Euler's Method

Math 131: Numerical Analysis

J.C. Meza

April 18, 2024

Section 1

Introduction

Initial-Value Problems for ODEs

The areas we will cover include:

1. General statement of Initial Value Problems, systems of ODEs, etc.
2. Euler's Method including a simple error analysis
3. Higher-order methods
4. Multi-step methods

Roadmap

Let's first start with a roadmap for the lectures to follow. We will:

- be mostly concerned with introducing methods for the solution of IVPs and providing advantages and disadvantages of them
- not discuss problems that are ill-posed, stiff ODE's, or have other structure
- Our reason for this particular focus is that there is a lot of good software available for these problems, so you may never need to actually implement one of these methods.
- Nevertheless, it will be important to know the differences between the methods, what types of problems they can be used on, and the pros and cons of each method.

Some Applications

- Weather/Climate Modeling (primitive equations)
- Chemical reactions (combustion)
- Molecular dynamics simulations

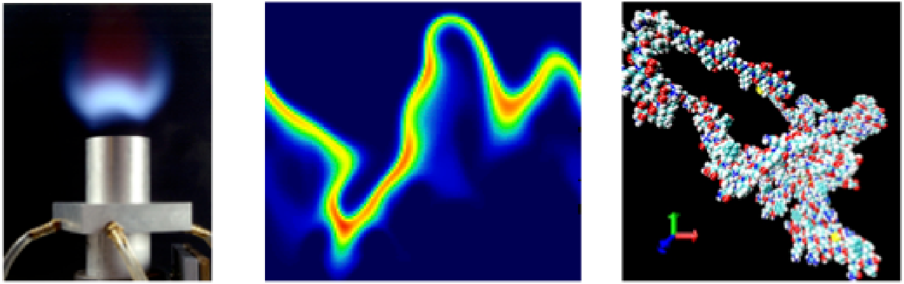


Figure 1a) Jet Flame, b) Combustion simulation, c) protein folding

Figure 2

Existence and Uniqueness

The material on the existence and uniqueness of the IVP is in the supplemental materials section including:

1. Concepts of Lipschitz continuity and convex sets
2. Fundamental Existence and Uniqueness of solutions to the IVP
3. Concept of ***well-posed*** problems

Section 2

IVP

Initial Value Problem

The scalar ***initial-value problem*** (IVP) has the form:

$$y' = \frac{dy}{dt} = f(t, y(t)), \quad a \le t \le b, \quad y(a) = \alpha. \tag{1}$$

In the general case, we would consider a *system* of ODEs, i.e.

$$y' = \frac{dy}{dt} = f(t, y), \quad a \le t \le b, \quad y(a) = \alpha,$$

where

$$y' = \begin{bmatrix} y_1' \\ \vdots \\ y_n' \end{bmatrix} \;\; f(t, y) = \begin{bmatrix} f_1(t, y_1, \dots, y_n) \\ \vdots \\ f_n(t, y_1, \dots, y_n) \end{bmatrix} \tag{2}$$

We will keep it simple for now and assume we have an IVP of the form given by Equation 1.

Higher-order ODEs and Systems of ODEs

- Many science and engineering problems are in the form of higher-order ODES.
- Higher-order ODEs can be reduced to a ***system*** of ODEs
- It can be shown that a general ***n-th*** order ODE:

$$y^{(n)}(t) = f(t, y(t), y'(t), \dots, y^{(n-1)}(t)).$$

can be written in the form of a system of ODEs.

Example 1: Second order ODE

Consider the simple second order IVP given by:

$$y'' + ay' + by = f$$

If we let:

$$\begin{aligned} y_1 &= y \\ y_2 &= y' \end{aligned}$$

then we can rewrite the IVP as two first order ODEs in the form of Equation 2:

$$\begin{aligned} y_1' &= y_2 \\ y_2' &= f - (ay_2 + by_1). \end{aligned}$$

Example 2: Motion of a pendulum

Consider the second order IVP describing the motion of a pendulum:

$$\theta''(t) = -g\sin(\theta(t)),$$

where θ is the angle between the pendulum and the negative vertical axis, g gravity, t is time. If we let:

$$\begin{aligned} y_1(t) &= \theta(t), \\ y_2(t) &= \theta'(t) \end{aligned}$$

then we can rewrite the IVP as a system of two first order ODEs:

$$\begin{aligned} y_1' &= y_2 \\ y_2' &= -g\sin(y_1). \end{aligned}$$

Autonomous Systems

Autonomous Systems

An IVP where the function f does not depend explicitly on t is said to be in ***autonomous form***, i.e.

$$y' = f(y)$$

- Many software packages for the solution of IVPs assume that the function is given in this form.
- This is generally achieved through the addition of an additional equation of the form $t' = 1$.

Section 3

Euler's Method

Euler's Method

We will now present the simplest method for solving an IVP.

Note

For the remainder of the discussion we will assume that our IVP is well-posed (details to follow).

Recall we are looking for solutions to the IVP:

$$\frac{dy}{dt} = f(t, y), \quad a \le t \le b, \quad y(a) = \alpha.$$

As in previous lectures, we will take an approach for numerically solving this problem by approximating it on a discrete grid.

Grids and mesh points

In this case, the points are referred to as ***mesh points*** and are typically of the form:

$$t_i = a + ih, \ i = 0, 1, 2, \ldots, N,$$

The difference between two consecutive points $t_{i+1} - t_i$ is called the step size and is given by:

$$t_{i+1} - t_i = \frac{(t_N - t_0)}{N} = h.$$

Terminology

Many other references use Δt in reference to the time evolution of the IVP, in which case the step size is called the ***time step***.

Derivation of Euler's method

In order to derive Euler's method we can either make use of Taylor's Theorem or just use the numerical approximation for the first derivative that we used previously:

$$y(t_{i+1}) = y(t_i) + hy'(t_i) + \frac{h^2}{2}y''(\xi_i) \quad i = 0, 1, 2, \dots, N-1,$$

where $h = t_{i+1} - t_i$, and $\xi_i \in [t_i, t_{i+1}]$.

Now remember that y' satifies the IVP. As a result, we can rewrite the above equation as:

$$y(t_{i+1}) = y(t_i) + hf(t_i, y(t_i)) + \frac{h^2}{2}y''(\xi_i) \tag{3}$$

(cont.)

Taking the first 2 terms on the right hand side of this equation as our approximation to $y(t_{i+1})$ leads us to propose the following algorithm:

Euler's Method

$$\begin{aligned} y_0 &= \alpha \\ y_{i+1} &= y_i + hf(t_i, y_i), \quad i = 0, 1, \dots, N-1 \end{aligned} \tag{4}$$

Here $y_i \approx y(t_i)$ (approximation to the true solution).

This type of equation is known as a ***difference equation***. You can think of it as being derived from a forward difference approximation to the derivative. Another interpretation is that it is the discretization (in time) of the continuous differential equation.

Section 4

Example

Example

Example

Solve the IVP given by:

$$y' = f(t, y) = y - t^2 + 1 \quad y(0) = 0.5 \quad 0 \leq t \leq 2,$$

with $h = 0.5$.

(cont.)

I find it easier before I start, to write down a table with some of the important variables, where I can keep track of the steps. Something like the following is helpful:

Table 1: Euler Computations

i	t_i	y_i
0	$t_0 = a$	$y_0 = y(a)$
1	$t_1 = a + h$	$y_1 = ...$
2	$t_2 = a + 2h$	
3		
...	...	...
N	$t_N = b$	

(cont.)

I then fill in the initial conditions in the first row and as I compute subsequent y_i I fill in the table with those values.

Table 2: Euler Computations

i	t_i	y_i
0	0	0.5
1	0.5	
2	1.0	
3	1.5	
N	2.0	

Example

- Solve the IVP with a step size of $h = 0.5$ using the built-in ODE solver with method chosen to be "euler".
- Note that this IVP, $f(t, y) = y - t^2 + 1$, has an exact solution given by $(t + 1)^2 - 0.5 \exp(t)$
- We can also compute the true solution and the associated error generated by Euler's method.

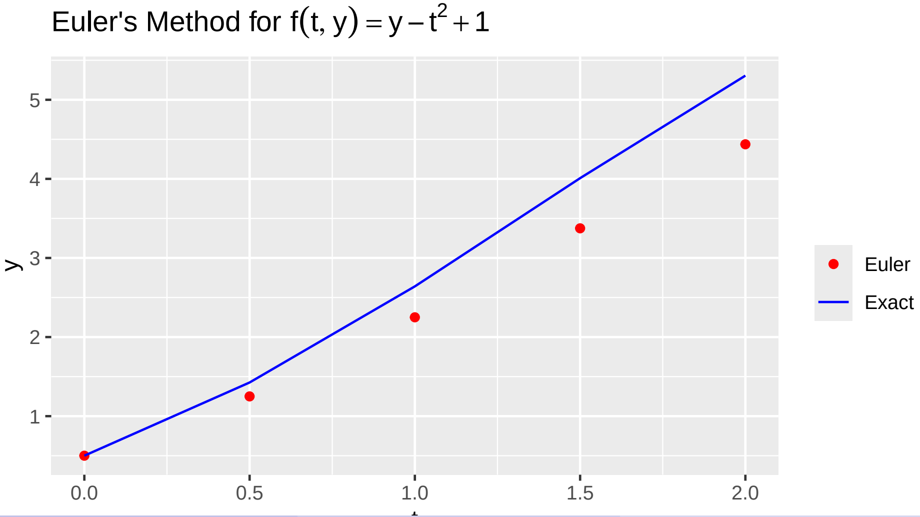
Exact solution versus Euler's method solution

- The table below summarizes the output from the ode solver and compares it to the exact solution.
- What do you notice about the error, especially as time increases?

	time	ysol	yexact	yerr
1	0.0	0.5000	0.500000	0.0000000
2	0.5	1.2500	1.425639	0.1756394
3	1.0	2.2500	2.640859	0.3908591
4	1.5	3.3750	4.009155	0.6341555
5	2.0	4.4375	5.305472	0.8679720

Plots

Let's plot the solution from Euler alongside the exact solution



Experimenting with ICs

Let's explore what happens to the solutions by solving the IVP with several different initial conditions (see the plot below).

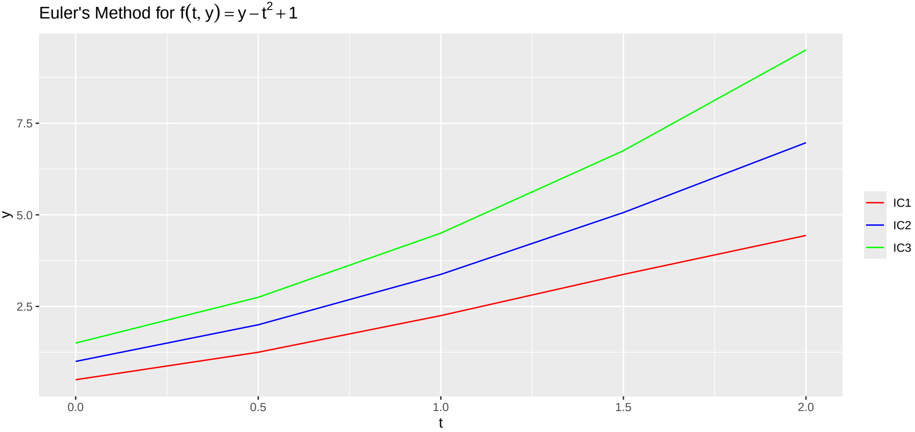


Figure 4: IVP with different ICs

Observations

- One immediate observation is that the IVP generates a family of solutions that can be parameterized by the specific initial condition chosen.
- In this case, also notice that the curves do not converge to a single line.
- What implications would this have on the solutions generated by Euler's method?
- We'll have more to say in later lectures on what is happening and what we can do to help us attain more accurate solutions.

Section 5

Exercise

Exercise

Exercise

Solve the IVP given by:

$$y' = f(t, y) = 1 + (t - y)^2 \quad y(2) = 1.0 \quad 2 \leq t \leq 3$$

with $h = 0.5$ Fill out the table below with your calculations.

Solution

Solution:

Table 3: In class exercise

i	t_i	y_i
0		
1		
2		
3		
...		
N		

Section 6

Backward Euler

Backward Euler

What if we had used a backward difference formula to approximate the derivative of y? In other words:

$$y'(t_i) = \frac{y(t_i) - y(t_{i-1})}{h} = f(t_i, y(t_i))$$

Following the same procedure as before we would have:

$$y(t_i) = y(t_{i-1}) + hf(t_i, y(t_i))$$

Since we really want to compute the approximation at the next time step, let's shift the index by 1:

$$y(t_{i+1}) = y(t_i) + hf(t_{i+1}, y(t_{i+1}))$$

Backward Euler

Letting the approximation to the true solution be denoted by $y_i \approx y(t_i)$, leads to what is known as ***Backward Euler:***

Backward Euler

$$
\begin{aligned}
y_0 &= \alpha \\
y_{i+1} &= y_i + hf(t_{i+1}, y_{i+1}), \quad i = 0, 1, \ldots, N-1
\end{aligned}
$$

Notice that y_{i+1} appears on both the right and left hand sides of this equation. This will therefore require an iterative method to compute the solution at the next time step.

Explicit vs. Implicit Methods

Explicit/Implicit Methods

- *(Forward) Euler's Method is an example of a type of method called an **explicit** method*, because everything we need to compute a quantity at time t_{i+1} is given by known quantities at the previous time step t_i.
- *Backward Euler on the other hand is an example of an* **implicit** *method* since we have y_{i+1} on both sides of the equation.
- There are advantages and disadvantages to both approaches.
- In general, one can take longer timesteps with an implicit method. On the other hand, an implicit method will generally require the solution of a system of equations.

Section 7

Summary

Key Points

- Initial Value Problems arise in many scientific and engineering problems
- Euler's method can be used to solve the IVP by using a forward difference approximation to the derivative of y.
- Forward Euler is easy to implement and relatively cheap computationally.
- Using the backward difference approximation yields a similar method, but requires having to solve the difference equation *implicitly.*