CS5016: Computational Methods and Applications

Ordinary Differential Equations

Department of Computer Science and Engineering
Indian Institute of Technology Palakkad

14 March, 2023

What is an ODE?

An equation involving one or more derivatives of an unknown function.

If all derivatives are taken with respect to a single independent variable we get an **ordinary differential equation**.

The differential equation (ordinary or partial) has order p if p is the maximum order of differentiation in the equation.

A simple order 1 ODE

$$\frac{d(x(t))}{dt} = -x(t)$$

Verify that the function $x(t) = e^{-t}$ satisfies the above ODE.

An example: prey predator dynamics

The Lotka–Volterra equations[1], also known as the predator–prey equations, are a pair of first-order nonlinear differential equations, frequently used to describe the dynamics of biological systems in which two species interact, one as a predator and the other as prey. The populations change through time according to the pair of equations:

$$\frac{d(x(t))}{dt} = \alpha x(t) - \beta x(t)y(t) \qquad \frac{d(y(t))}{dt} = \delta x(t)y(t) - \gamma y(t)$$

$x(t)$ and $y(t)$ denotes number of prey and predators at time t, respectively.

[1] https://en.wikipedia.org/wiki/Lotka-Volterra_equations

Order reduction

An ODE of order $p > 1$ can always be reduced to a system of p equations of order 1.

Consider the following order 3 ODE

$$\frac{d(x(t))}{dt} + x(t)\frac{d^2(x(t))}{dt^2} + 3x(t)^2\frac{d^3(x(t))}{dt^3} = 4x(t)^3$$

The above ODE is equivalent to the following system of order 1 ODEs. Verify!!!

$$\begin{aligned} u(t) + x(t)v(t) + 3x(t)^2\frac{d(v(t))}{dt} - 4x(t)^3 &= 0 \\ u(t) - \frac{d(x(t))}{dt} &= 0 \\ v(t) - \frac{d(u(t))}{dt} &= 0 \end{aligned}$$

The Cauchy problem

An ordinary differential equation in general admits an infinite number of solutions.

For e.g., $\frac{d(x(t))}{dt} = -x(t)$ admits the solution $x(t) = Ce^{-t}$, where C an arbitrary constant.

If we impose the condition $x(0) = 2$, we get a unique solution $x(t) = 2e^{-t}$.

Cauchy problem

Find $x : I \to \mathbb{R}$ such that

$$x^{'}(t) = f(t, x(t)) \quad \forall t \in I \qquad \text{and} \qquad x(t_0) = x_0$$

where I is an interval of $\mathbb{R}$.

If certain conditions are met, the *Cauchy problem* has a unique solution. What are these conditions?

Explicit and implicit solution

The ODE $\frac{d(x(t))}{dt} = -x(t)$ has an explicit solution $x(t) = Ce^{-t}$, i.e., x can be written as a function of t.

Consider the following ODE

$$\frac{d(x(t)}{dt} = \frac{(x(t) - t)}{(x(t) + t)}$$

Show that the following satisfies the above ODE

$$\frac{1}{2}\ln(t^2 + x(t)^2) + \tan^{-1}\frac{x(t)}{t} = C$$

$x(t)$ and t are related according to the above law. However, it is not possible to write $x(t)$ as a function of t.

Euler methods

Subdivide integration interval $I = [t_0, T]$, with $T < \infty$, into N_h intervals of length $h = (T - t_0)/N_h$; h is called the **discretization step**.

At each $t_n, n \in \{0, 1, \ldots, N_h - 1\}$ we seek the unknown value x_n that approximates $x(t_n)$. The set of values $\{x_n\}_{n=0}^{N_h-1}$ is our numerical solution.

Forward Euler method

$$x_{n+1} = x_n + hf(t_n, x_n) \quad \forall n \in \{0, 1, \ldots, N_h - 1\}$$

Backward Euler method

$$x_{n+1} = x_n + hf(t_{n+1}, x_{n+1}) \quad \forall n \in \{0, 1, \ldots, N_h - 1\}$$

Euler methods

Consider the ODE

$$\frac{d(x(t))}{dt} = -x(t)^4$$

Forward Euler method gives

$$x_{n+1} = x_n - h \cdot x_n^4$$

An explicit expression

Backward Euler method gives

$$x_{n+1} = x_n - h \cdot x_{n+1}^4$$

i.e., x_{n+1} should be a real root of the polynomial

$$y^4 - \frac{y}{h} - \frac{x_n}{h} = 0$$

An implicit expression

Implicit methods enjoy better stability properties than explicit ones.

Stability on unbounded intervals

Consider the following

$$x^{'}(t) = \lambda x(t) \quad \forall t \in (0, \infty) \qquad \text{and} \qquad x(0) = 1$$

It is easy to check that $x(t) = e^{\lambda t}$ is the exact solution. Note that if $\lambda < 0$, then $\lim_{t\to\infty} x(t) = 0$.

Forward Euler method with $x_0 = 1$ gives

$$x_{n+1} = x_n(1 + \lambda h) = (1 + \lambda h)^n \quad \forall n \geq 0$$

$\lim_{n\to\infty} x_n = 0$ only if $h \in (0, 2/|\lambda|)$

Backward Euler method with $x_0 = 1$ gives

$$x_{n+1} = x_n/(1 - \lambda h) = 1/(1 - \lambda h)^n \quad \forall n \geq 0$$

$\lim_{n\to\infty} x_n = 0$ for all $h > 0$

Systems of ODEs

Consider the following system of first-order ODEs with unknowns $x_1(t), \ldots, x_m(t)$

$$\begin{aligned} x_1'(t) &= f_1(t, x_1(t), \ldots, x_m(t)) \\ &\vdots \\ x_m'(t) &= f_m(t, x_1(t), \ldots, x_m(t)) \end{aligned}$$

where $t \in (t_0, T]$ with initial conditions $x_{1,0}, \ldots x_{m,0}$.

Let us write the above system of ODEs as

$$\boldsymbol{x}'(t) = \boldsymbol{F}(t, \boldsymbol{x}(t))$$

Now, we can apply any of the methods used to solve the Cauchy problem.

Higher order methods

More sophisticated schemes, such as **Runge-Kutta** methods, can achieve better accuracy.

The SciPy module `scipy.integrate` offers methods to solve ODEs. To know more visit `https://docs.scipy.org/doc/scipy/reference/tutorial/integrate.html`

Thank You