Numerical Analysis or Scientific Computing

Concerned with design and analysis of algorithms for solving mathematical problems that arise in computational science and engineering.

Distinguishing features:

- Deals with quantities that are continuous rather than discrete
- Concerned with *approximations* and their effects

Approximations are not used just by choice: they are inevitable in most problems.

General Strategy

Replace difficult problem by easier one that has same solution, or at least closely related solution.

- complicated → simple
- nonlinear → linear
- infinite → finite
- differential → algebraic

Solution obtained may only approximate that of original problem

Sources of Approximation

Before computation begins:

- modeling
- empirical measurements
- previous computations

During computation:

- truncation or discretization
- rounding

Accuracy of final result may reflect combination of approximations, and perturbations may be amplified by nature of problem or algorithm.

Example: Approximations

Computing surface area of Earth using formula

$$A = 4\pi r^2$$

involves several approximations:

- Earth is modeled as sphere, an idealization of its true shape
- Value for radius is based on empirical measurements and previous computations

- Value for π requires truncating an infinite process
- Values for input data and results of arithmetic are rounded in computer

Data Error and Computational Error

Typical problem: compute value of function $f:\Re \to \Re$ for given argument.

True value of input is x, desired result is $f(x)$.

Inexact input $\hat{x}$ is used instead.

Approximate function computed is $\hat{f}$.

Total error = $\hat{f}(\hat{x}) - f(x) =$
$$(\hat{f}(\hat{x}) - f(\hat{x})) + (f(\hat{x}) - f(x)) =$$
computational error + propagated data error

Choice of algorithm has no effect on propagated error.