MM553/837 - Computational Physics
Tutorial Exercise - Week 36

- Set up a C++ compiler!
- (Taken from Landau Sec. 1.8.2) Write a program that implements the series evaluation of the sin function discussed in class, where the accuracy is an input parameter and determines when the series sum should be stopped.
- Try code that sums the series in a good way (no factorials) to one that calculates the sum in a bad way (explicit factorials). Start with a tolerance of 10^{-8}.
- Show that for sufficiently small values of x, your algorithm converges (the changes are smaller than your tolerance level) and that it converges to the correct answer.
- Compare the number of decimal places of precision obtained with that expected from the stopping criterion.
- Without using the identity $\sin(x + 2n\pi) = \sin(x)$, show that there is a range of somewhat large values of x for which the algorithm converges, but that it converges to the wrong answer.
- Show that as you keep increasing x , you will reach a regime where the algorithm does not even converge.
- Now make use of the identity $\sin(x + 2n\pi) = \sin(x)$ to compute $\sin x$ for large x values where the series otherwise would diverge.
- Repeat the calculation using the bad version of the algorithm (the one that calculates factorials) and compare the answers.
- Set your tolerance level to a number smaller than machine precision and see how this affects your conclusions.