Introduction to Numerically Solving Differential Equations

Math 2C03
Sec 2.6, 9.1, 9.2 (Zill), Sec 3.1 - 3.3 (Trench)

July 13, 2022

- We are *not* always able to analytically find a solution to differential equations, and/or we must computationally solve a differential equation. To overcome this, we must numerically solve the differential equation and resort to numerical methods to help us approximate solutions
- Numerical approximations are very common in practice, and several online solvers rely on numerically computed differential equations

1 Simple Numerical Method

Suppose we have a first-order initial value problem

$$\begin{cases} \frac{dy}{dx} = f(x, y), \\ y(x_0) = y_0. \end{cases} \tag{1}$$

We can use tangent lines to approximate the solution to Equation (1)! We shall do this, by remembering that we can approximate a derivative by

$$\frac{dy}{dx} \approx \frac{y(x) - y(x_0)}{x - x_0}, \tag{2}$$

and we can use equation of the tangent line

$$y(x) = y_0 + \frac{dy}{dx}\,(x - x_0).$$

Noting that $\frac{dy}{dx} = f(x, y)$ in Equation (1), we have

$$y(x) = y_0 + f(x_0, y_0)\,(x - x_0). \tag{3}$$

It is required that $x - x_0$ be reasonably small, in order to obtain a good approximation of our derivative in Equation (2). Denoting $h = x - x_0$, we can consider this as a "step-size" from our initial value x_0. Thus, using our initial condition $y(x_0) = y_0$ and Equation (3), we can approximate the solution for $y_1 = y(x_0 + h)$ by solving

$$y_1 = y_0 + f(x_0, y_0)\,h.$$

Denoting $x_1 = x_0 + h$, we repeat this process, to find $y_2 = y(x_1 + h) = y(x_0 + 2h)$ by solving

$$y_2 = y_1 + f(x_1, y_1)\, h.$$

This can be defined as a general recursive formula

$$y_{n+1} = y_n + h\, f(x_n, y_n), \tag{4}$$

where $x_n = x_0 + nh$, $n = 0, 1, 2, \ldots$ Using this algorithm of utilizing successive tangent line approximation to find $y_1, y_2, \ldots$ is known as **Euler's Method**.

2 Example of Euler's Method

Use Euler's Method to obtain an approximation of $y(0.8)$ for

$$\begin{cases} y' = y^2 + x^2, \\ y(0) = 1, \end{cases} \tag{5}$$

with a step-size of 0.1.

Solution Given in the problem (5), we are told that $h = 0.1$, and we wish to compute the solution up to $x = 0.8$. Thus, we have $x_0 = 0, x_1 = 0.1, x_2 = 0.2, x_3 = 0.3, \ldots x_7 = 0.7, x_8 = 0.8$. Using Euler's Method: $y_{n+1} = y_n + h\, f(x_n, y_n)$

$$\begin{aligned}
y(0) = y_0 &= 1 \\
y_1 &= y_0 + h\, f(x_0, y_0) \\
&= y_0 + h\,(y_0^2 + x_0^2) \\
&= 1 + (0.1) \cdot (1^2 + 0^2) \\
y(0.1) = y_1 &= 1.1 \\
y_2 &= y_1 + h\, f(x_1, y_1) \\
&= y_1 + h\,(y_1^2 + x_1^2) \\
&= 1.1 + (0.1) \cdot (1.1^2 + 0.1^2) \\
y(0.2) = y_2 &= 1.222 \\
y(0.3) = y_3 &= 1.3753284 \\
y(0.4) = y_4 &= 1.573481220784657 \\
y(0.5) = y_5 &= 1.837065536000854 \\
y(0.6) = y_6 &= 2.199546514357064 \\
y(0.7) = y_7 &= 2.719347001239096 \\
y(0.8) = y_8 &= 3.507831812553902
\end{aligned}$$

3 Refinements

Euler's Method only uses information about the derivative at the "left endpoint" of our interval.

- Instead we can take an average over the interval, where we take a trial step across the interval first using Euler's Method, then use the trial step to improve our approximation. This is called the **Improved Euler's Method**

$$\begin{aligned} \widetilde{y}_{n+1} &= y_n + h\,f(x_n, y_n) && \text{(trial Euler step)} \\ y_{n+1} &= y_n + \frac{1}{2}h\left[f(x_n, y_n) + f(x_{n+1}, \widetilde{y}_{n+1})\right] && \text{(true step)} \end{aligned}$$

- We can further expand this idea, to improve our approximation. Common and practical methods are *Runge-Kutta* methods, which use judiciously chosen steps and weights:

$$\begin{aligned} k_1 &= h\,f\,(x_n, y_n) \\ k_2 &= h\,f\left(x_n + \frac{1}{2}h, y_n + \frac{1}{2}k_1\right) \\ k_3 &= h\,f\left(x_n + \frac{1}{2}h, y_n + \frac{1}{2}k_2\right) \\ k_4 &= h\,f\,(x_n + h, y_n + k_3) \\ y_{n+1} &= y_n + \frac{1}{6}\,\left[k_1 + 2\,k_2 + 2\,k_3 + k_4\right] \end{aligned}$$

  The Runge-Kutta method shown above is the **RK4** method, which is the "workhorse" for many differential equations solvers

4 Errors in Approximations

When using an approximation (such as a numerical method), we must remember it is just that, an approximation. The approximation can be improved upon by refining the step-size, however it should be noted that when using computers we must be careful not to use too small of a step-size (one should be aware of numerical precision and round-off errors).

A natural question that arises when obtaining numerical solutions is: What is my error to the true solution? It is important to know the accuracy of the numerical method you are using.
Local truncation error is the error that arises due to the fact that our discretized solution will not be precisely equal to the continuous problem (also known as formula error or discretization error). Furthermore, this error will occur at each time step and accumulate for each step! For Euler's method we can derive a formula for local truncation error, by using Taylor's formula with remainder. Let $y(x)$ be a $k+1$ differentiable function, on an open interval $x \in I$. Then, we can represent this as a power series, and in particular a Taylor series

$$y(x) = y(a) + y'(a)\frac{x-a}{1!} + \ldots + y^{(k)}(a)\frac{(x-a)^k}{k!} + y^{(k+1)}(c)\frac{(x-a)^{k+1}}{(k+1)!},$$

where $a, c \in I$, and c is a point between a and x. Notice, if we set $k = 1$, $a = x_n$, and $x = x_{n+1}$, then

$$
\begin{aligned}
y(x_{n+1}) &= y(x_n) + y'(x_n)\frac{h}{1!} + y^{(2)}(c)\frac{h^2}{2!}, \\
y(x_{n+1}) &= y_n + h\, f(x_n, y_n) + y^{(2)}(c)\frac{h^2}{2!}, \\
y(x_{n+1}) - y_{n+1} &= y^{(2)}(c)\frac{h^2}{2!}.
\end{aligned}
$$

We usually cannot determine the error exactly, because the value of c is usually unknown, however we obtain an upper bound of the error by

$$\max_{x_n < x < x_{n+1}} \left| y^{(2)}(x) \right| \frac{h^2}{2!}.$$

So, Euler's Method has a local truncation error of $\mathcal{O}(h^2)$.

- Euler's Method's **global truncation error** is first-order accurate, so the error are $\mathcal{O}(h)$. This is good for an introduction to numerical methods, but not practical for use (converges too slowly to true solution). Similarly, Improved Euler's Method is second-order accurate, $\mathcal{O}(h^2)$
- There are higher order methods can be used, however this comes at the higher computational cost to solve differential equations
- The RK4 method is a good balance between low computational cost, while achieving a fourth-order globally accurate method.
- Runge-Kutta methods are also typically numerically stable for several differential equations

To illustrate order of accuracy, we numerically solve the logistic equation

$$\begin{cases} u' = u\,(1-u), \\ u(0) = 2, \end{cases} \tag{6}$$

which has solution $u(t) = \frac{2}{2-e^{-t}}$. The solution to Equation (6) is shown in Figure 1, with $h = 0.1$. We also show the convergence to the true solution in Figure 2 at $T = 2$.

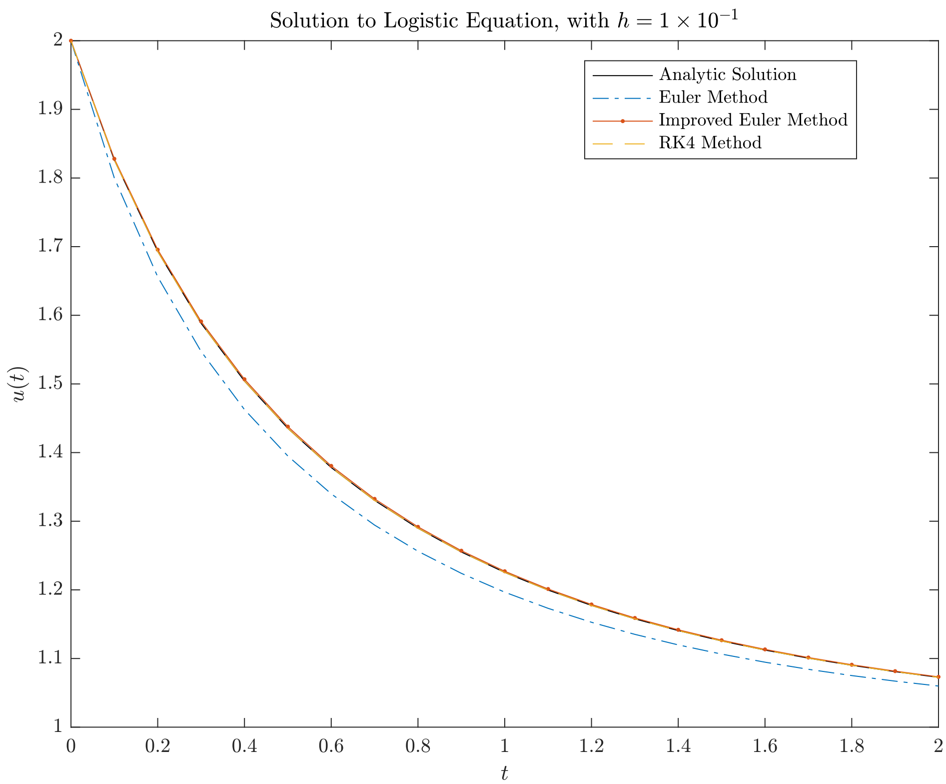

Figure 1: Solution to the logistic initial value problem Equation (6), using $h = 0.1$. Shown is the solution found: analytically (black, solid line), using Euler's method (blue, dashed-dot line), Improved Euler's method (red, solid-dotted line), and RK4 method (yellow, dashed line)

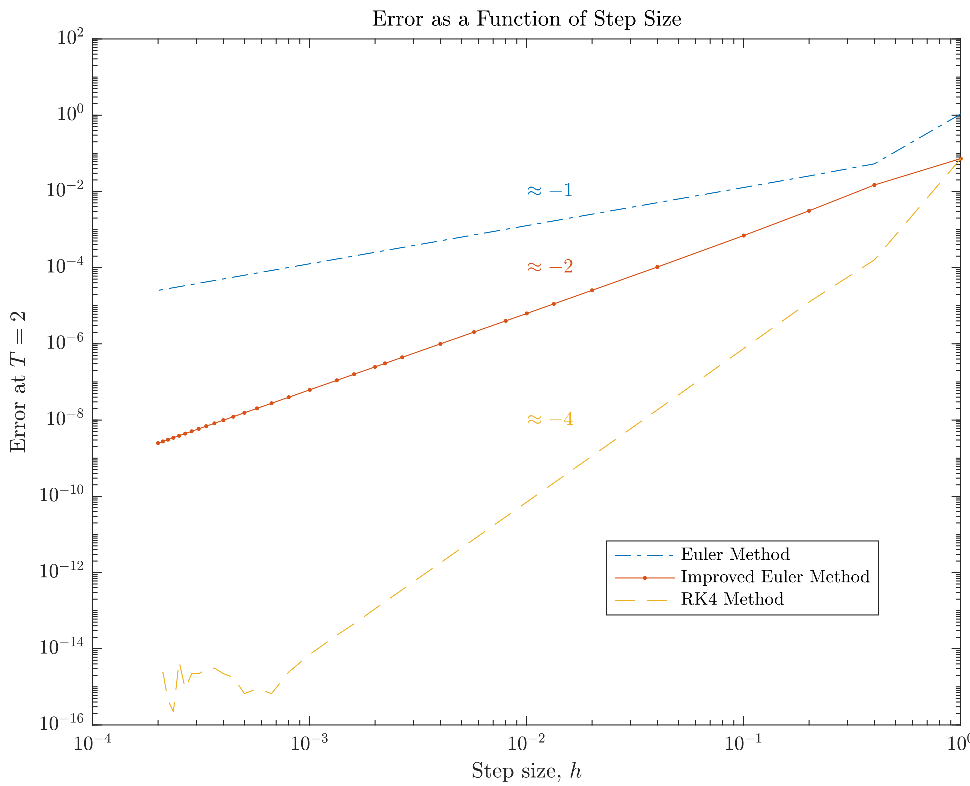

Figure 2: Convergence of the approximated solutions to the analytic solution of Equation (6) at $T = 2$, as step-size h is refined. Shown is the solution given by Euler's method (blue, dashed-dot line), Improved Euler's method (red, solid-dotted line), and RK4 method (yellow, dashed line).

In addition, there are several numerical methods that can be used to solve several problems, and some of them work better than others, depending on the problem! To show this, we can solve this IVP

$$\begin{cases} y' = |x|^3, \\ y(-1) = 0, \end{cases} \tag{7}$$

and we show the convergence of the solution at $x = 1$ in Figure 3. In Figure 4 we show what the grid points are of an example for an equispaced and non-equispaced numerical method.

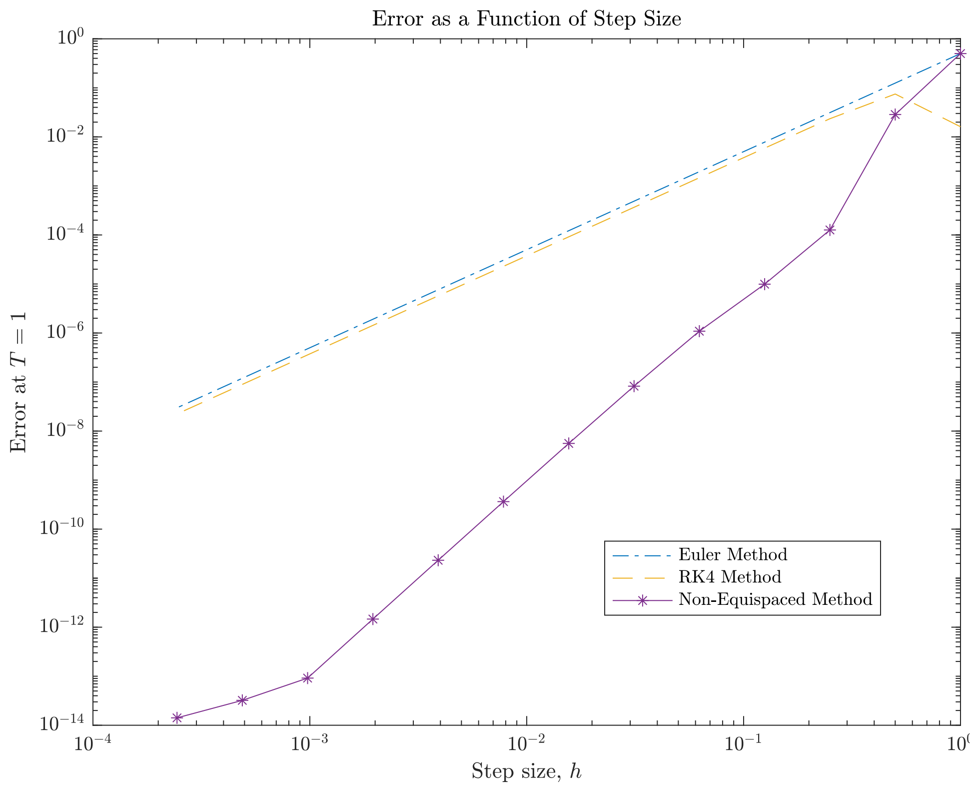

Figure 3: Convergence of the numerical solutions to the true solution of the IVP given in Equation (7) at $T = 1$, as step-size h is refined. Shown is the solution given by Euler's method (blue, dashed-dot line) and RK4 method (yellow, dashed line) with both use Equispaced grids, and a method using a Non-Equispaced grid (purple, solid-star line).

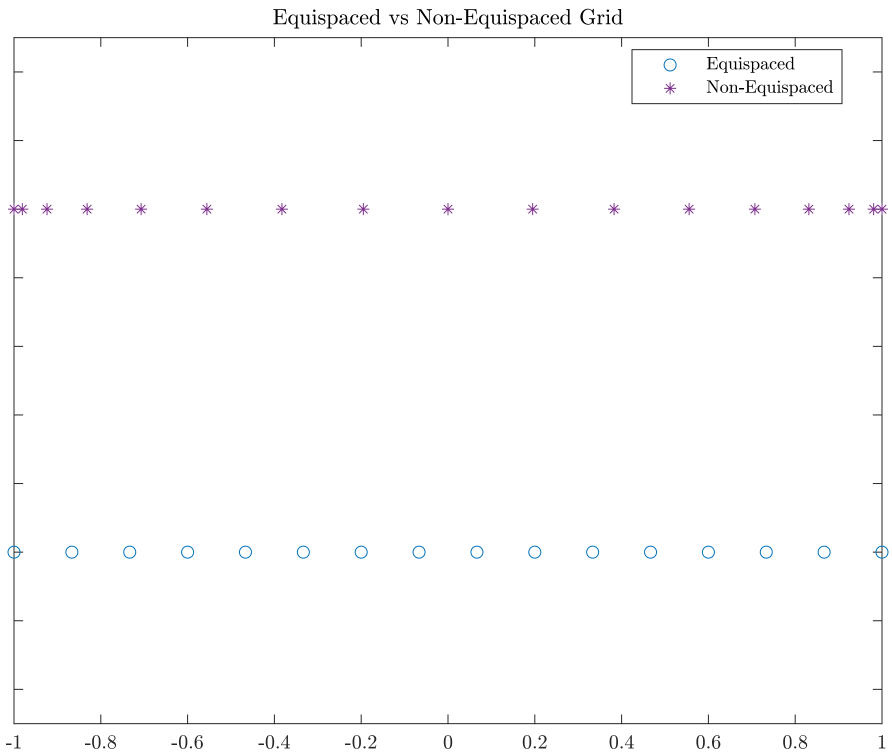

Figure 4: Example of grid using 16 equispaced points (blue, circles) vs using 16 non-equispaced points (purple, stars).

Textbook Problems

Zill:
Sec 4.7 #1, 3
Sec 9.1 #9, 17, 21
Sec 9.2 #10, 13

Trench:
Sec 3.1 #1, 2, 3
Sec 3.2 #1, 2, 3
Sec 3.3 #1, 2, 3