Chapter 1. Numerical Algorithms

Scientific computing is a discipline concerned with the development and study of *numerical algorithms* for solving mathematical problems that arise in various science and engineering disciplines. The starting point is a *mathematical model*. We will concentrate on those mathematical models which are *continuous* (or *piecewise continuous*) and are difficult or impossible to solve analytically.

In order to solve such a model approximately on a computer, the continuous or piecewise continuous problem is approximated by a discrete one, i.e. $\Delta S \approx S' \Delta t$. Functions are approximated by finite arrays. Algorithms, like Euler's method, are then sought which approximately solve the mathematical problem efficiently, accurately, and reliably. This is the heart of scientific computing. Numerical analysis is the theory behind these algorithms.

Chapter 1. Numerical Algorithms

The next step after devising algorithms is their implementation: loops, assignment, arrays, if-then, while, ...

1.2 Errors

There are two basic types of error measurements for some quantity v and its approximation w.

- *absolute error:* $|v - w|$.
- *relative error:* $\frac{|v-w|}{|v|}$.
- The relative error is often more meaningful.

1.2 Errors

Here is a list of sources of common errors in numerical algorithms.

- Errors in the problem (model or data or ...) are impossible to fix.
- Approximation Errors: Discretization errors and Convergence errors.
- Roundoff errors.
- We assume that approximation errors mostly dominate roundoff errors.

1.2 Errors

Throughout this text we consider various computational errors depending on a discretization step size $h > 0$ (or $\Delta t > 0$). In other instances, such as when estimating the efficiency of a particular algorithm, we are interested in a bound on the work estimate as a parameter n increases unboundedly.

- $e = \mathcal{O}(h^q)$.
- means there is a positive constant C so that $|e| \leq Ch^q$.
- We say the method is qth order.
- Euler's method is order 1. RK4 is order 4.

1.2 Errors

EXAMPLE: Use the algorithm $\frac{f(x_0+h)-f(x_0)}{h}$ to approximate $f'(x_0)$ when $f(x) = \sin x$ and $x_0 = 1.2$.

1.3 Algorithms Properties

Useful Calculus Results for analyzing numerical methods.

- Intermediate Value Theorem (IVT)
- Mean Value Theorem (MVT)
- Rolle's Theorem
- Taylor's Theorem

1.3 Criteria for Analyzing Algorithms

- Accuracy: Must state what type of error is expected.
- Efficiency: A good algorithm terminates reasonably quickly.
- Robustness: The algorithm must be stable.

1.3 Algorithm Properties

In order to reduce the number of computations, polynomials are evaluated on computers in way that is natural for humans to understand.

EXAMPLE: $p(x) = 3x^5 - 0.5x^4 + 11.1x^3 - 2x^2 + x - 1.$

1.3 Problem Conditioning

The appraisal of a given computed approximation requires understanding *problem sensitivity* and *algorithm stability*

- ill-conditioned (too sensitive): Small perturbations in data lead to large differences in the result.
- well-conditioned (stable): Yields numerical solution which is an exact solution of a slightly perturbed problem.

1.4 Error Accumulation

- $E_n \approx c_0 n E_0$ for some constant c_0 represents linear growth.
- $E_n \approx c_1^n E_0$ for some constant c_1 represents exponential growth. Exponential growth should be avoided.
- Evaluate the integrals $y_n = \int_0^1 \frac{x^n}{x+10} dx$ for $n = 1, 2, \ldots, 30$.