14. Ordinary differential equations

Last time

- Numerical integration (quadrature)
- Interpolate then integrate
- Newton–Cotes methods (equally-spaced points)
- Error analysis

Goals for today

- Ordinary differential equations: review
- Euler method

Ordinary differential equations

- **Ordinary differential equation** (ODEs):
  implicitly defines a function $x(t)$ *by relating* x *and its derivatives*

Ordinary differential equations

- **Ordinary differential equation** (ODEs):
  implicitly defines a function $x(t)$ *by relating* x *and its derivatives*
- e.g.

$$\dot{x} = -\lambda x$$

- Models radioactive decay: $x(t)$ is proportion of radioactive nuclei at time t

Ordinary differential equations

- **Ordinary differential equation** (ODEs):
  implicitly defines a function $x(t)$ *by relating* x *and its derivatives*
- e.g.

$$\dot{x} = -\lambda x$$

- Models radioactive decay: $x(t)$ is proportion of radioactive nuclei at time t
- Need to start somewhere: Initial condition $x(t = t_0) = x_0$

Solution to an ODE

- More general: $\dot{x} = f(x)$

Solution to an ODE

- More general: $\dot{x} = f(x)$
- This means

$$\dot{x}(t) = f(x(t)) \quad \forall t \in [t_0, t_{\text{final}}]$$

Solution to an ODE

- More general: $\dot{x} = f(x)$
- This means

$$\dot{x}(t) = f(x(t)) \quad \forall t \in [t_0, t_{\text{final}}]$$

- Tells us how fast solution changes if currently at given value

Solution to an ODE

- More general: $\dot{x} = f(x)$
- This means

$$\dot{x}(t) = f(x(t)) \quad \forall t \in [t_0, t_{\text{final}}]$$

- Tells us how fast solution changes if currently at given value
- Together with the initial condition, this **implicitly** determines the value of $x(t)$ at all times t

Solution to an ODE

- More general: $\dot{x} = f(x)$
- This means

$$\dot{x}(t) = f(x(t)) \quad \forall t \in [t_0, t_{\text{final}}]$$

- Tells us how fast solution changes if currently at given value
- Together with the initial condition, this **implicitly** determines the value of $x(t)$ at all times t

- Solution is a **function** $x(t)$ for $t \in [t_0, t_{\text{final}}]$

Existence and uniqueness

- Does ODE *really* specify a solution $x(t)$?

Existence and uniqueness

- Does ODE *really* specify a solution $x(t)$?

- (Usually) **yes**!: Existence and uniqueness theorem

Existence and uniqueness

- Does ODE *really* specify a solution $x(t)$?
- (Usually) **yes**!: Existence and uniqueness theorem
- Sufficient condition is $f \in C^1$ (continuous first derivative)

Meaning of an ODE

- Start at initial condition (known)
- ODE tells us derivative of solution at that point

Meaning of an ODE

- Start at initial condition (known)
- ODE tells us derivative of solution at that point
- So we have tangent vector to the solution at $t = t_0$
- i.e. we know *in which direction* we should move

Meaning of an ODE

- Start at initial condition (known)
- ODE tells us derivative of solution at that point
- So we have tangent vector to the solution at $t = t_0$
- i.e. we know *in which direction* we should move
- As soon as we move a bit, must change to new direction!

Euler method

- This suggests a **numerical method**
- We literally follow the above prescription
- Need method to approximate the true, unknown solution
- One possible way (but not the only one, by any means) is to creep forward in small time steps

Time stepping

- Take equal time steps of length h (for now)
- So $t_n := t_n + n\,h$
- Want to approximate true solution $x(t_n)$
- Call approximation x_n
- Calculate sequence of approximations x_0, x_1, ..., x_N

Euler method

- Simplest idea: Suppose derivative constant over time step
- Equivalent: approximate $f(x(s))$ by constant function
- So $x_1 - x_0 = \int_{t_0}^{t_1} f(x_0)\, ds = hf(x_0)$
- Compare rectangular rule for integration
- Repeating this for each step gives the **Euler method**:

$$x_{n+1} = x_n + h\, f(x_n)$$

Convergence of Euler method

- A proposed numerical method like this is worthwhile only if it is **convergent**:
- Call $x_{n,h}$ the solution x_n with time-step h
- As $h \to 0$, the solution produced by the Euler method, i.e. the collection of values $(x_{0,h}, x_{1,h}, \ldots, x_{N(h),h})$, should converge to the true solution $(x(t_0), x(t_1), \ldots, x(t_N))$
- I.e. maximum distance should $\to 0$ as $h \to 0$
- This can be proved correct: See e.g. Iserles, *A First Course in the Numerical Analysis of Differential Equations*

Rate of convergence

- Want **rate of convergence** of error as function of h
- Look at single step and suppose start at exact value:

$$
\begin{aligned}
x(t_{n+1}) &= x(t_n) + h\,\dot{x}(t_n) + \frac{1}{2}h^2\ddot{x}(\xi) \\
&= x(t_n) + hf(x(t_n)) + h^2\ddot{x}(\xi)
\end{aligned}
$$

- So local error is $\mathcal{O}(h^2)$ at each step – **order 1**
- There are $N \sim \frac{1}{h}$ steps so expect global error to be $\mathcal{O}(h)$

Inhomogeneous ODEs

- f can depend on time too:

$$\dot{x}(t) = f(t, x(t))$$

- E.g. if there is periodic external forcing

Inhomogeneous ODEs

- f can depend on time too:

$$\dot{x}(t) = f(t, x(t))$$

- E.g. if there is periodic external forcing

- Then Euler method becomes

$$x_{n+1} = x_n + h_n \, f(t_n, x_n)$$

in general case with different step sizes h_n

Alternative: Trapezium rule

- Can approximate integral using any quadrature method

Alternative: Trapezium rule

- Can approximate integral using any quadrature method
- E.g. trapezium rule:

$$x_{n+1} = x_n + \frac{h}{2}\left[f(x_n) + f(x_{n+1})\right]$$

Alternative: Trapezium rule

- Can approximate integral using any quadrature method
- E.g. trapezium rule:

$$x_{n+1} = x_n + \frac{h}{2}\left[f(x_n) + f(x_{n+1})\right]$$

- How can we find x_{n+1}? Now **implicit** method

Alternative: Trapezium rule

- Can approximate integral using any quadrature method
- E.g. trapezium rule:

$$x_{n+1} = x_n + \frac{h}{2}\left[f(x_n) + f(x_{n+1})\right]$$

- How can we find x_{n+1}? Now **implicit** method
- Must solve nonlinear equation at each step, e.g. using Newton method

Alternative: Trapezium rule

- Can approximate integral using any quadrature method
- E.g. trapezium rule:

$$x_{n+1} = x_n + \tfrac{h}{2}\left[f(x_n) + f(x_{n+1})\right]$$

- How can we find x_{n+1}? Now **implicit** method
- Must solve nonlinear equation at each step, e.g. using Newton method
- More expensive but necessary under certain (common) circumstances: **stiff equations** (see later)

Alternative: Trapezium rule

- Can approximate integral using any quadrature method
- E.g. trapezium rule:

$$x_{n+1} = x_n + \frac{h}{2}\left[f(x_n) + f(x_{n+1})\right]$$

- How can we find x_{n+1}? Now **implicit** method
- Must solve nonlinear equation at each step, e.g. using Newton method
- More expensive but necessary under certain (common) circumstances: **stiff equations** (see later)
- Local error is $\mathcal{O}(h^3)$ and global is $\mathcal{O}(h^2)$

Systems of equations

- Usually there will be > 1 dependent variable, e.g.

$$\dot{x} = f(x, y)$$
$$\dot{y} = g(x, y)$$

- x and y are **coupled** together
- So **cannot** solve equations independently

Systems of equations II

- Rewrite in vector form:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$$

$$\dot{\mathbf{x}}(t) = \mathbf{f}(t, \mathbf{x}(t))$$

- **f** is now a **vector field**

Solving systems of equations

- $\mathbf{x}(t) = (x_1(t), x_2(t), \ldots, x_d(t))$ if d variables
- Taylor expand:

$$x_i(t_k + h) = x_i(t_k) + h\,\dot{x}_i(t_k) + \mathcal{O}(h^2)$$

- Get Euler method

$$\mathbf{x}_{k+1} = \mathbf{x}_k + h\,\mathbf{f}(\mathbf{x}_k)$$

- *Same* method but now with vectors
- *Same* code

Higher derivatives

- How should we deal with higher-order equations (higher derivatives)?
- E.g. damped harmonic oscillator

$$\ddot{x} + b\dot{x} + \omega^2 x = 0$$

Higher derivatives

- How should we deal with higher-order equations (higher derivatives)?
- E.g. damped harmonic oscillator

$$\ddot{x} + b\dot{x} + \omega^2 x = 0$$

- There are some special methods for second-order
- But usually reduce to system of 1st-order equations:

Higher derivatives

- How should we deal with higher-order equations (higher derivatives)?
- E.g. damped harmonic oscillator

$$\ddot{x} + b\dot{x} + \omega^2 x = 0$$

- There are some special methods for second-order
- But usually reduce to system of 1st-order equations:
- Introduce new variable $v := \dot{x}$ to get system

$$\begin{aligned} \dot{x} &= v \\ \dot{v} &= -bv + \omega^2 x \end{aligned}$$

Summary

- Reviewed ordinary differential equations (ODEs)
- Solution is a function
- Approximate solution using time stepping: Euler method