Unit 2. Partial differential equations (PDEs) in physics

Lecture 202: error sources

Reference book:

“Introduction to Computational Astrophysical Hydrodynamics” by Zingale.
http://bender.astro.sunysb.edu/hydro_by_example/CompHydroTutorial.pdf

W. Banda-Barragán, 2022

Sources of error

With any algorithm, there are two sources of error we are concerned with:

- *truncation error*
- *roundoff error*

Partial differential equations, generalities and classification

Truncation errors are a feature of an algorithm—we typically approximate an operator or function by expanding about some small quantity.

When we throw away higher-order terms, we are truncating our expression, and introducing an error in the representation. If the quantity we expand about truly is small, then the error is small.

Example (truncation error):

A simple example is to consider the Taylor series representation of sin(*x*):

$$\sin(x) = \sum_{n=1}^{\infty} (-1)^{n-1} \frac{x^{2n-1}}{(2n-1)!}$$

For $|x| \ll 1$, we can approximate this as:

$$\sin(x) \approx x - \frac{x^3}{6}$$

in this case, our truncation error has the leading term $\propto x^5$, and we say that our approximation is $\mathcal{O}(x^5)$, or 5th-order accurate.

Partial differential equations, generalities and classification

Roundoff errors arise from the error inherent in representing a floating point number with a finite number of bits in the computer memory.

Some reorganisation of algorithms can help minimise roundoff, e.g. avoiding the subtraction of two very large numbers by factoring but roundoff error will always be present at some level.

$$x^3 - y^3 = (x - y)(x^2 + xy + y^2)$$

Norms:

Often we will need to measure the "size" of an error to a discrete approximation. For example, imagine that we know the exact function, $f(x)$ and we have have an approximation, f_i defined at N points, $i = 0,\ldots,N-1$.

The error at each point i is $\varepsilon_i = |f_i - f(x_i)|$. But this is N separate errors—we want a single number that represents the error of our approximation. This is the job of a vector norm.

There are many different norms we can define. For a vector **q**, we write the norm as $\|q\|$. Often, we put a subscript after the norm brackets to indicate which norm was used.

Norms:

Some popular norms are:

- *inf norm*:

$$\|q\|_\infty = \max_i |q_i|$$

- *L1 norm*:

$$\|q\|_1 = \frac{1}{N}\sum_{i=0}^{N-1} |q_i|$$

- *L2 norm*:

$$\|q\|_2 = \left[\frac{1}{N}\sum_{i=0}^{N-1} |q_i|^2\right]^{1/2}$$

- *general p-norm*

$$\|q\|_p = \left[\frac{1}{N}\sum_{i=0}^{N-1} |q_i|^p\right]^{1/p}$$

Norms:

Note that these norms are defined such that they are normalized—if you double the number of elements (*N*), the normalisation gives you a number that can still be meaningfully compared to the smaller set.

For this reason, we will use these norms when we look at the convergence with resolution of our numerical methods.

We'll look into how the choice of norm influences your convergence criterion, but inspection shows that the inf-norm is local—a element in your vector is given the entire weight, whereas the other norms are more global.