In-class Exercise 8.2
Error analysis for time-stepping routines
AMATH 301
University of Washington
Jakob Kotas

Truncation error for time-stepping methods

Method	Local error	Global error
Euler's	$O((\Delta t)^2)$	$O(\Delta t)$
Backward Euler's	$O((\Delta t)^2)$	$O(\Delta t)$
Heun's	$O((\Delta t)^3)$	$O((\Delta t)^2)$
Runge-Kutta	$O((\Delta t)^5)$	$O((\Delta t)^4)$

1. Conceptual questions

   (a) What are the two types of error in time-stepping methods?

   - Truncation error, due to only keeping a finite number of terms in Taylor series expansion. Typically written as $O((\Delta t)^k)$ for some k, where higher k means better accuracy.
   - Roundoff error, due to only keeping a finite number of digits on the floating-point numbers while doing calculations. Always present regardless of method.

   Even if truncation error was made very small by using a very sophisticated method, roundoff error will still be present. The sweet spot for many problems is Runge-Kutta.

   (b) Why is global error one power lower than local error?

   Intuitively, because errors accumulate as you take more and more steps.

   Mathematically, there are $N = (b-a)/(\Delta t)$ steps, where a and b are the starting and ending times. If you sum up N local error terms which are $O((\Delta t)^k)$, the global error is

$$O((\Delta t)^k) \cdot \frac{b-a}{\Delta t} = O((\Delta t)^{k-1})$$

   (c) In the 8.1 in-class exercises we looked at the IVP:

$$y' = -y, \;\; y(0) = 1$$

   and found that Euler's method was unstable with $\Delta t = 2$ but stable with $\Delta t = 0.5$. We also found that Backward Euler's method was stable with $\Delta t = 2$. What are the largest Δt we would expect to be stable for this IVP using Euler's and Backward Euler's methods?

   From the Kutz video, Euler's method is unstable (roundoff error grows) for $|1 + \lambda\Delta t| > 1$ and Backward Euler's method is unstable for $\left|\frac{1}{1-\lambda\Delta t}\right| > 1$. The video looked at the IVP $y' = \lambda y$ so here $\lambda = -1$.

   Thus for Euler's method we will have stability if:

$$|1 - \Delta t| < 1$$
$$-1 < 1 - \Delta t < 1$$

$$\Delta t < 2$$

where we drop the left inequality as it is not physically meaningful. We avoid the edge-case exactly at $|1 + \lambda\Delta t| = 1$. Similarly, for Backward Euler's method we will have stability if:

$$\left|\frac{1}{1+\Delta t}\right| < 1$$

$$|1 + \Delta t| > 1$$

$$1 + \Delta t > 1$$

(again, ignoring the case where Δt is negative).

$$\Delta t > 0$$

Meaning any Δt will be stable for Backward Euler's method on this IVP.

2. Consider the initial value problem:

$$\frac{dy}{dt} = \frac{3(t-1)^2}{y}, \;\; y(0) = 2.$$

The analytical solution is:

$$y(t) = \sqrt{2x^3 - 6x^2 + 6x + 4}.$$

   (a) Approximate the solution on $0 \le t \le t_{\text{fin}} = 4$ using Euler's method with a step size of $\Delta t = 0.5$. Save the local error after one step:

   $$\text{absolute local error } = |y_{\text{Euler}}(\Delta t) - y_{\text{true}}(\Delta t)|$$

   and global error after all steps:

   $$\text{absolute global error } = |y_{\text{Euler}}(t_{\text{fin}}) - y_{\text{true}}(t_{\text{fin}})|$$

   (b) Reduce Δt to half its original value and repeat, saving both errors again.

   (c) Repeat until $\Delta t = 2^{-8}$. Also plot the analytical solution. Do the curves approach the analytical solution as $\Delta t \to 0$?

   (d) Make a plot of $\ln(\Delta t)$ vs. $\ln(\text{error})$ for both the local and global errors. Do both decrease as $\Delta t \to 0$?

   (e) Preview of chapter 4: find the slope of the best-fit line through these data sets and print the values. Use `np.polyfit`. Do we see evidence of second-order local error, first-order global error?

3. The Blasius equation is used in fluid dynamics to describe the laminar boundary layer on a semi-infinite plate under constant fluid flow. The stream function $f(x)$ is defined by the differential equation:

$$2\frac{d^3 f}{dx^3} + \frac{d^2 f}{dx^2} f = 0$$

with initial/boundary conditions:

$$f'(0) = 0, \;\; f(0) = 0, \;\; f'(\infty) = 1$$

   (a) Use the shooting method to find a value of $f''(0) = \alpha$ such that $f'(100) \approx 1$ when solving with `solve_ivp`. (We assume that 100 is a large enough time that $f'(100) \approx f'(\infty)$.)

Let $f' = g$, $g' = h$. Then the system of equations is:

$$\frac{dv}{dt} = \frac{d}{dt}\begin{bmatrix} f \\ g \\ h \end{bmatrix} = \begin{bmatrix} g \\ h \\ -fh/2 \end{bmatrix}$$

with initial conditions:

$$\begin{bmatrix} f(0) \\ g(0) \\ h(0) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \alpha \end{bmatrix}$$

and we wish to find α such that $f''(0) = \alpha$ leads to $f'(100) \approx 1$.

4. There are some differential equations where time-stepping methods perform poorly because the number of time-steps required to avoid instability is extremely large. These are called **stiff equations**.

   As an example, solve the IVP below:

$$\begin{cases} x'(t) &= -56x + 55y \\ y'(t) &= 44x - 45y \end{cases}, \qquad x(0) = 1,\ \ y(0) = 0.$$

   on the time interval $0 \le t \le 1000$.

   (a) Use `solve_ivp` with the `RK45` method; use `time.time()` to print the elapsed time. You'll need to `import time` first.

   (b) Repeat part (a) with the `Radau` method, which is designed to perform better on stiff equations. Print the elapsed time.