Taylor Series [1]

MATLAB has an interactive Taylor series calculator called `taylortool`. It plots `f` and the `N`-th degree Taylor polynomial on an interval. After `taylortool` is started, we can change `f`, `N`, the interval, or the point `a`.

1. (a) Enter the command: `taylortool('sin(x)')`
   (b) In the taylortool window, change `N` to be 3. You can change the degree `N` using the buttons >> or <<. Also you can just enter the value for `N` in the box for `N`.
   (c) For what domain does the Taylor polynomial appear to be a good approximation of the function?
   (d) Now use the button >> to increase `N` until the approximation appears to be accurate on the whole interval.
   (e) For the degree `N` above, use Taylor's Formula (by hand) to find an upper bound on the error of the approximation.
2. In the `taylortool` window, change the function to $f(x) = e^x$ (use `exp(x)`), the interval to $[-3, 3]$ and N to 3. Repeat the process above.
3. Repeat the above process for $\sin(e^x)$ on the interval $[0, 3]$. What problems do you encounter. What do you think causes this? Does $\sin(e^x)$ equal its Taylor series? For roughly what range of x and N would $T_N(x)$ be a practical approximation tool? What might be a more reasonable strategy for approximating $\sin(e^x)$?
4. Prepare a brief (< 1 page) written report describing what happened and answering the questions. Use complete sentences and standard mathematical notation. Do **not** get a printout.

The taylortool can help us gain some appreciation for the loss of accuracy of the Taylor approximation as x varies farther from the approximation point a. We also encounter the difficulty of approximating a function that oscillates. Although a Taylor Series does actually equal a certain function, computers can only do polynomial operations. So for instance, the sine function on calculators or computers **must** be approximated using polynomial computations and knowing the accuracy is important.