7. Numerical methods for ordinary differential equations

Initial value problems

Outline

1. Definitions and introduction
2. Numerical methods
   - Explicit methods: Euler, Heun, RK4
   - Implicit methods: Implicit Euler, Crank-Nicolson
3. Extension to systems of differential equations
4. Higher-order differential equations

Definitions

- An *ordinary differential equation* (ODE) is one where derivatives of an unknown function of one variable appear.
- The *order* of a differential equation is the highest derivative that appears in it.
  - An equation can be first-order, second-order, ...

Standard form for ODE initial value problems (IVPs)

- First-order: $y'(t) = f(t, y(t))$ and $y(t = t_0) = y_0$ where t_0, y_0 are known values, f is a known function, and $y(t)$ is an unknown function
- Second-order: $y''(t) = f(t, y(t), y'(t))$ and $y(t = t_0) = y_0$ and $y'(t = t_0) = \mathrm{dy}_0$
- Etc. for higher order; in general, an ODE of order n can be written as

$$\frac{d^n y}{dt^n} = f\left(t, y, \frac{dy}{dt}, \frac{d^2 y}{dt^2}, \ldots, \frac{d^{n-1} y}{dt^{n-1}}\right)$$

and the corresponding initial values as

$$y(t_0) = y_0^{(0)}, \frac{dy}{dt}(t_0) = y_0^{(1)}, \frac{d^2 y}{dt^2}(t_0) = y_0^{(2)}, \ldots, \frac{d^{n-1} y}{dt^{n-1}}(t_0) = y_0^{(n-1)}$$

(all n initial values must be given at the same $t = t_0$)

Some examples of ODEs

- First-order:
  - Cooling of something hot
  - Epidemic spread (e.g. the susceptible-recovered-infected [SIR] model)
- Second-order
  - Mechanics (Newton's Second Law), electromagnetism
  - Waves, vibrations (e.g. shear-building model)
- Fourth-order
  - Beam deflection (Euler-Bernoulli theory)

A typical first-order ODE initial-value problem

- Given the ODE $y'(t) = f(t, y(t))$, and knowing that $y(a) = y_0$ (the initial value; the initial point is denoted a or t_0), find $y(b)$ (where b is some other point).
  - Or sometimes we may want to find y at various points between a and b, not just at the one point b
- Example: If $y'(t) = t - y$ and $y(t = 0) = 1$, estimate $y(t = 2)$.
  - So $a = 0, b = 2, y_0 = 1$
- Some ODEs can be solved analytically (cf. your differential equations class). Here, we will concentrate on numerical methods that enable us to find approximate solutions even where there is no analytic solution.

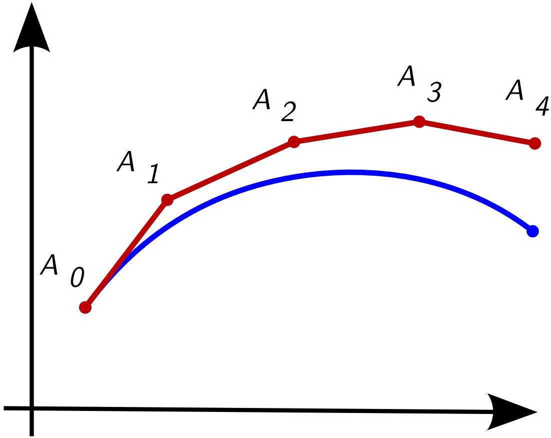
Solving by steps

- Typically, we divide the interval $[a, b]$ into segments by taking points (often equally spaced) $a = t_0, t_1, t_2, \dots t_N = b$ and apply a numerical solution procedure to each segment starting with the initial point
  - Thus, we would start with t_0, where we know that the value of y is y_0. Then we would estimate the value of y at t_1 as y_1.
  - We use this estimated y_1 as the new initial value to estimate y at t_2, calling this estimate y_2
  - And continue until, after N steps, we get y_N, which is the estimated value of y at $t = b$
- Generally, for a given numerical method applied at each step, the smaller each step is, the more accurate our estimate of $y(b)$

Euler's method

- As we saw before, for this numerical method, each step is $y_{i+1} = y_i + hf(t_i, y_i)$, where h is the step size $(t_{i+1} - t_i)$.
- For our example case, if we take $N = 3$ equal steps so that h is $\frac{b-a}{N} = \frac{2}{3}$,
  - $y_1 = y_0 + h(t_0 - y_0) = \frac{1}{3}$
  - $y_2 = y_1 + h(t_1 - y_1) = \frac{5}{9}$
  - $y_3 = y_2 + h(t_2 - y_2) = \frac{29}{27} \approx 1.074$, which is the obtained estimate for $y(t = 2)$.

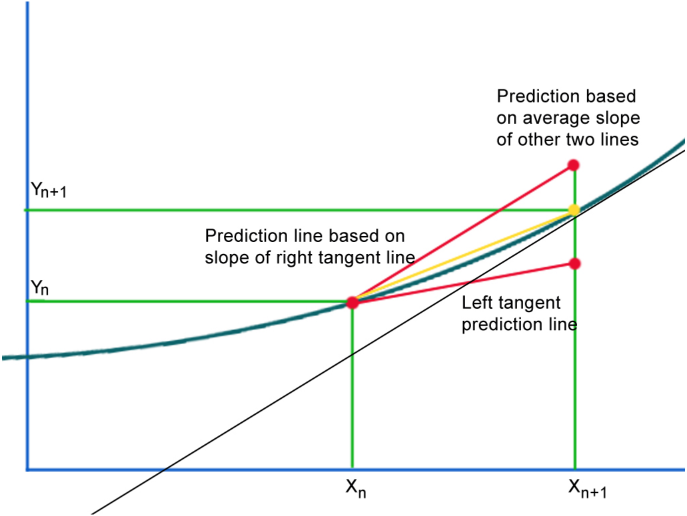
Sketch of several Euler method steps



Heun's method

- This is more elaborate, in that at each step there are two estimates of the average value of f between t_i and t_{i+1}, which are averaged to estimate how much y changes over the step:
- $y_{i+1} = y_i + \frac{1}{2}(K_1 + K_2)$, where
  - $K_1 = hf(t_i, y_i)$
  - $K_2 = hf(t_i + h, y_i + K_1)$
- For our example case, with $N = 3$ equal steps as before,
  - First step: $K_1 = h(t_0 - y_0) = -\frac{2}{3}, K_2 = h(t_1 - (y_0 + K_1)) = \frac{2}{9}, y_1 = y_0 + \frac{1}{2}(K_1 + K_2) = \frac{7}{9}$
  - Second step: $K_1 = h(t_1 - y_1) = -\frac{2}{27}, K_2 = h(t_2 - (y_1 + K_1)) = \frac{34}{81}, y_2 = y_1 + \frac{1}{2}(K_1 + K_2) = \frac{77}{81}$
  - Third step: $K_1 = h(t_2 - y_2) = \frac{62}{243}, K_2 = h(t_3 - (y_2 + K_1)) = \frac{386}{729}, y_3 = y_2 + \frac{1}{2}(K_1 + K_2) = \frac{979}{729} \approx 1.343$, which is the obtained estimate for $y(t = 2)$.

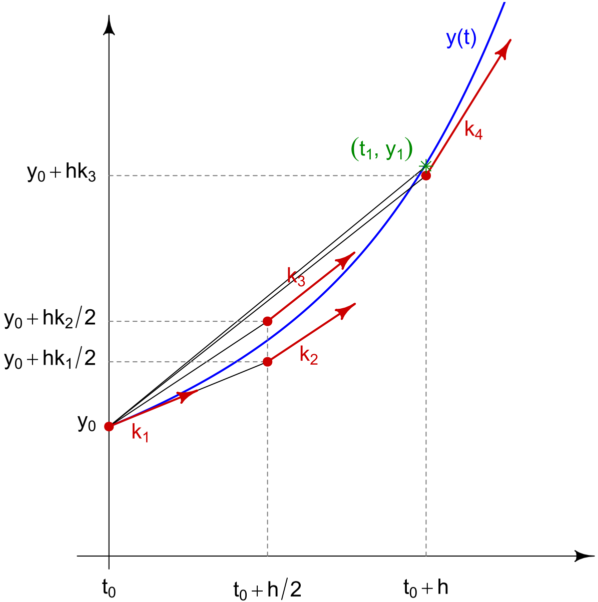
Heun method sketch



The classic fourth-order Runge-Kutta method (RK4)

- Even more elaborate, with 4 estimates per step, 1 for the value of f at its beginning, 2 for the middle, and 1 for the end. between t_i and t_{i+1}:
- $y_{i+1} = y_i + (\frac{K_1}{6} + \frac{K_2}{3} + \frac{K_3}{3} + \frac{K_4}{6})$, where
  - $K_1 = hf(t_i, y_i)$
  - $K_2 = hf(t_i + \frac{h}{2}, y_i + \frac{K_1}{2})$
  - $K_3 = hf(t_i + \frac{h}{2}, y_i + \frac{K_2}{2})$
  - $K_4 = hf(t_i + h, y_i + K_3)$
- For our example case,
  - First step: $K_1 = h(t_0 - y_0) = -\frac{2}{3}, K_2 = h((t_0 + \frac{h}{2}) - (y_0 + \frac{K_1}{2})) = -\frac{2}{9}, K_3 = h((t_0 + \frac{h}{2}) - (y_0 + \frac{K_2}{2})) = -\frac{10}{27}, K_4 = h((t_1 - (y_0 + K_3)) = \frac{2}{81}, y_1 = y_0 + (\frac{K_1}{6} + \frac{K_2}{3} + \frac{K_3}{3} + \frac{K_4}{6}) = \frac{169}{243}$
  - Second step: $K_1 = -\frac{14}{729}, K_2 = \frac{458}{2187}, K_3 = \frac{874}{6561}, K_4 = \frac{6622}{19683}, y_2 = \frac{101866}{118098} \approx 0.8626$
  - Third step: $K_1 \approx 0.3139, K_2 \approx 0.4315, K_3 \approx 0.3923, K_4 \approx 0.4968, y_3 \approx 1.2722$, which is the obtained estimate for $y(t = 2)$.

RK4 method sketch



Accuracy of these numerical methods

- For each method, the error in each step can be thought of as truncation of a Taylor series (as we saw explicitly for the Euler method), with a remainder term that has a derivative of f times a power of h
- The error in $y_N = y(b)$ for each method is typically proportional to a positive power of h (less error if more steps are taken so that h is smaller). This power is the *o*rder of accuracy of the method (which is not the same as the order of the differential equation, explained above)
  - Euler's method has first-order accuracy
  - Heun's method has second-order accuracy
  - The RK4 method has fourth-order accuracy (hence its name)

Implicit methods

- The methods so far for solving ODE IVPs were *explicit*, in that at each step, a formula for y_{i+1} is given in terms of quantities from the previous step
- By contrast, *implicit* methods have at each step a general formula for y_{i+1} that is not given in terms of only known values. This makes them more complicated to apply, but for some problems they can be accurate at larger step sizes, compared to explicit methods.
- The implicit methods we will discuss are
  - The implicit Euler method, which has first-order accuracy
  - The Crank-Nicolson method, which has second-order accuracy

Implicit Euler method

- Each step is $y_{i+1} = y_i + hf(t_{i+1}, y_{i+1})$
- Note that y_{i+1} is unknown and appears on both sides of the equation, which is what makes this method implicit
- At least where f is a simple enough function, we can nevertheless find y_{i+1} at each step
- For our example, where $f(t, y) = t - y$, each step is $y_{i+1} = y_i + h(t_{i+1} - y_{i+1})$, which we can write as $y_{i+1} = \frac{y_i + ht_{i+1}}{1+h}$. So
  - $y_1 = \frac{13}{15}$
  - $y_2 = \frac{79}{75}$
  - $y_3 = 1.432$, which is the obtained estimate for $y(t = 2)$.
- Note that where the usual (explicit) Euler method gives an overestimate of the solution, the implicit method will usually give an underestimate, and vice versa.

Crank-Nicolson method

- Each step is $y_{i+1} = y_i + \frac{h}{2}(f(t_i, y_i) + f(t_{i+1}, y_{i+1}))$ – can be thought of as averaging the explicit and implicit Euler methods
- Note that y_{i+1} is unknown and appears on both sides of the equation, which is what makes this method implicit
- At least where f is a simple enough function, we can nevertheless find y_{i+1} at each step
- For our example, where $f(t, y) = t - y$, each step is $y_{i+1} = y_i + \frac{h}{2}(t_i + t_{i+1} - y_i - y_{i+1})$, which we can write as $y_{i+1} = \frac{(1-\frac{h}{2})y_i + \frac{h}{2}(t_i + t_{i+1})}{1+\frac{h}{2}}$. So
  - $y_1 = \frac{2}{3}$
  - $y_2 = \frac{5}{6}$
  - $y_3 = 1.25$, which is the obtained estimate for $y(t = 2)$.
- Tends to give results in between (and more accurate than either) the two Euler methods

Solution accuracies

- Sometimes, we can find the error of the numerical solution exactly, because the true answer can be calculated
- For the example problem, the general solution to the differential equation is $y(t) = Ce^{-t} + t - 1$. Putting in the initial value tells us that $C = 2$. So $y(2) = 2e^{-2} + 2 - 1 \approx 1.2707$
- The fractional error of the different methods with $h = \frac{2}{3}$ is therefore
  - Euler: 0.15
  - Implicit Euler: 0.13
  - Crank-Nicolson: 0.016
  - Heun: 0.057
  - RK4: 0.0012
- This is the usual pattern: for the same step size, the more complicated higher-order methods give better accuracy

Stability

- In some cases, explicit methods require a very small step size, otherwise the computed solution goes to infinity after many steps (*numerical instability*), while implicit methods can give a moderately accurate solution with a larger step size.
- The classic example of this is the ODE $y'(t) = -ky$, where k is some given positive number, and initial value $y(0) = 1$
- The analytic solution is $y(t) = e^{-kt}$ (exponential decay)
- Applying the Euler method to this problem gives $y_{i+1} = y_i + h(-ky_i) = (1 - hk)y_i$. So after n steps, the computed solution is $y_n = (1 - hk)^n$. This is a good approximation if h is small enough. But if $h > \frac{2}{k}$, then $|1 - hk| > 1$, so $\lim_{i \to \infty} |y_i| = \infty$
- By contrast, the implicit Euler method computed solution correctly goes to zero no matter how large h is: $y_{i+1} = y_i + h(-ky_{i+1})$, so $y_{i+1} = \frac{y_i}{1+hk}$, and $y_n = \frac{1}{(1+hk)^n}$

Estimating solution accuracy

- In most cases, using higher-order accurate methods or taking smaller step sizes improves accuracy, but requires more computation to solve a problem.
- We don't know the analytic solution to most ODE problems. So we need to estimate the error of numerical solutions to assess whether they are accurate enough.
- The usual way to do this is to compare the computed solution using two different numerical methods, or one numerical method with the different step sizes
- For example, Python's `scipy.integrate.solve_ivp` by default applies a 4th-order and a 5th-order accurate RK method at each step, and then compares the two to estimate the error for that step. If the error is too big, then the step is made smaller for more accuracy. If the error is smaller than required, the next step is made bigger to reduce the amount of computation.

Types of numerical methods for ODE IVPs

- All the methods discussed here are examples of *Runge-Kutta methods*.
  - The lecture notes give the general definition for a Runge-Kutta method, though you won't be required to know the details
- Other types of numerical methods for solving ODE IVPs, such as *predictor-corrector* methods, are also used, but will not be discussed in this class

Systems of first-order ODEs

- So far we saw how to use numerical methods for solving single first-order ODEs.
- However, engineering problems often involve systems of ODEs, as we saw, e.g., with the shear building model
- The same methods can be used to solve such systems also.
- Standard form for first-order ODE IVP systems is

$$\begin{cases} z_1'(t) = f_1(t, z_1(t), z_2(t) \cdots z_n(t)); z_1(t = t_0) = z_{1,0} \\ z_2'(t) = f_2(t, z_1(t), z_2(t) \cdots z_n(t)); z_2(t = t_0) = z_{2,0} \\ \cdots \\ z_n'(t) = f_n(t, z_1(t), z_2(t) \cdots z_n(t)); z_n(t = t_0) = z_{n,0} \end{cases}$$

  - Each unknown function $z_i(t)$ has its own first-order differential equation (involving a known function f_i) and initial value
  - All initial values are at a common t_0
  - The problem might be to find the values of some or all of the unknown functions at $t = b$
- More compactly, we can also write such systems in standard form using vector notation, as

$$\mathbf{z}'(t) = \mathbf{f}(t, \mathbf{z}); \mathbf{z}(t = t_0) = \mathbf{z}_0$$

Example with Euler method

- Suppose we have $\begin{cases} z_1'(t) = t - z_2; z_1(0) = 3 \\ z_2'(t) = z_1; z_2(0) = 2 \end{cases}$ or, in vector form, $\mathbf{f} = \begin{pmatrix} t - z_2 \\ z_1 \end{pmatrix}, t_0 = 0, \mathbf{z}_0 = \begin{pmatrix} 3 \\ 2 \end{pmatrix}$
- To find $\mathbf{z}(2)$ with 3 equal-length steps ($h = \frac{2}{3}$), we would do
  - First step: $\mathbf{z}_1 = \mathbf{z}_0 + h \begin{pmatrix} t_0 - z_{2,0} \\ z_{1,0} \end{pmatrix} = \begin{pmatrix} \frac{5}{3} \\ 4 \end{pmatrix}$
  - Second step: $\mathbf{z}_2 = \mathbf{z}_1 + h \begin{pmatrix} t_1 - z_{2,1} \\ z_{1,1} \end{pmatrix} = \begin{pmatrix} -\frac{5}{9} \\ \frac{46}{9} \end{pmatrix}$
  - Third step: $\mathbf{z}_3 = \mathbf{z}_2 + h \begin{pmatrix} t_2 - z_{2,2} \\ z_{1,2} \end{pmatrix} = \begin{pmatrix} -\frac{83}{27} \\ \frac{128}{27} \end{pmatrix}$, which is the obtained estimate for $\mathbf{z} = \begin{pmatrix} z_1 \\ z_2 \end{pmatrix}$ at $t = 2$.

Example with Heun method

- For the same system and step size, we would do
  - First step:

    $\mathbf{K}_1 = h \cdot \mathbf{f}(t_0, \mathbf{z}_0) = \begin{pmatrix} -\frac{4}{3} \\ 2 \end{pmatrix}, \mathbf{K}_2 = h \cdot \mathbf{f}(t_1, \mathbf{z}_0 + \mathbf{K}_1) = \begin{pmatrix} -\frac{20}{9} \\ \frac{10}{9} \end{pmatrix}, \mathbf{z}_1 = \mathbf{z}_0 + \frac{1}{2}\mathbf{K}_1 + \frac{1}{2}\mathbf{K}_2 = \begin{pmatrix} \frac{11}{9} \\ \frac{32}{9} \end{pmatrix}$
  - Second step:

    $\mathbf{K}_1 = h \cdot \mathbf{f}(t_1, \mathbf{z}_1) = \begin{pmatrix} -\frac{52}{27} \\ \frac{22}{27} \end{pmatrix}, \mathbf{K}_2 = h \cdot \mathbf{f}(t_2, \mathbf{z}_1 + \mathbf{K}_1) = \begin{pmatrix} -\frac{164}{81} \\ -\frac{38}{81} \end{pmatrix}, \mathbf{z}_2 = \mathbf{z}_1 + \frac{1}{2}\mathbf{K}_1 + \frac{1}{2}\mathbf{K}_2 = \begin{pmatrix} -\frac{61}{81} \\ \frac{302}{81} \end{pmatrix}$
  - Third step: $\mathbf{K}_1 = h \cdot \mathbf{f}(t_1, \mathbf{z}_1) = \begin{pmatrix} -\frac{388}{243} \\ -\frac{122}{243} \end{pmatrix}, \mathbf{K}_2 = h \cdot \mathbf{f}(t_2, \mathbf{z}_1 + \mathbf{K}_1) = \begin{pmatrix} -\frac{596}{729} \\ -\frac{1142}{729} \end{pmatrix},$

    $\mathbf{z}_3 = \mathbf{z}_2 + \frac{1}{2}\mathbf{K}_1 + \frac{1}{2}\mathbf{K}_2 = \begin{pmatrix} -\frac{1429}{729} \\ \frac{1964}{729} \end{pmatrix}$ which is the obtained estimate for $\mathbf{z} = \begin{pmatrix} z_1 \\ z_2 \end{pmatrix}$ at $t = 2$.

Converting higher-order ODEs to first order

- Engineering problems can also involve higher derivatives, e.g., with the shear building model and with differential equations for beam deflection
- To use the above methods to solve a system with one or more higher-order ODEs, each ODE of order $n > 1$ needs to be converted into an equivalent system of n first-order ODEs.

Converting to first order: procedure

- Start with the higher-order ODE in standard form:

$$\frac{d^n y}{dt^n} = f\left(t, y, \frac{dy}{dt}, \frac{d^2 y}{dt^2}, \ldots, \frac{d^{n-1} y}{dt^{n-1}}\right)$$

  with initial conditions

$$y(t_0) = y_0^{(0)}, \frac{dy}{dt}(t_0) = y_0^{(1)}, \frac{d^2 y}{dt^2}(t_0) = y_0^{(2)}, \ldots, \frac{d^{n-1} y}{dt^{n-1}}(t_0) = y_0^{(n-1)}$$

- Define a vector of unknown functions

$$\mathbf{z} = (z_1, z_2, z_3, \ldots, z_n) \equiv \left(y, \frac{dy}{dt}, \frac{d^2 y}{dt^2}, \ldots, \frac{d^{n-1} y}{dt^{n-1}}\right)$$

- Write the system of first-order ODEs in terms of these unknowns as
  $z_1' = z_2,$
  $z_2' = z_3, \ \ldots$
  $z_{n-1}' = z_n,$
  $z_n' = f(t, z_1, z_2, z_3, \ldots, z_n)$
- Write the initial conditions as $z_i(t_0) = y_0^{(i-1)}$, $i = 1, 2, \ldots, n$

Converting to first order: example

- Pendulum motion (with friction and a driving force) second-order ODE: $\frac{d^2\theta}{dt^2} + c\frac{d\theta}{dt} + \frac{g}{L}\sin(\theta) = a\sin(\Omega t)$
- In standard form: $\frac{d^2\theta}{dt^2} = a\sin(\Omega t) - c\frac{d\theta}{dt} - \frac{g}{L}\sin(\theta)$
- New unknown functions: $z_1 = \theta, z_2 = \theta'$
- New system of first-order ODEs:
  $\begin{cases} z_1'(t) = z_2 \\ z_2'(t) = a\sin(\Omega t) - \frac{g}{L}\sin(z_1) - cz_2 \end{cases}$
- If the initial values are given as, e.g., $\theta(t_0) = 0, \theta'(t_0) = 0.1$, we would convert them to $\begin{cases} z_1(t_0) = 0 \\ z_2(t_0) = 0.1 \end{cases}$