**Taylor’s Polynomial using MATLAB Tools**

**Topic#1: Use of Matlab GUI for Taylor’s Polynomial: taylortool**

Matlab has a very interesting device that computes and plots a function and its Taylor polynomial. A pop-up window is obtained by typing

**>> taylortool**

in Matlab command window. Try this. You should see the following window:



The window contains a region where the graph of f (in blue) is plotted with the graph of the Taylor polynomial pn(x) (in dashed red). You can

* type a new function inside the box next to **f(x) =,**
* choose the order **N** of the Taylor polynomial (either type it or use the arrows to increase or decrease it),
* choose the point **a** where the Taylor polynomial is centered,
* choose the region along the *x*-axis where the functions are plotted.

**Topic #2: Computing the Taylor’s polynomial for the function and its Error using Matlab**

We will learn the use of Matlab function **taylor ( )** to calculate the Taylor’s polynomial for a given function. Then use Matlab to plot the error between the given function and its Taylor’s polynomial representation. This error is usually plotted as a function of x.

We know how to measure the error between the values of f(x) and pn(x) at a point *x*. This error is

**| f (x)−pn(x)|** .

On the other hand, f(x) and pn(x) *are* defined over an interval [a,b] and for some choices of x, this error may be large while for some other values of *x* this error may be small. What do we do when have to consider a whole range of values for *x*? Take the largest one!

**Definition** The *error* between f and its nth order Taylor’s polynomial over the interval [*a*,*b*] is the maximum value of

**| *f* (*x*)−*pn*(*x*)|** ----- (**Eq#1**)

when *x* belongs to [*a*, *b*].

In practice, the **Eq#1** as given above is a function of *x* and by plotting it we can estimate its largest value. Let’s do this for **cos(*x*)** and its Taylor’s polynomial of order 4. Type the following:

**>> syms x**

**>> f=cos(x);**

**>> t4=taylor(f,'Order',5,'ExpansionPoint',0);**

**>> ezplot( abs(f-t4) ,[-2,2]);**

What do these matlab commands represent?

The first line takes the variable ‘x’ as symbolic, and then generate a symbolic function ‘f’ using symbolic variable ‘x’. (For symbolic representations see the Symbolic Toolbox in Matlab).

The expression **t4** is the 4th order (note for 4th order we use 5) Taylor’s polynomial for f at x=0*.* For the above function t4 should be*:*

**t4 =x^4/24 - x^2/2 + 1**

The **abs** command stands for absolute value. The ezplot command plotted the **Eq#1** over the interval [−2,2].

Unfortunately, we can’t see the largest value of the graph because it blows up near the ends of the interval. Let’s resize our screen with the axis command. Try

**>> axis([-2,2,0,0.1]);**

* You can also change the y-axis limits by directly editing the plot in the Figure window.

The final plot is shown below



Now we can see that the graph of the **Eq#1** is always smaller than 0.09. Therefore we claim that the error between f and t4 over the interval [−2, 2] is less than 0.09*.*