAB *starts at index 1 and the pseudocodes start at 0.*) Make sure that `npolyval()` works with any *array* of points `xx`. Also, note that you do not need to pay any attention to the "divided differences" interpretation of Newton's form, as described on pages 180–183.

Of course, show your codes in your solutions. Then test them by the following plot comparison which *must produce two identical figures*:

```
f = @(x) sin(5 * x);
xi = [0 1 2 2.5 pi];
p = polyfit(xi,f(xi),4);          % coefficients in standard form
a = npolyfit(xi,f(xi),4);         % coefficients in Newton form
xx = -1:.01:4;
figure(1),  plot(xx,f(xx),xx,polyval(p,xx),xi,f(xi),'o')
figure(2),  plot(xx,f(xx),xx,npolyval(a,xi,xx),xi,f(xi),'o')
```

Show the figures in your solutions.

Explain briefly why your new pair `npolyfit`, `npolyval` is actually more efficient than `polyfit` and `polyval`, for a plotting task like the above.