Math 4610 Fundamentals of Computational Mathematics - Lecture 3.

The accumulation of round off error can be a serious problem in computational mathematics. The idea in this lecture is to focus on errors that occur in every arithmetic operation performed by a computer. Some simple examples of how to mitigate inexact arithmetic interval analysis will be covered. Documentation of algorithms and computer codes is essential to any successful attempt to write reusable code.

Content Items:

- Measuring Errors - Absolute and Relative Errors: The usual way to measure errors is either the absolute value of the difference between an approximation and the exact value. Relative or percent error gives a percent error relative to the exact value. These concepts will be defined and examples will be given. It should be noted that in many algorithms, how error is measured will determine stopping criteria for the algorithms. > > go there (pdf)
- **Types of Error:** There a number of sources of errors encountered in scientific computation. A list of the types of errors will be covered in this part of the lecture. > > go there (pdf)
- **Building a Shared Library for Reusing Code:** An important skill that computational mathematicians should learn is to take relatively small codes and turn these into reusable object files. These can be collected into a shared library. A shared library can be linked to by other programs from this point on. Note that the equivalent of a shared library in a Windows setting is called a DLL. > > go there (pdf)
- **Wrap up and Questions:** If there is time and anyone has questions about the lecture, these will be addressed.