Chapter 5: ORDINARY DIFFERENTIAL EQUATIONS (ODES) INITIAL-VALUE PROBLEMS

5.1 ODE initial value problem statement

A **differential equation** is a relationship between a function, $f(x)$, its independent variable, x, and any number of its derivatives. An **ordinary differential equation** or **ODE** is a differential equation where the independent variable and its derivatives are in one dimension. For the purpose of this book, we assume that an ODE can be written as

$$F\left(x, f(x), \frac{df(x)}{dx}, \frac{d^2f(x)}{dx^2}, \frac{d^2f(x)}{dx^2}, \ldots, \frac{d^{n-1}f(x)}{dx^{n-1}}\right) = \frac{d^nf(x)}{dx^n},$$

where F is an arbitrary function that incorporates one or all of the input arguments, and n is the **order** of the differential equation. This equation is referred to as an n**th order ODE**.

To give an example of an ODE, consider a pendulum of length l with a mass, m, at its end; see Fig. 5.1. The angle the pendulum makes with the vertical axis over time, $\Theta(t)$, in the presence of vertical gravity, g, can be described by the pendulum equation, which is the ODE

y''

$$ml\frac{d^2\Theta(t)}{dt^2} = -mg\sin(\Theta(t)).$$

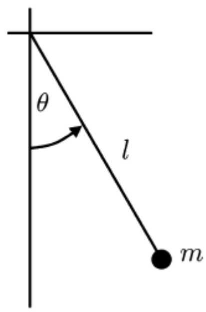

FIGURE 5.1 Pendulum system.

<def> ODE

$f(x,y,y')=c$ implicit

$\boxed{y'=f(x,y)}$ explicit

$f(x,y) \Rightarrow df = \frac{\partial f}{\partial x}dx + \frac{\partial f}{\partial y}dy$ (全微分)

<Thm> $\boxed{\frac{dy}{dx}} = \frac{-M(x,y)}{N(x,y)}$

↓ 同乘 dx

$dy \leftarrow \boxed{\frac{dy}{dx}dx} = \frac{-M(x,y)}{N(x,y)}dx$

$$dy = \frac{-M(x,y)}{N(x,y)}dx$$

分数

$$M(x,y)dx + N(x,y)dy = 0$$

O.D.E

①

$$\frac{dy}{dx} = h(x)\,g(y)$$ 同乘dx

$$\Rightarrow \boxed{\frac{dy}{dx}dx} = h(x)\,g(y)\,dx$$

$$dy = h(x)\,g(y)\,dx \Rightarrow \frac{dy}{g(y)} = h(x)\,dx$$

$$\Rightarrow \int \frac{dy}{g(y)} = \int h(x)\,dx$$

<ex> $\frac{dy}{dx} = g\left(\boxed{\frac{y}{x}}\right)$

令 $u = \frac{y}{x} \Rightarrow y = x \cdot u$

$du = \boxed{\frac{du}{dx}} dx$

$$\frac{dy}{dx} = u + x\frac{du}{dx}$$

$$= u + x\frac{du}{dx} = g(u)$$

$$\Rightarrow x\frac{du}{dx}dx = (g(u) - u)\,dx$$

$$\Rightarrow \frac{dx}{x} = \frac{du}{g(u)-u} \Rightarrow \int \frac{dx}{x} = \int \frac{du}{g(u)-u}$$

<<ep>> $\frac{dy}{dx} = \frac{x^2+y^2}{xy - x^2}$

先整理 同除 x^2 $\Rightarrow \frac{dy}{dx} = \frac{1+(\frac{y}{x})^2}{\frac{y}{x} - 1}$

令 $u = \frac{y}{x} \Rightarrow y = x \cdot u \Rightarrow \frac{dy}{dx} = \boxed{u + x\frac{du}{dx}}$

$$u + x\frac{du}{dx} = \frac{u^2+1}{u-1} \Rightarrow x\frac{du}{dx} = \frac{u^2+1}{u-1} - u$$

$$= \frac{\cancel{u^2}+1}{u-1} - \frac{\cancel{u^2}-u}{u-1}$$

$$= \frac{u+1}{u-1}$$

$$x\frac{du}{dx} = \frac{u+1}{u-1}$$ → 同乘 dx

$$\Rightarrow x\boxed{\frac{du}{dx}dx} = \frac{u+1}{\underline{\underline{u-1}}}dx$$

$$\downarrow$$

$$\underline{\underline{du}}$$

$$[u-2(u+1)]$$

$$= \frac{u-1}{u+1}du = \frac{dx}{x} \Rightarrow \int\frac{u-1}{u+1}du = \int\frac{dx}{x}$$

$$\Rightarrow u - 2\ln|u+1| = \ln|x| + C$$

$$= \frac{y}{x} - 2\ln\left|\frac{y}{x}+1\right| = \ln|x| + C$$

ex. $xy' = y - (y-x)^3$; $y(1) = 2$

令 $u = y - x \Rightarrow y = x + u \Rightarrow \frac{dy}{dx} = 1 + \frac{du}{dx}$

$x(1+\frac{du}{dx}) = y - u^3 \Rightarrow x + x\frac{du}{dx} = y - u^3$

$\Rightarrow x\frac{du}{dx} = \underbrace{(y-x)}_{u} - u^3$

$\Rightarrow x\frac{du}{dx} = u - u^3$ 同除 dx $\Rightarrow \int \frac{du}{u-u^3} = \int \frac{dx}{x}$

$\Rightarrow \ln|u| - \frac{1}{2}\ln|1-u| - \frac{1}{2}\ln|1+u|$

$= \ln|x| + C$

$\Rightarrow 2\ln|u| - \ln|1-u| - \ln|1+u| = 2\ln|x| + C$

$\Rightarrow \ln\left|\dfrac{u^2}{(1-u)(1+u)}\right| = \ln|x^2| + \cancel{C}\ e$

↓同取e

$\Rightarrow \dfrac{u^2}{(1-u)(1+u)} = kx^2$ 代入 $u = y-x$

$\Rightarrow \dfrac{(y-x)^2}{(1-y+x)(1+y-x)} = kx^2$, 代入 $x=1$, $y=2$

$\Rightarrow k =$

線性 ODE $y' + P(x)y = r(x)$

$\Rightarrow \frac{dy}{dx} + [P(x)y - r(x)] = 0$ 同乘 dx

$\Rightarrow \boxed{\frac{dy}{dx}dx} + [P(x)y - r(x)]dx = 0$

$\rightarrow dy$

$dy + [P(x)y - r(x)]dx = 0$，必3積分因子 u

$$\Rightarrow \boxed{u[P(x) - 0] = \frac{\partial u}{\partial x} - [P(x)y - r(x)]\frac{\partial u}{\partial y}}$$

$\downarrow$

$\frac{du}{dx}$

技巧

令 $u=u(x)$ $\quad u(x)P(x)=\frac{du}{dx}$ 同乘 dx

$\Rightarrow u(x)p(x)dx=\boxed{\frac{du}{dx}dx}\;du$

$\Rightarrow p(x)dx=\frac{du}{u(x)} \Rightarrow \int P(x)dx=\ln|u|$

$\Rightarrow u=e^{\int P(x)dx}$ 积分因子

$\Rightarrow d\phi=u[P(x)y-r(x)]dx+u\,dy$

$$\Rightarrow d\phi = \underbrace{e^{\int p(x)dx}[p(x)y - r(x)]}_{\frac{\partial \phi}{\partial x}}dx + \underbrace{e^{\int p(x)dx}}_{\frac{\partial \phi}{\partial y}}dy$$

$$\Rightarrow \frac{\partial \phi}{\partial y} = e^{\int p(x)dx} \Rightarrow \phi = y \cdot e^{\int p(x)dx} + \underbrace{h(x)}_{h'(x)}$$

$$\Rightarrow \frac{\partial \phi}{\partial x} = e^{\int p(x)dx}[p(x)y + r(x)]$$

$$= [y\, p(x) e^{\int p(x)dx} - e^{\int p(x)dx} r(x)]$$

$$h(x) = \int e^{\int p(x)dx} r(x)\, dx$$

$$\phi = y \cdot e^{\int p(x)dx} - \int r(x) e^{\int p(x)dx} dx = c$$

$$\Rightarrow y = e^{-\int p(x)dx}\left[\int r(x) e^{\int p(x)dx} dx + c\right]$$

<Thm> for $y' + P(x) = r(x)$ 线性ODE

→ 恒有 $y = e^{-\int P(x)dx}\left[r(x)e^{\int P(x)dx} + C\right]$

① $(x^2+2x)y' = 2(x+1)y$

$\Rightarrow y' - \frac{2(x+1)}{x^2+2x}y = 0 \Rightarrow P(x) = -\frac{2(x+1)}{x^2+2x}$

$r(x) = 0$

代入 $y = e^{+\int \frac{2(x+1)}{x^2+2x}dx}\left[\cancelto{0}{\int 0 \cdot e^{-\int \frac{2(x+1)}{x^2+2x}dx}} + C\right]$

$$\int \frac{2(x+1)}{x(x+2)} \leftarrow \frac{1}{x} + \frac{1}{x+2} = \frac{x+2+x}{x(x+2)} = \frac{2(x+1)}{x(x+2)}$$

$$= \int \frac{dx}{x} + \int \frac{dx}{x+2} = \ln|x| + \ln|x+2|$$

$$= \ln|(x)(x+2)|$$

$$y = e^{\ln|x\cdot(x+2)|}[c] = c(x\cdot(x+2))$$

$$x \neq 0, \quad x \neq -2$$

<Def> 对 $y''+ay'+by=\underline{\underline{r(x)}}$

猜由 $r(x)$

$r(x)$	y_p
αe^{kx}	$\underline{ce^{kx}}$ 基底
$\begin{cases} \alpha \cos kx \\ \beta \sin kx \end{cases}$	$c_1\cos wx + c_2 \sin wx$ ✓
$\alpha_m x^m + \cdots x^1 + C$	$C_n x^n + C_{n-1} x^{n-1} + \cdots + C_1 x^1 + C_0$

This equation can be derived by summing the forces in the x and y direction, and then changing them to polar coordinates.

In contrast, a **partial differential equation** or **PDE** is a general form differential equation where x is a vector containing the independent variables $x_1, x_2, x_3, \ldots, x_m$, and the partial derivatives can be of any order with respect to any combination of variables. An example of a PDE is the heat equation, which describes the evolution of temperature in space over time:

$$\frac{\partial u(t,x,y,z)}{\partial t} = \alpha\left(\frac{\partial u(t,x,y,z)}{\partial x} + \frac{\partial u(t,x,y,z)}{\partial y} + \frac{\partial u(t,x,y,z)}{\partial z}\right).$$

Here, $u(t, x, y, z)$ is the temperature at (x, y, z) at time t, and α is a thermal diffusion constant.

A **general solution** to a differential equation is a $g(x)$ that satisfies the differential equation. Although there are usually many solutions to a differential equation, they are still difficult to solve. For an ODE of order n, a **particular solution** is a $p(x)$ that satisfies the differential equation and has n explicitly **known values** of the solution, or its derivatives at certain points. Generally stated, $p(x)$ must satisfy the differential equation and $p^{(j)}(x_i) = p_i$, where $p^{(j)}$ is the j th derivative of p, for n triplets, (j, x_i, p_i). For the purpose of this text, we refer to the particular solution simply as the **solution**.

$\lim_{x\to 0} \frac{\sin x}{x} = 1$

Problem 5.1 Returning to the pendulum example, if we assume the angles are very small (i.e., $\sin(\Theta(t)) \approx \Theta(t)$), then the pendulum equation reduces to

$\Theta(t) \Rightarrow y(t)$

$$l\frac{d^2\Theta(t)}{dt^2} = -g\Theta(t).$$

$y_p = C_1 \cos \omega t + C_2 \sin \omega t$

Ans:

$l y'' = -g y(t)$

$\omega^2 = \frac{g}{l}$ $\quad \omega = \sqrt{\frac{g}{l}}$.

$y'' = -\omega^2 y(t)$

解 A.B.

令 $y = A\cos \omega t + B\sin \omega t$

NOTE! An **analytical solution** of an ODE is a mathematical expression of the function $f(x)$ that satisfies the differential equation and has the initial value. But in many cases, an analytical solution is impossible in engineering and science. A numerical solution of an ODE is a set of discrete points (numerical grid) that approximate the function $f(x)$; we can obtain the solution using these grids.

A common set of known values for an ODE solution is the **initial value**. For an ODE of order n, the initial value is a known value for the 0th to $(n-1)$th derivatives at $x = 0$, namely $f(0), f^{(1)}(0), f^{(2)}(0), \ldots, f^{(n-1)}(0)$. For a certain class of ordinary differential equations, the initial value is sufficient to find a unique particular solution. Finding a solution to an ODE given an initial value is called the **initial value problem**. Although the name suggests we will only cover ODEs that evolve in time, initial value problems can also include systems that evolve in other dimensions such as space. Intuitively, the pendulum equation can be solved as an initial value problem because under only the force of gravity, an initial position and velocity should be sufficient to describe the motion of the pendulum for all time afterward.

The remainder of this chapter covers several methods of numerically approximating the solution to initial value problems on a numerical grid. Although initial value problems encompass more than just differential equations in time, we use time as the independent variable. We use several notations for the derivative of $f(t)$: $f'(t)$, $f^{(1)}(t)$, $\frac{df(t)}{dt}$, and $\dot{f}$, whichever is most convenient for the context.

5.2 Reduction of order 降階

Many numerical methods for solving initial value problems are designed specifically to solve first-order differential equations. To make these solvers useful for solving higher-order differential equations, we must often **reduce the order** of the differential equation to a first order problem. To reduce the order of a differential equation, consider a vector, *S(t)*, which is the **state** of the system as a function of time. In general, the state of a system is a collection of all the dependent variables that are relevant to the behavior of the system. Recalling that the ODEs of interest in this book can be expressed as

$$f^{(n)}(t) = F\left(t, f(t), f^{(1)}(t), f^{(2)}(t), f^{(3)}(t), \dots, f^{(n-1)}(t)\right),$$

for initial value problems, it is useful to take the state to be

$$S(t) = \begin{bmatrix} f(t) \\ f^{(1)}(t) \\ f^{(2)}(t) \\ f^{(3)}(t) \\ \vdots \\ f^{(n-1)}(t) \end{bmatrix}.$$

$$\Rightarrow S_2 = \begin{bmatrix} f'(x) \\ \vdots \end{bmatrix}$$

Then the derivative of the state is

$$\frac{dS(t)}{dt} = \begin{bmatrix} f^{(1)}(t) \\ f^{(2)}(t) \\ f^{(3)}(t) \\ f^{(4)}(t) \\ \vdots \\ F\left(t, f(t), f^{(1)}(t), \dots, f^{(n-1)}(t)\right) \end{bmatrix} = \begin{bmatrix} S_2(t) \\ S_3(t) \\ S_4(t) \\ S_5(t) \\ \vdots \\ F\left(t, S_1(t), S_2(t), \dots, S_{n-1}(t)\right) \end{bmatrix},$$

where $S_i(t)$ is the ith element of $S(t)$. With the state written in this way, $\frac{dS(t)}{dt}$ can be written using only $S(t)$ (i.e., no $f(t)$) or its derivatives. In particular, $\frac{dS(t)}{dt} = F\big(t, S(t)\big)$, where F is a function that assembles the vector appropriately, describing the derivative of the state. This equation is in the form of a first-order differential equation in S. Essentially, what we have done is turn an nth order ODE into n first-order ODEs that are **coupled** together, meaning they share the same terms.

Problem 5.2 Reduce the second-order pendulum equation to a first-order equation, where

$$S(t) = \begin{bmatrix} \Theta(t) \\ \dot{\Theta}(t) \end{bmatrix}. = \begin{bmatrix} \dot{\theta}(t) \\ \ddot{\theta}(t) \end{bmatrix}$$

Ans:

$$y'' = \boxed{-\frac{g}{l} y} \qquad \boxed{S = \begin{bmatrix} y \\ y' \end{bmatrix}}$$

$$\Rightarrow S_1 = y$$
$$S_2 = y'$$

$$\frac{ds}{dt} = \begin{bmatrix} y' \\ y'' \end{bmatrix} = \begin{bmatrix} S_2 \\ -\frac{g}{l} y \end{bmatrix} \rightarrow S_1$$

$$= \begin{bmatrix} 0 & 1 \\ -\frac{g}{l} & 0 \end{bmatrix} \begin{bmatrix} S_1 \\ S_2 \end{bmatrix}$$

The ODEs that can be written in this way are said to be **linear ODEs**. Although reducing the order of an ODE to first-order results in an ODE with multiple variables, all the derivatives are still taken with respect to the same independent variable, t ; therefore, the ordinariness of the differential equation is retained. Note that the state can hold multiple dependent variables and their derivatives as long as the derivatives are the same with respect to the independent variable.

Problem 5.3 A very simple model to describe the change in the population of rabbits, $r(t)$, due to wolves, $w(t)$, might be

$$\frac{dr(t)}{dt} = \boxed{4r(t)} - 2w(t)$$

$S_1 = r(t)$

$4S_1 - 2S_2$

and

$S_2 = w(t)$

$$\frac{dw(t)}{dt} = r(t) + w(t).$$

$S_1 + S_2$

Ans:

$$S = \begin{bmatrix} r(t) \\ w(t) \end{bmatrix} \Rightarrow \frac{dS}{dt} = \begin{bmatrix} \frac{dr}{dt} \\ \frac{dw}{dt} \end{bmatrix} = \begin{bmatrix} 4 & -2 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} S_1 \\ S_2 \end{bmatrix}$$

5.3 The Euler method

Let $\frac{dS(t)}{dt} = F(t, S(t))$ be an explicitly defined first order ODE, that is, F is a function that returns the derivative, or change, of a state given a time and state value. Also, let t be a numerical grid of the interval [t_0, t_f] with spacing h. Without loss of generality, we assume that $t_0 = 0$ and that $t_f = Nh$ for some positive integer, N.

The linear approximation of $S(t)$ around t_j at t_{j+1} is

時間步

$$S(t_{j+1}) = S(t_j) + \overset{\Delta t}{(t_{j+1} - t_j)}\frac{dS(t_j)}{dt},$$

which can also be written

目前狀態 + h·F

$$S(t_{j+1}) = S(t_j) + hF\left(t_j, S(t_j)\right).$$

This formula is called the **Explicit Euler Formula**. It allows us to compute an approximation for the state at $S(t_{j+1})$ given the state at $S(t_j)$. This is actually based on the Taylor series we discussed in ~~Chapter 18~~ Chap4, whereby we used only the first order item in Taylor series to linearly approximate the next solution. Later in this chapter, we will present a formula using higher terms to increase the accuracy. Starting from a given initial value of $S_0 = S(t_0)$, we can use this formula to integrate the states up to $S(t_f)$; these $S(t)$ values are then an approximation for the solution of the differential equation.

The explicit Euler formula is the simplest and most intuitive method for solving initial value problems. At any state $\left(t_j, S(t_j)\right)$ it uses F at that state to "point" linearly toward the next state and then moves in that direction a distance of h, as shown in Fig. 5.2.

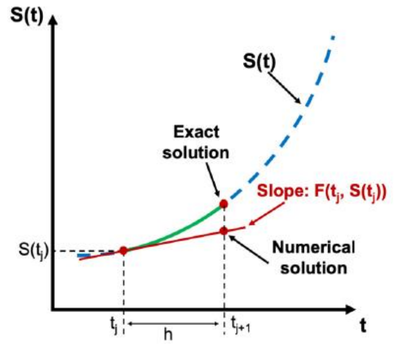

FIGURE 5.2 The illustration of the explicit Euler method.

Although there are more sophisticated and accurate methods for solving these problems, they all have the same fundamental structure. As such, we enumerate explicitly the steps for solving an initial value problem using the explicit Euler formula.

WHAT IS HAPPENING?

Assume we are given a function $F(t, S(t))$ that computes $\frac{dS(t)}{dt}$, a numerical grid, *t*, of the interval, $[\, t_0, t_f \,]$, and an initial state value $S_0 = S(t_0)$. We can compute $S(t_j)$ for every t_j in *t* using the following steps:

1. Store $S_0 = S(t_0)$ in an array, S.
2. Compute $S(t_1) = S_0 + hF(t_0, S_0)$.
3. Store $S_1 = S(t_1)\ in\ S$.
4. Compute $S(t_2) = S_1 + hF(t_1, S_1)$.
5. Store $S_2 = S(t_1)\ in\ S$.
6. . . .
7. Compute $S(t_f) = S_{f-1} + hF(t_{f-1}, S_{f-1})$.
8. Store $S_f = S(t_f)\ in\ S$.
9. S is an approximation of the solution to the initial value problem.

When using a method with this structure, we say the method **integrates** the solution of the ODE.

Problem 5.4 The differential equation $\frac{df(t)}{dt} = e^{-t}$ with initial condition $f_0 = -1$ has the exact solution $f(t) = -e^{-t}$. Approximate the solution to this initial value problem between zero and 1 in increments of 0.1 using the explicit Euler formula. Plot the difference between the approximated solution and the exact solution.

Ans:

2024/4/23

The explicit Euler formula is called "explicit" because it only requires information at t_j to compute the state at t_{j+1}. That is, $S(t_{j+1})$ can be written explicitly in terms of values we have (i.e., t_j and $S(t_j)$). The **Implicit Euler Formula** can be derived by taking the linear approximation of $S(t)$ around t_{j+1} and computing it at t_j :

$$S(t_{j+1}) = S(t_j) + hF(t_{j+1}, S(t_{j+1})).$$

解非线性方程

This formula is peculiar because it requires that we know $S(t_{j+1})$ in order to compute $S(t_{j+1})$! However, it happens that sometimes we *can* use this formula to approximate the solution to initial value problems. Before we provide details on how to solve these problems using the implicit Euler formula, we introduce another implicit formula called the **Trapezoidal Formula**, which is the average of the explicit and implicit Euler formulas:

$$S(t_{j+1}) = S(t_j) + \frac{h}{2}\left(F(t_j, S(t_j)) + F(t_{j+1}, S(t_{j+1}))\right).$$

To illustrate how to solve these implicit schemes, consider again the pendulum equation, which has been reduced to a first-order equation:

$$\frac{dS(t)}{dt} = \begin{bmatrix} 0 & 1 \\ -\frac{g}{l} & 0 \end{bmatrix} S(t).$$

For this equation,

$$F(t_j, S(t_j)) = \boxed{\begin{bmatrix} 0 & 1 \\ -\frac{g}{l} & 0 \end{bmatrix} S(t_j)}.$$

If we plug this expression into the explicit Euler formula, we obtain the following equation:

$$S(t_{j+1}) = S(t_j) + h\begin{bmatrix} 0 & 1 \\ -\frac{g}{l} & 0 \end{bmatrix} S(t_j)$$

$$= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} S(t_j) + h\begin{bmatrix} 0 & 1 \\ -\frac{g}{l} & 0 \end{bmatrix} S(t_j) = \begin{bmatrix} 1 & h \\ -\frac{gh}{l} & 1 \end{bmatrix} S(t_j).$$

Similarly, we can plug the same expression into the implicit Euler formula to obtain

$$\begin{bmatrix} 1 & -h \\ \frac{gh}{l} & 1 \end{bmatrix} S(t_{j+1}) = S(t_j),$$

inv

and into the trapezoidal formula to obtain

$$\begin{bmatrix} 1 & -\frac{h}{2} \\ \frac{gh}{2l} & 1 \end{bmatrix} S(t_{j+1}) = \begin{bmatrix} 1 & \frac{h}{2} \\ \frac{gh}{2l} & 1 \end{bmatrix} S(t_j).$$

With some rearrangement, these equations become respectively

$$S(t_{j+1}) = \begin{bmatrix} 1 & -h \\ \frac{gh}{l} & 1 \end{bmatrix}^{-1} S(t_j),$$

implicit euler

$$S(t_{j+1}) = \begin{bmatrix} 1 & -\frac{h}{2} \\ \frac{gh}{2l} & 1 \end{bmatrix}^{-1} \begin{bmatrix} 1 & \frac{h}{2} \\ -\frac{gh}{2l} & 1 \end{bmatrix} S(t_j).$$

Trapezoidal

These equations allow us to solve the initial value problem since at each state, $S(t_j)$, we can compute the next state at $S(t_{j+1})$. In general, this is possible to do when an ODE is linear.

implicit stability > explicit

5.4 Numerical error and instability

There are two main issues to consider about integration schemes for ODEs: **accuracy** and **stability**. Accuracy refers to a scheme's ability to get close to the exact solution, which is usually unknown, as a function of the step size h. Previous chapters have referred to accuracy using the notation $O(h^p)$. The same notation can be used to solve ODEs. The stability of an integration scheme is its ability to keep the error from growing as it integrates forward in time. If the error does not grow, then the scheme is stable; otherwise it is unstable. Some integration schemes are stable for certain choices of h and unstable for others; these integration schemes are also referred to as unstable.To illustrate issues of stability, we numerically solve the pendulum equation using the explicit and implicit Euler, as well as trapezoidal, formulas.

Problem 5.5 Use the explicit and implicit Euler, as well as trapezoidal, formulas to solve the pendulum equation over the time interval $[\,0, 5\,]$ in increments of 0.1, and for an initial solution of $S_0 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$. For the model parameters using $\sqrt{\frac{g}{l}} = 4$, plot the approximate solution on a single graph.

Ans:

The generated figure above compares the numerical solution to the pendulum problem. The exact solution is a pure cosine wave. The explicit Euler scheme is clearly unstable. The implicit Euler scheme decays exponentially, which is not correct. The trapezoidal method captures the solution correctly, with a small phase shift as time increases.

5.5 Predictor–corrector and Runge–Kutta methods

2024/4/23

5.5.1 Predictor–corrector methods

Given any time and state value, the function, $F(t, S(t))$, returns the change of state $\frac{dS(t)}{dt}$. The **predictor–corrector** methods of solving initial value problems improve the approximation accuracy of non-predictor–corrector methods by querying the F function several times at different locations (predictions). Then, using a weighted average of the results (corrections), updates the state. Essentially, it uses two formulas: **a predictor** and **a corrector**. The predictor is an explicit formula and estimates the solution at t_{j-1} first, i.e., we can use Euler method or some other methods to finish this step. After obtaining the solution $S(t_{j+1})$, we apply the corrector to improve the accuracy. Using the found $S(t_{j+1})$ on the right-hand side of an otherwise implicit formula, the corrector can calculate a new, more accurate solution.

The **midpoint method** has a predictor step:

$$S\left(t_j + \frac{h}{2}\right) = S(t_j) + \frac{h}{2} F\left(t_j, S(t_j)\right),$$

which is the prediction of the solution value halfway between t_j and t_{j+1}.

It then computes the corrector step:

$$S(t_{j+1}) = S(t_j) + hF\left(t_j + \frac{h}{2}, S\left(t_j + \frac{h}{2}\right)\right),$$

which computes the solution at $S(t_{j+1})$ from $S(t_j)$ but uses the derivative from $S\left(t_j + \frac{h}{2}\right)$.

5.5.2 Runge–Kutta methods

Runge–Kutta (RK) methods are among the most widely used methods for solving ODEs. Recall that the Euler method uses the first two terms in Taylor series to approximate the numerical integration, which is linear: $S(t_{j+1}) = S(t_j + h) = S(t_j) + h \cdot S'(t_j)$.

We can greatly improve the accuracy of numerical integration if we keep more terms of the series as

$$S(t_{j+1}) = S(t_j + h) = S(t_j) + S'(t_j)h + \frac{1}{2!}S''(t_j)h^2 + \cdots + \frac{1}{n!}S^{(n)}(t_j)h^n.$$

In order to obtain this more accurate solution, we need to derive the expressions of $S''(t_j), S'''(t_j), \ldots, S^{(n)}(t_j)$. This extra work can be avoided using the RK methods, which are based on truncated Taylor series but do not require computation of these higher derivatives.

5.5.2.1 Second-Order Runge–Kutta Method

Let us first derive the second order RK method. Let $\frac{dS(t)}{dt} = F(t, S(t))$. Then we assume an integration formula the form of

$$S(t+h) = S(t) + c_1 F\big(t, S(t)\big)h + c_2 F\big[t+ph, S(t) + qhF\big(t, S(t)\big)\big]h\,. \qquad (5.1)$$

We can attempt to find these parameters c_1, c_2, p, q by matching the above equation to the second-order Taylor series:

$$S(t+h) = S(t) + S'(t)h + \frac{1}{2!}S''(t)h^2 = S(t) + F\big(t, S(t)\big)h + \frac{1}{2!}F'\big(t, S(t)\big)h^2. \quad (5.2)$$

Note that

$$F'\big(t, s(t)\big) = \frac{\partial F}{\partial t} + \frac{\partial F}{\partial S}\frac{\partial S}{\partial t} = \frac{\partial F}{\partial t} + \frac{\partial F}{\partial S}F\,. \qquad (5.3)$$

Therefore, Eq. (5.2) can be written as

$$S(t+h) = S + Fh + \frac{1}{2!}\left(\frac{\partial F}{\partial t} + \frac{\partial F}{\partial S}F\right)h^2\,. \qquad (5.4)$$

In Eq. (5.1), we rewrite the last term by applying Taylor series in several variables:

$$F\big[t+ph, S(t) + qhF\big(t, S(t)\big)\big] = F + \frac{\partial F}{\partial t}ph + qh\frac{\partial F}{\partial S}F\,,$$

thus Eq. (5.1) becomes

$$S(t+h) = S + (c_1 + c_2)Fh + c_1\left[\ \frac{\partial F}{\partial t}p + q\frac{\partial F}{\partial S}F\ \right]h^2\,. \qquad (5.5)$$

Comparing Eqs. (5.4) and (5.5), we can easily obtain

$$c_1 + c_2 = 1, c_2 p = \frac{1}{2}, c_2 q = \frac{1}{2}. \tag{5.6}$$

Because (5.6) has four unknowns and only three equations, we assign any value to one of the parameters and get the rest of the parameters. One popular choice is:

$$c_1 = \frac{1}{2}, c_2 = \frac{1}{2}, p = 1, q = 1.$$

We can also define:

$$k_1 = F\left(t_j, S(t_j)\right),$$

$$k_2 = F\left(t_j + ph, S(t_j) + qhk_1\right),$$

where we obtain

$$S(t_{j+1}) = S(t_j) + \frac{1}{2}(k_1 + k_2)h.$$

5.5.2.2 Fourth-Order Runge–Kutta Method

A classical method for integrating ODEs with a high order of accuracy is the **Fourth Order Runge–Kutta** (RK4) method. This method uses four points k_1, k_2, k_3, and k_4. A weighted average of these predictions is used to produce the approximation of the solution. The formula is as follows:

$$k_1 = F\left(t_j, S(t_j)\right),$$

$$k_2 = F\left(t_j + \frac{h}{2}, S(t_j) + \frac{1}{2}k_1 h\right),$$

$$k_3 = F\left(t_j + \frac{h}{2}, S(t_j) + \frac{1}{2}k_2 h\right),$$

$1 \quad \frac{1}{2} \quad \frac{1}{2} \quad 1$

$$k_4 = F\left(t_j + h, S(t_j) + k_3 h\right),$$

Therefore, we will have

$$\boxed{S(t_{j+1}) = S(t_j) + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4).}$$

As indicated by its name, the RK4 method is fourth-order accurate, or $O(h^4)$.

5.6 Python ODE solvers

In `SciPy`, there are several built-in functions for solving initial value problems. The most common function is the `scipy.integrate.solve_ivp` function. The function construction is shown below:

Construction:

Let F be a function object to the function that computes

$$\frac{dS(t)}{dt} = F\left(t, S(t)\right),$$

$$S(t_0) = S_0.$$

The variable t is a one-dimensional independent variable (time), $S(t)$ is an n-dimensional vector-valued function (state), and $F\left(t, S(t)\right)$ defines the differential equations; S_0 is an initial value for S. The function F must have the form $dS = F(t, S)$, although the name does not have to be F. The goal is to find the $S(t)$ that approximately satisfies the differential equations given the initial value $S(t_0) = S_0$.

Using the solver to solve the differential equation is as follows:

```
solve_ivp(fun,t_span,s0,method "RK45",t_eval=None)
```

where `fun` takes in the function in the right-hand side of the system; `t_span` is the interval of integration (t_0, t_f) where t_0 is the start and t_f is the end of the interval; s_0 is the initial state. There are a couple of methods to choose from: the default is "`RK45`", which is the explicit Runge–Kutta method of order 5(4). There are other methods you can use as well; see the end of this section for more information; `t_eval` takes in the times at which to store the computed solution, and must be sorted and lie within `t_span`.

Let us try one example below.

Problem 5.6 Consider the ODE

$$\frac{dS(t)}{dt} = \cos(t)$$

for an initial value of $S_0 = 0$. The exact solution to this problem is $S(t) = sin(t)$. Use `solve_ivp` to approximate the solution to this initial value problem over the interval $[0, \pi]$. Plot the approximate solution versus the exact solution and the relative error over time.

Ans:

2024/4/23

Problem 5.7 Using the `rtol` and `atol` to make the difference between the approximate and exact solution less than 10^{-7}.

Ans:

Problem 5.8 Consider the ODE

$$\frac{dS(t)}{dt} = -S(t) \, ,$$

with an initial value of $S_0 = 1$. The exact solution to this problem is $S(t) = e^{-t}$. Use `solve_ivp` to approximate the solution to this initial value problem over the interval $[0, 1]$. Plot the approximate solution versus the exact solution, and the relative error over time.

Ans:

Problem 5.9 Let the state of a system be defined by $S(t) = \begin{bmatrix} x(t) \\ y(t) \end{bmatrix}$, and let the evolution of the system be defined by the ODE

$$\frac{dS(t)}{dt} = \begin{bmatrix} 0 & t^2 \\ -t & 0 \end{bmatrix} S(t) \, .$$

Use `solve_ivp` to solve this ODE for the time interval [0, 10] with an initial value of $S_0 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$. Plot the solution in $(x(t), y(t))$.

Ans:

5.7 Summary

1. Ordinary differential equations (ODEs) are equations that relate a function to its derivatives, and initial value problems are a specific kind of ODE-solving problem.
2. Because most initial value problems cannot be integrated explicitly, they require numerical solutions.
3. There are explicit, implicit, and predictor–corrector methods for numerically solving initial value problems.
4. The accuracy of the scheme used depends on its order of approximation of the ODE.
5. The stability of the scheme used depends on the ODE, scheme, and choice of the integration parameters.