Investigation 4: Manufactured Solutions

Christopher Pattison

Error Sources

It is desirable to quantify how much error a solution contains in order to determine how closely the computed solution corresponds to the actual solution to the PDE. This is important in validating that the solver works as intented.

Round-Off Error

Error can arise due to round-off error in the floating point calculations. This is a result of finite memory used to store a (potentially) non-finite number of digits. Beyond a certain number of digits, two numbers will be rounded to the same stored value and therefore are indistinguishable. The direction of rounding is evenly distributed in $\mathbb{R}$ and unpredictable after rounding which makes the resulting error random. The number of significant base-10 digits that can be reasonably expected from a floating point format is equation (1) where N is the number of bits in the significand.

$$d = N \log_{10} 2 \tag{1}$$

Truncation Error

or due to the non-continuous nature of the descritization The latter is a result of the truncation of the Taylor Series used to take a finite difference in the Finite Volume Method.

$$\frac{df}{dx} \approx \frac{f(x+h) - f(x-h)}{2h} + O(h^2) \tag{2}$$

For a second order finite difference (2) where h is the grid spacing, error ϵ decreases with the square of h: A $\frac{1}{2}$ refinement in h will result in a $\frac{1}{4}$ decrease in ϵ.

Manufactured Solutions

The method of manufactured solutions forces the solution to a known equation, which allows error to be quantified.

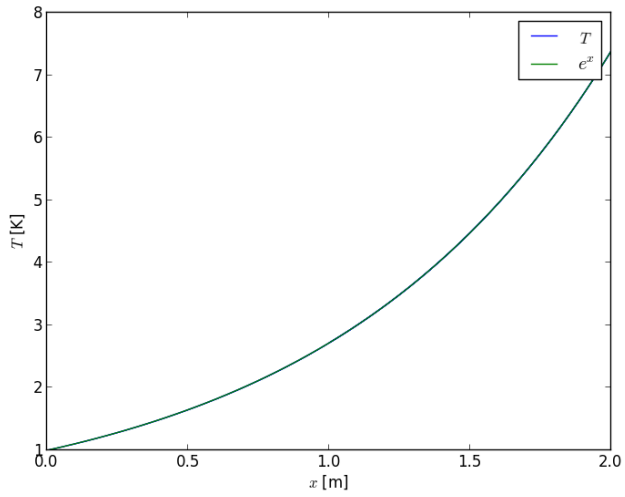

Figure 1: Manufactured Solution

Derivation

$$\frac{dkdT}{dx^2} + q + S = 0 \tag{3}$$

Starting with the heat equation (3), we plug in the desired solution (4) to T. Ideally, a solution should have infinitely many derivatives so that it is not possible for a truncated Taylor Series to precisely represent the solution. Thus, $T = x$ or $T = 1$ would be a poor choice.

$$T = e^x \tag{4}$$

$$\frac{dkde^x}{dx^2} + q + S = 0 \tag{5}$$

The equation (5) is then solved for the source term

$$S = -q - ke^x \tag{6}$$

The source term (6) is then plugged into the solver to force the solution to that value.

Error Quantification

Once a solution is obtained, error can be quantified with equation (7). Since the solver's solution is discrete, ϵ can instead be expressed as equation (8). $\epsilon(h)$

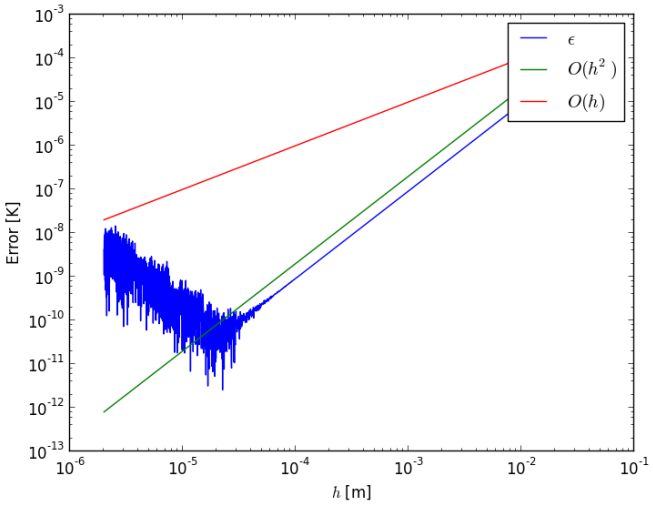

Figure 2: Extended Precision Solver Error

can be graphed and the slope compared against $O(h)$ and $O(h^2)$ convergence.

$$\epsilon = \sqrt{\int_\Omega (\hat{T} - \tilde{T})^2 d\Omega} \tag{7}$$

$$\epsilon \propto \sqrt{\frac{1}{N}\sum_{i=1}^{N} (\hat{T}(x_i) - \tilde{T}_i)^2} \tag{8}$$

Truncation Error

By evaluating ϵ with several grid spacings, relationship (9) can be experimentally determined.

$$\epsilon \propto h^n \tag{9}$$

In fig. 2, it is evident that the error is proportional to h^2. Thus, the solver can be confirmed to be second order. In the region dominated by truncation error, the graph is a straight line. However, the region where round-off error dominates is identifiable by a lack of smoothness as it is random.

Round-off Error

Plugging in various values of the significand width into eq. (1). It can be seen that a larger floating point format can represent many more significant digits. In general, a trade-off must be made between resource usage and solver accuracy. As floating point precision goes up, the decrease in solver speed becomes more noticeable as hardware does not natively support the larger formats: Generation of fig. 2 took 2 minutes whereas the generation of fig. 4 took 2 hours. It is notable that an increase in grid resolution eventually *must* be followed by an increase in floating point precision.

Floating Point Format	Significand Width (bits)	Significant Decimal Digits
Single	24	7
Double	53	16
Extended	64	19
Quad	113	34

Figure 3: Floating Point Decimal Precision

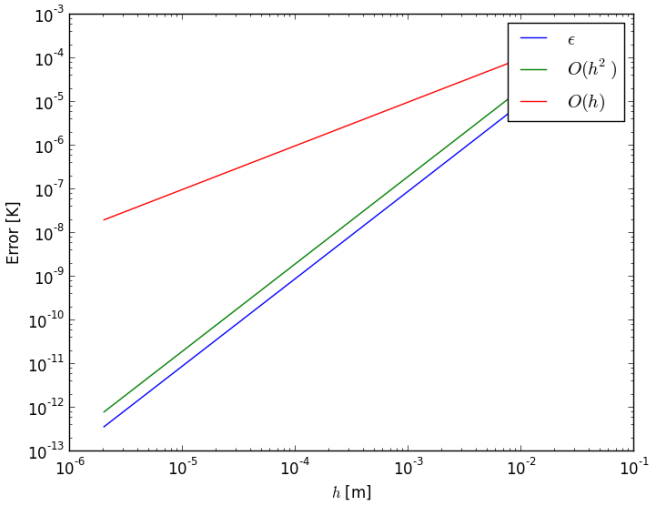

Figure 4: Quad Precision Solver Error