Error in ODE numerics

Ahmet Sacan

Some figures from:
Advanced Engineering Mathematics, Kreyszig
Wikipedia

Error in Approximation

- Truncation error
  - intrinsic to the method used for approximation
  - can be improved by using larger number of smaller intervals or switching to a better approximation method
- Round-off error
  - error due to inexact binary representation of numbers.
- We are usually only worried about truncation error and ignore round-off error.
- Reducing truncation error (by using a very small interval) may increase round-off error.

Error in Approximation

- Local error
  - Error made in each step.
- Global error
  - Results from propagation of local errors
  - Average error across all steps.
  - Error in the final value.

Taylor Series - Remainder Term

$$f(x) = \sum_{i=0..n} \frac{f^{(i)}(a)}{i!}(x-a)^i \quad + \frac{f^{(n+1)}(\boldsymbol{\tau})}{(n+1)!}(x-a)^{n+1}$$

– for some $\boldsymbol{\tau}$: $\mathrm{a} \leq \tau \leq x$

- e.g., for single-Euler's method:

$$f(a+h) = f(a) + hf'(a) + \frac{h^2}{2} f''(\boldsymbol{\tau})$$

Example

- What is an approximate value of sin(1) ?
  - Use Euler's method with $x=0$ as starting point, and $h=1$.
  - i.e., single iteration of Euler's method
  - i.e., first order Taylor Series.
- What is the error bound for the local truncation error?

odeerror_examples.m

Exercise

- Consider the initial value problem:

$$\frac{dy}{dx} = \frac{1}{x} \qquad y(e) = 1$$

- Calculate an approximate value for y(3)
  - use a single-iteration of Euler's method.
- What is the error bound for the local truncation error?

Order of the Local Error

- Let's repeat Taylor Expansion

$$f(a+h) = f(a) + hf'(a) + \frac{h^2}{2} f''(\tau)$$

- The remainder term $\mathrm{R} = \frac{h^2}{2} f''(\tau)$ changes by h^2. We use the O notation to represent the order of the remainder term:

$$f(a+h) = f(a) + hf'(a) + O(h^{\mathbf{2}})$$

- $O(h^{\mathbf{2}})$ represents how fast the error decays as we decrease h.

Order of the Global Error

- In Global Error, we consider the accumulated error as a result of multiple iterations in the Euler's method.
- Over a fixed interval, the number of steps is proportional to 1/h.
- Propagation of the local error $O(h^2)$ over 1/h steps gives us:

$$O\left(h^2 \ * \frac{1}{h}\right) = O(h^{\mathbf{1}})$$

- For this reason, we say Euler method is a **first-order method**.

Automatic Step Size Selection

- Since the Remainder term provides an error bound, we can use it to calculate h for any given tolerance level.
  - TOL: "tolerance", maximum amount of error we are willing to tolerate.

$$\frac{h^2}{2}|f''(\tau)| \leq TOL$$

Rearranging gives:

$$h \leq \sqrt{\frac{2TOL}{|f''(\tau)|}}$$

Let K be the minimum value of $|f''(\tau)|$ over the interval being stepped through. h must then be selected as:

$$h \leq \sqrt{2TOL/K}$$

Improved Euler Method (Heun's Method)

- Perform Euler for $a \rightarrow\ a + h$ to get y*

$$f^*(a+h) = f(a) + hf'(a, y0)$$

- Repeat the Euler calculation but use the average of the slopes at a and a+h:

$$f(a+h) \approx f(a) + h\frac{f'(a, y0) + f'(a+h, y^*)}{2}$$

- Justify this geometrically. What happens for f with an increasing slope? and a decreasing slope?
- Local error of Improved Euler is h^3 and the order of global error is h^2. So, the Improved Euler method is a **second-order method**.

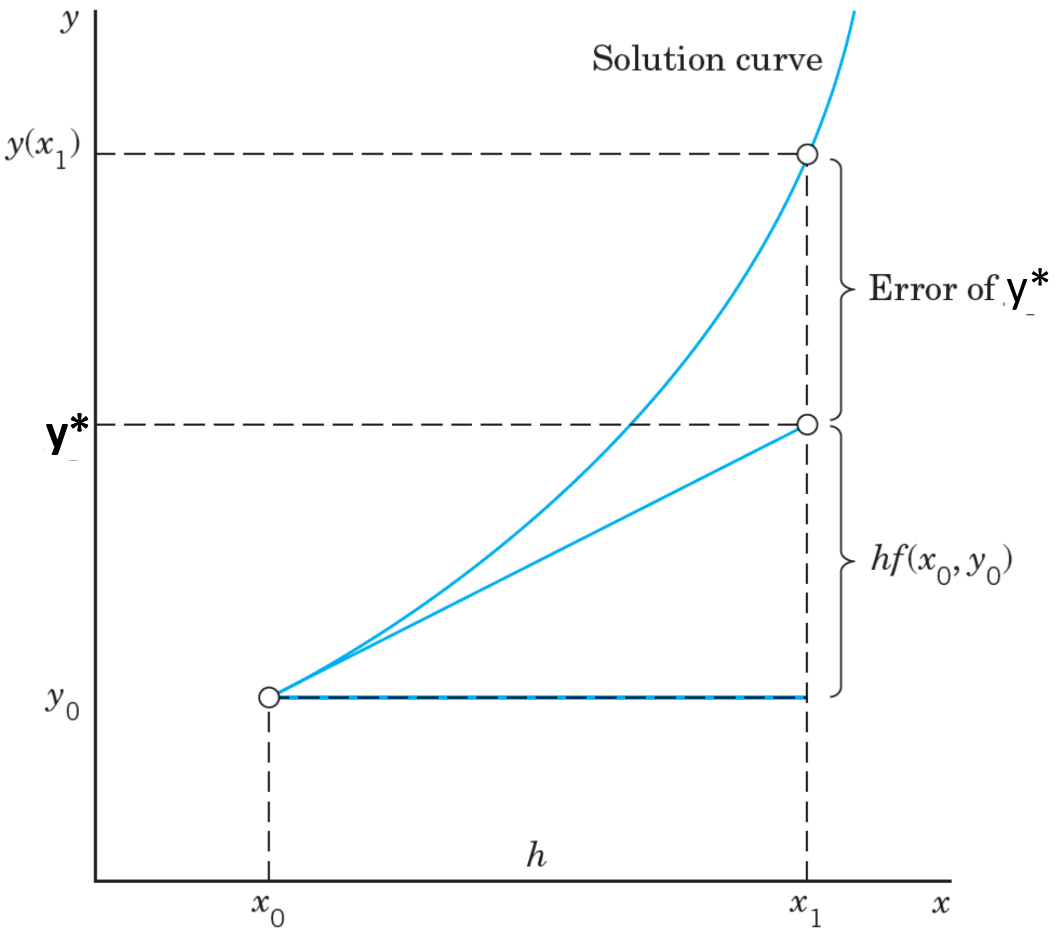

Fig. 8. First Euler step, showing a solution curve, its tangent at (x_0, y_0), step h and increment $hf(x_0, y_0)$ in the formula for y^*

Example

- Use the Improved Euler method to find an approximate value of sin(1), starting with x=0, sin(0)=0, and h=1.

odeerror_examples.m

Exercise

- Use the Improved Euler method to find an approximate value of $\ln(3)$, starting with $x=e$, $\ln(e)=1$, and $h=3-e$.

- Note that $\ln'(x) = 1/x$

Runge-Kutta Methods

- One iteration of Runge-Kutta is as follows:
  - Let $\mathrm{y_a}$: $f(a)$

$$k1 = hf'(a, y_a)$$

$$k2 = hf'\left(a + \frac{h}{2}, y_a + \frac{k1}{2}\right)$$

$$k3 = hf'\left(a + \frac{h}{2}, y_a + \frac{k2}{2}\right)$$

$$k4 = hf'(a + h, y_a + k3)$$

$$f(a + h) \approx y_a + \frac{k1 + 2k2 + 2k3 + k4}{6}$$

- Runge-Kutta methods are **fourth-order**.

Matlab implementations: ode23(), ode45()

- ode23 and ode45 are:
  - automatic step-size Runge-Kutta-Fehlberg integration methods.
  - G.E. Forsythe, M.A. Malcolm and C.B. Moler, *Computer Methods for Mathematical Computations*, Prentice-Hall, 1977.
- ode23: second and third order pair of formulas for medium accuracy
- ode45: fourth and fifth order pair for higher accuracy.
- Automatic step-size:
  - take larger steps where the solution is more slowly changing.
  - ode45 uses higher order formulas, and usually takes fewer integration steps and gives a solution more rapidly.