13 Ordinary Differential Equations (ODEs)

- Differential equations are used to model systems that change.
- They may change with respect to time (dynamical systems) or with respect to any other variable.
- We formulate ordinary differential equations by using regular (ordinary) derivatives, in other words we only have 1 independent variable.
- When we have multiple dependent variables we can produce a system of ODEs.
- When more than 1 independent variable is involved (for example time with spatial coordinates) we must formulate partial differential equations (PDEs).

13.1 Euler's Method (Forward)

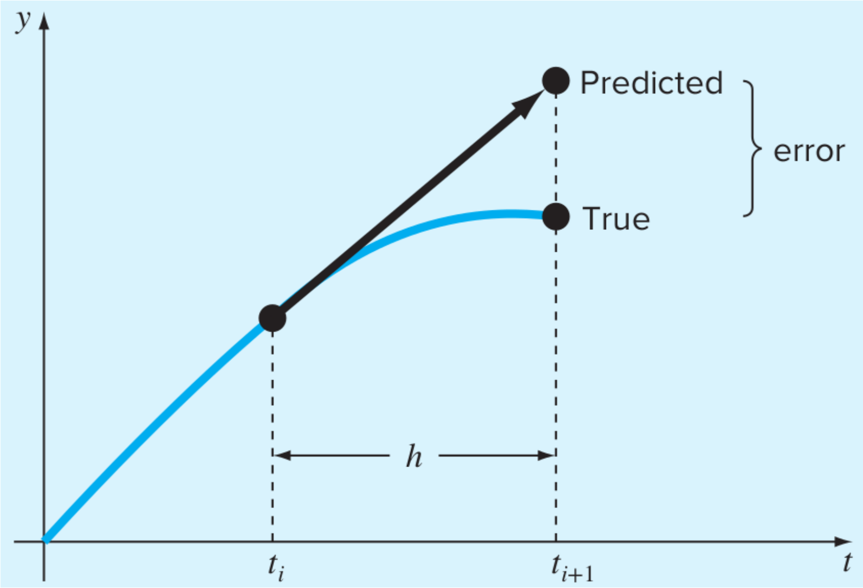
- We aim to solve ODEs of the form: $$\frac{dy}{dt} = f(t, y)$$
- The general idea is to pick an initial value of the independent variable at which we know the solution value ($y(t)$), then take a linear approximation at that point in order to obtain a new solution value for a small increment in the dependent variable.

$$y_{i+1} = y_i + \phi h$$

- The slope, Φ, is called the **increment function**.
- If our step size is small enough then the linear approximation will be a good enough approximation of the actual solution.
- This is a **one-step method** because the next value is estimated from the information at a single point.

- We take the first derivative as the increment function (slope):

$$\phi = f(t_i, y_i) \longrightarrow y_{i+1} = y_i + f(t_i, y_i)h$$



EXAMPLE 1 Use Euler's method to solve the following 1st order ODE for $0 \leq t \leq 4$ with step size $h = 1$ and initial condition $y(0) = 2$.

$$y' = 4e^{0.8t} - 0.5y \longleftarrow \boldsymbol{f(t,y)}$$

First step in time

$$y(1) = y(0) + f(0, 2)(1) \longleftarrow f(0, 2) = 4e^0 - 0.5(2) = 3$$

$$= 2 + 3(1) = 5$$

Second step in time

$$y(2) = y(1) + f(1, 5)(1)$$

$$= 5 + [4e^{0.8(1)} - 0.5(5)]\,(1) = 11.40216$$

Similarly for 3rd and 4th time steps

t	y_{true}	y_{Euler}	$\lvert\varepsilon_t\rvert$ (%)
0	2.00000	2.00000	
1	6.19463	5.00000	19.28
2	14.84392	11.40216	23.19
3	33.67717	25.51321	24.24
4	75.33896	56.84931	24.54

Get true relative error using the exact solution

$$y_{\mathrm{true}} = \frac{4}{1.3}\left(e^{0.8t} - e^{-0.5t}\right) + 2e^{-0.5t}$$

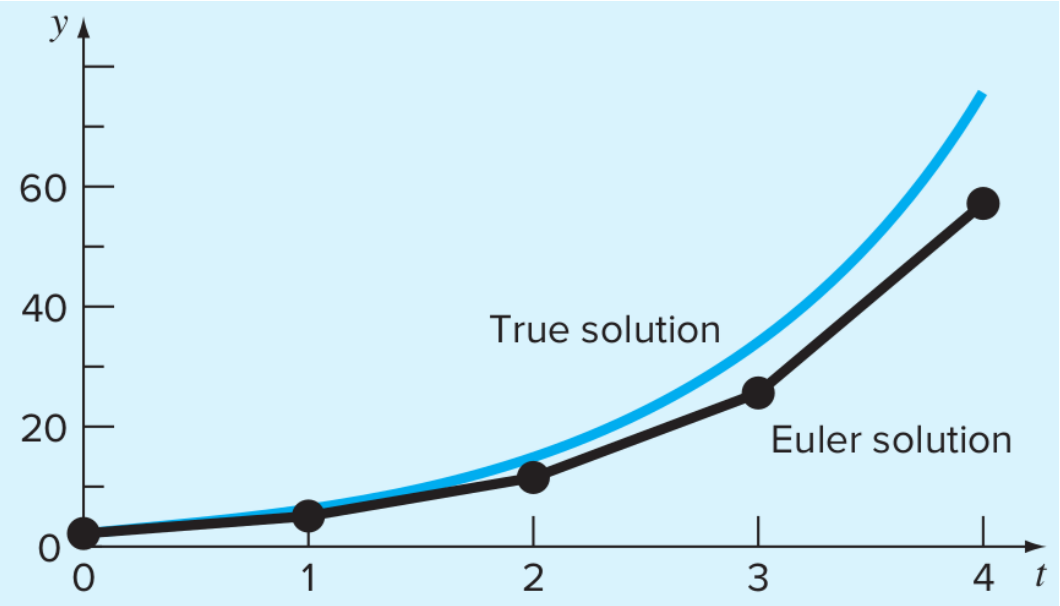
Error after first time step

$$\varepsilon_t = \left|\frac{6.19463 - 5}{6.19463}\right| \times 100\% = 19.28\%$$

- Notice how the error gets compounded meaning that the approximation diverges from the true solution.

- We call this **error propagation** since the initial error can be thought of as being transmitted through the following computations.

- One way of reducing the error is by taking a smaller step size.



Euler's Method Error

- Since Euler's method is a linear approximation, it corresponds with the first order Taylor expansion of *y* and we call it a **first order method**.

- We already know that the error from the Taylor expansion truncated after the first derivative is given by:

$$E_t = \frac{f'(t_i, y_i)}{2!}h^2 + \cdots + O(h^{n+1}) \qquad \left(\text{since } \frac{dy}{dt} = f(t, y)\right)$$

- For small step sizes we neglect the higher order terms and give the approximate error as,

$$E_a = \frac{f'(t_i, y_i)}{2!}h^2 = O(h^2)$$

for a single step.

- Since the error at each step is $O(h^2)$ we assume that the total (global) error is approximately n times the error at each step.

- Since $n = (t_n - t_0)/h$ we have,

$$E_{a,\text{global}} \approx n \frac{f'(t_i, y_i)}{2!} h^2 = \frac{t_n - t_0}{h} \frac{f'(t_i, y_i)}{2!} h^2 = O(h)$$

- The above result can be proved formally but requires the use of some other theorems that we have not covered.

- We have in general that an ***n*th order method** corresponds with a **local error of $O(h^{n+1})$** and a **global error of $O(h^n)$**.

Stability

- We call a solution to an ODE using a numerical method **unstable** if the numerical solution grows exponentially when the true solution is bounded.

- The stability varies depending on the equation to be solved, the numerical method itself, and the step size.

EXAMPLE 2 Examine the stability of the following ODE (with constant, *a*) by comparing with the exact solution.

$$\frac{dy}{dt} = -ay \qquad y(0) = y_0$$

Exact solution

$$y = y_0 e^{-at}$$

Euler's Forward Method

$$y_{i+1} = y_i + \frac{dy_i}{dt}h = y_i - ay_i h = y_i\,(1 - ah)$$

- From the exact solution we know that y should start with a value y_0 then tend to 0 as $t \to \infty$.
- On the other hand our Euler method gave: $y_{i+1} = y_i (1 - ah)$
- This only tends to 0 when $|1 - ah| < 1$. In other words when,

$$0 < h < \frac{2}{a}$$

- So if the step size is too big then the solution will diverge to infinity and is called unstable.
- For this ODE we have **conditional stability**.
- For other ODEs we sometimes find that the solution is unstable for any step size/method. We call these **ill-conditioned** ODEs.

Or go to www.pollev.com/jsands601

When poll is active, respond at **PollEv.com/jsands601**

For $y' = -ay$ and Euler's backward method: $y_{i+1} = y_i + hf(t_{i+1}, y_{i+1})$, for which step size, h is the solution stable?

- $h > 0$
- $h > a$
- $h > \frac{2}{a}$
- $h > \frac{a}{2}$

Start the presentation to see live content. For screen share software, share the entire screen. Get help at **pollev.com/app**

13.2 Heun's Method

- This is an improvement on Euler's method that takes the first derivative at both the starting and end point of a subinterval. The average value of the derivative using these 2 points is calculated and used in the same way as before.

Original Euler's Method (single step)

$$y_{i+1}^{0} = y_i + f(t_i, y_i)h$$

Estimate of 1st derivative at end of subinterval

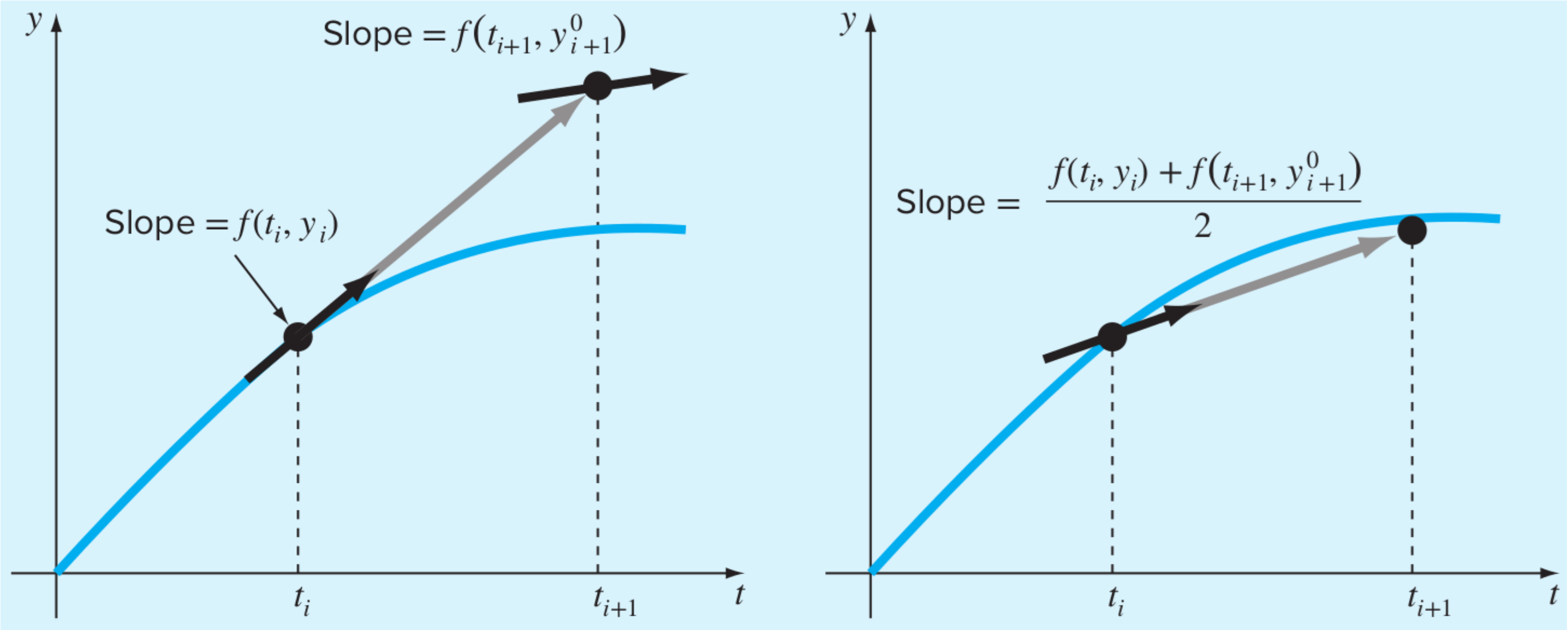
$$\bar{y}' = \frac{f(t_i, y_i) + f(t_{i+1}, y_{i+1}^{0})}{2}$$

Estimate from average slope

Heun's Method

$$y_{i+1} = y_i + \frac{f(t_i, y_i) + f(t_{i+1}, y_{i+1}^{0})}{2}h$$

- Heun's method works as a predictor-corrector pairing.
- The first equation gives a prediction based on an initial linear extrapolation but then uses it to correct the initial prediction in the final linear extrapolation.
- This process can be applied iteratively, in other words the corrected value can then be used as the next predictor etc.



EXAMPLE 3 Use ***Heun's method with iteration*** to solve $y' = 4e^{0.8t} - 0.5y$ from $t = 0$ to 4 with a step size of 1. The initial condition is $y(0) = 2$. Employ a stopping criterion of 0.00001% to terminate the corrector iterations.

Slope at initial point

$$y_0' = 4e^0 - 0.5(2) = 3$$

First prediction at $t = 1$

$$y_1^0 = 2 + 3(1) = 5$$

Correcting slope at $t = 1$

$$y_1' = f(x_1, y_1^0) = 4e^{0.8(1)} - 0.5(5) = 6.402164$$

Average slope between $t = 0$ and $t = 1$

$$\bar{y}' = \frac{3 + 6.402164}{2} = 4.701082$$

Corrected y value at $t = 1$

$$y_1^1 = 2 + 4.701082(1) = 6.701082$$

Approximate error

$$|\varepsilon_a| = \left|\frac{6.701082 - 5}{6.701082}\right| \times 100\% = 25.39\%$$

Next iteration uses previous value as initial predictor

$$y_1^2 = 2 + \frac{\overset{\text{Initial slope}}{3} + \overbrace{4e^{0.8(1)} - 0.5(6.701082)}^{\text{New predicted slope at } t=1}}{2}\, 1 = 6.275811$$

Approximate error

$$|\varepsilon_a| = \left|\frac{6.275811 - 6.701082}{6.275811}\right| \times 100\% = 6.776\%$$

Next iteration

$$y_1^3 = 2 + \frac{3 + 4e^{0.8(1)} - 0.5(6.275811)}{2}\, 1 = 6.382129$$

Approximate error

$$|\varepsilon_a| = \left|\frac{6.382129 - 6.275811}{6.382129}\right| = 1.67\%$$

- Repeating 9 more times results in an approximate solution at t = 1 of 6.36087 which has an approximate error less than 0.00001%.

- Note that the true error, however, is 2.68% so the method converged to an approximate solution but not to the true solution. **This method converges to a value with a finite truncation error for a given step size, not necessarily to the true value**.

				Without Iteration		With Iteration	
t	y_{true}	y_{Euler}	$\lvert\varepsilon_t\rvert$ (%)	y_{Heun}	$\lvert\varepsilon_t\rvert$ (%)	y_{Heun}	$\lvert\varepsilon_t\rvert$ (%)
0	2.00000	2.00000		2.00000		2.00000	
1	6.19463	5.00000	19.28	6.70108	8.18	6.36087	2.68
2	14.84392	11.40216	23.19	16.31978	9.94	15.30224	3.09
3	33.67717	25.51321	24.24	37.19925	10.46	34.74328	3.17
4	75.33896	56.84931	24.54	83.33777	10.62	77.73510	3.18

Heun's Method Error

- The error is the same as the trapezium rule since Heun's method is,

$$y_{i+1} = y_i + \frac{f(t_i) + f(t_{i+1})}{2} h$$

and the trapezium rule is,

$$\int_{t_i}^{t_{i+1}} f(t)\, dt = \frac{f(t_i) + f(t_{i+1})}{2} h$$

with global error $|E_T| \le \frac{nKh^3}{12} = \frac{K(b-a)^3}{12n^2} = O(h^2)$

- The local error is therefore $O(h^3)$.

13.3 Runge-Kutta Methods

- The previous 2 methods are part of a larger class of methods known as Runge-Kutta methods.

- They write the increment function: $y_{i+1} = y_i + \phi h$

where, $$\phi = a_1 k_1 + a_2 k_2 + \cdots + a_n k_n$$

$$
\begin{aligned}
k_1 &= f(t_i, y_i) \\
k_2 &= f(t_i + p_1 h, y_i + q_{11} k_1 h) \\
k_3 &= f(t_i + p_2 h, y_i + q_{21} k_1 h + q_{22} k_2 h) \\
&\vdots \\
k_n &= f(t_i + p_{n-1} h, y_i + q_{n-1,1} k_1 h + q_{n-1,2} k_2 h + \cdots + q_{n-1,n-1} k_{n-1} h)
\end{aligned}
$$

***p*’s and *q*’s are constants**

***k*’s are recurrence relationships**

1st & 2nd Order Runge-Kutta

- The first order Runge-Kutta with $a_1 = 1$ is simply **Euler's method**.

- **Heun's method** is a 2nd order Runge-Kutta with $a_1 = 1/2$, $a_2 = 1/2$, $p_1 = 1$ and $q_{11} = 1$.

- Other 2nd order methods can be derived by setting the corresponding Runge-Kutta method equal to a 2nd order Taylor series:

$$y_{i+1} = y_i + hf(t_i, y_i) + \frac{h^2}{2} f'(t_i, y_i) + O(h^3)$$

- But by the chain rule we have,

$$f'(t_i, y_i) = \frac{\partial f}{\partial t_i}\frac{\partial t_i}{\partial t_i} + \frac{\partial f}{\partial y_i}\frac{\partial y_i}{\partial t_i} = \frac{\partial f}{\partial t_i} + \frac{\partial f}{\partial y_i} f(t_i, y_i)$$

- Substituting into the Taylor expansion results in,

(*) $$y_{i+1} = y_i + hf(t_i, y_i) + \frac{h^2}{2}\left[\frac{\partial f}{\partial t_i} + \frac{\partial f}{\partial y_i} f(t_i, y_i)\right] + O(h^3)$$

- From the definition, the 2nd order Runge-Kutta is given by,

$$y_{i+1} = y_i + h\left(a_1 f(t_i, y_i) + a_2 f(t_i + p_1 h, y_i + q_{11} k_1 h)\right)$$

- But the multivariable Taylor expansion gives,

$$f(t_i + p_1 h, y_i + q_{11} k_1 h) = f(t_i, y_i) + \frac{\partial f}{\partial t_i} p_1 h + \frac{\partial f}{\partial y_i} q_{11} h f(t_i, y_i) + O(h^2)$$

- Substituting this into the 2nd order Runge-Kutta yields,

(**) $$\begin{aligned} y_{i+1} = y_i + ha_1 f(t_i, y_i) \quad \cdots \\ \cdots + ha_2\left[f(t_i, y_i) + \frac{\partial f}{\partial t_i} p_1 h + \frac{\partial f}{\partial y_i} q_{11} h f(t_i, y_i) + O(h^2)\right] \end{aligned}$$

- Comparing **(*)** with **(**)** we see that they are the same when,

$$a_1 + a_2 = 1 \qquad a_2 p_1 = 1/2 \qquad a_2 q_{11} = 1/2$$

- Treating a_2 as a free variable (since we have 3 equations with 4 unknowns) we can write these constants as,

$$a_1 = 1 - a_2$$
$$p_1 = q_{11} = \frac{1}{2a_2}$$

- There are an infinite number of constants that we can choose which give the same results for ODEs whose solutions are constant, linear or quadratic equation.

- For other cases, choosing different constants will mean different results.

Other Common 2nd Order Methods

- Assuming $a_2 = 1$ gives **The Midpoint Method**:

$$y_{i+1} = y_i + k_2 h$$
$$k_1 = f(t_i, y_i)$$
$$k_2 = f(t_i + h/2, y_i + k_1 h/2)$$

- Assuming $a_2 = 3/4$ gives **Ralston's Method**:

$$y_{i+1} = y_i + \left(\frac{1}{4}k_1 + \frac{3}{4}k_2\right)h$$
$$k_1 = f(t_i, y_i)$$
$$k_2 = f\left(t_i + \frac{2}{3}h, y_i + \frac{2}{3}k_1 h\right)$$

13.4 4th Order Runge-Kutta

- This is the most commonly used version and can also have infinite solutions for the constants.

- The version given below is the most standard one that has gained popularity:

$$y_{i+1} = y_i + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)h, \quad k_1 = f(t_i, y_i)$$

$$k_2 = f\left(t_i + \frac{1}{2}h, y_i + \frac{1}{2}k_1 h\right)$$

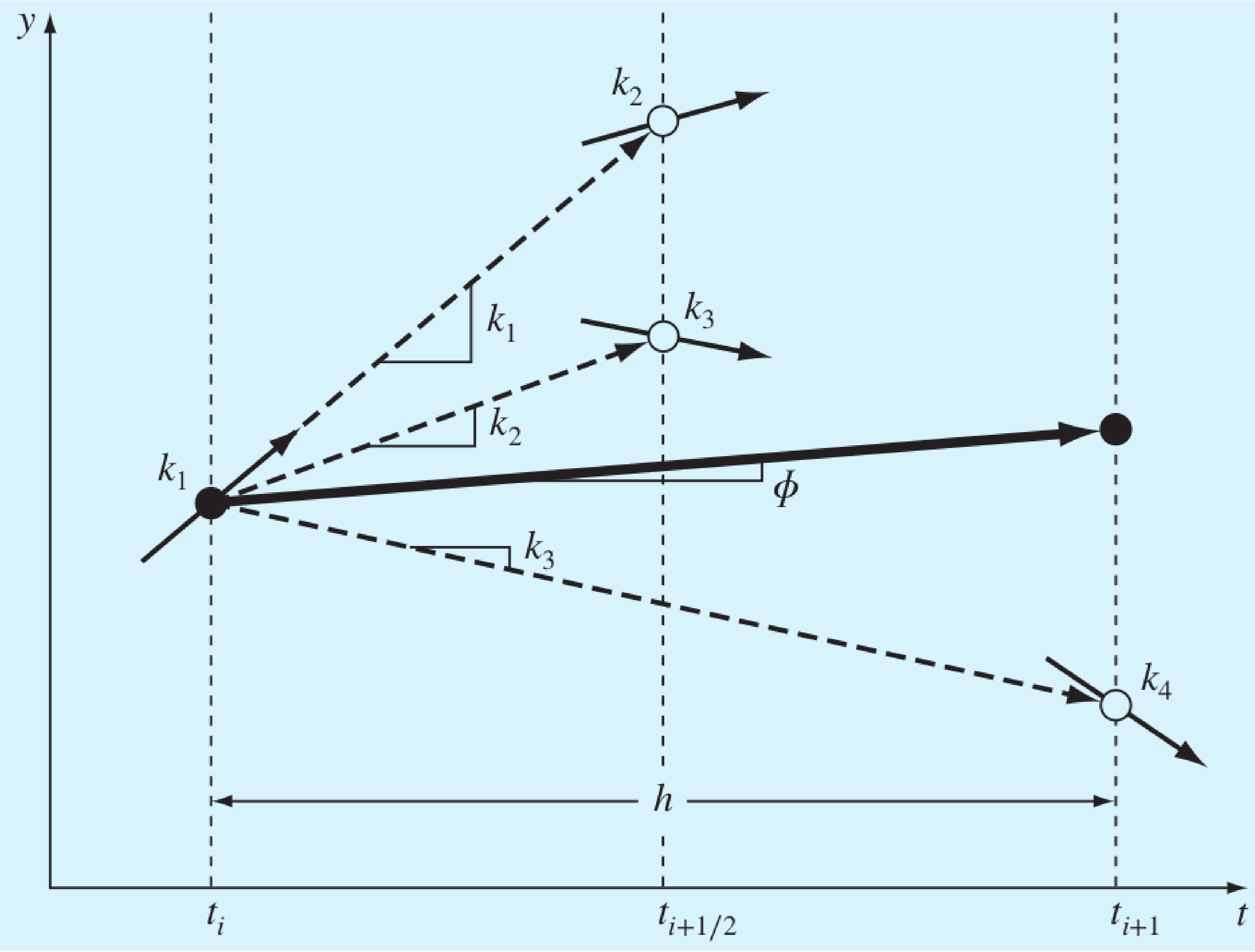
All *k* values are slopes

k_2 and k_3 slopes computed at the midpoint of the interval

$$k_3 = f\left(t_i + \frac{1}{2}h, y_i + \frac{1}{2}k_2 h\right)$$

k_4 is the slope computed at the end of the interval

$$k_4 = f(t_i + h, y_i + k_3 h)$$


y
k2
k1
k3
k2
k1
ϕ
k3
k4
h
ti
ti+1/2
ti+1
t

EXAMPLE 4 Use the 4th order Runge-Kutta method to solve $y' = 4e^{0.8t} - 0.5y$ from $t = 0$ to 1 with a step size of 1. The initial condition is $y(0) = 2$.

Compute k_1

$$k_1 = f(0, 2) = 4e^{0.8(0)} - 0.5(2) = 3$$

Get slope at the midpoint and use it in the k_2 formula

$$y(0.5) = 2 + 3(0.5) = 3.5$$

$$k_2 = f(0.5, 3.5) = 4e^{0.8(0.5)} - 0.5(3.5) = 4.217299$$

Similar process to get k_3

$$y(0.5) = 2 + 4.217299(0.5) = 4.108649$$

$$k_3 = f(0.5, 4.108649) = 4e^{0.8(0.5)} - 0.5(4.108649) = 3.912974$$

k_4 from end of the interval

$$y(1.0) = 2 + 3.912974(1.0) = 5.912974$$

$$k_4 = f(1.0, 5.912974) = 4e^{0.8(1.0)} - 0.5(5.912974) = 5.945677$$

Calculate increment function value

$$\phi = \frac{1}{6}[3 + 2(4.217299) + 2(3.912974) + 5.945677] = 4.201037$$

Use values in RK formula

$$y(1.0) = 2 + 4.201037(1.0) = 6.201037$$

Exact solution is 6.194631 $\longrightarrow$ $\varepsilon_t = 0.103\%$

- Note that all 2nd order methods have local error $O(h^3)$ and global error $O(h^2)$.
- All 4th order methods have local error $O(h^5)$ and global error $O(h^4)$.
- In general an nth order method has local error $O(h^{n+1})$ and global error $O(h^n)$.

13.5 Systems of ODEs

- Given a system of ODEs,

$$\frac{dy_1}{dt} = f_1(t, y_1, y_2, \ldots, y_n)$$
$$\frac{dy_2}{dt} = f_2(t, y_1, y_2, \ldots, y_n)$$
$$\vdots$$
$$\frac{dy_n}{dt} = f_n(t, y_1, y_2, \ldots, y_n)$$

  we simply take a single time step for each equation using any of the single step methods previously discussed.

- For an *n*-dimensional system we require *n* initial conditions.

EXAMPLE 5 Solve the following 2 dimensional system of 1st order ODEs with initial conditions $x(0) = v(0) = 0$ for $0 \le t \le 10$ using Euler's method with a step size of 2. The system represents a bungee jumper with mass 68.1 kg and drag coefficient 0.25 kg/m.

$$\frac{dx}{dt} = \upsilon$$

$$\frac{d\upsilon}{dt} = g - \frac{c_d}{m}\upsilon^2$$

Slopes at initial point (at $t = 0$)

$$\frac{dx}{dt} = 0$$

$$\frac{d\upsilon}{dt} = 9.81 - \frac{0.25}{68.1}(0)^2 = 9.81$$

Euler method first step (at $t = 2$)

$$x = 0 + 0(2) = 0$$

$$\upsilon = 0 + 9.81(2) = 19.62$$

Slopes at second time step (at *t* = 2)

$$\frac{dx}{dt} = 19.62$$

$$\frac{dx}{dt} = 9.81 - \frac{0.25}{68.1} \times 19.62^2 = 8.3968$$

Euler method second step (at *t* = 4)

$$x = 0 + 19.62(2) = 39.24$$

$$v = 19.62 + 8.3968(2) = 36.41368$$

t	x_{true}	v_{true}	x_{Euler}	v_{Euler}	$\varepsilon_t\,(x)$	$\varepsilon_t\,(v)$
0	0	0	0	0		
2	19.1663	18.7292	0	19.6200	100.00%	4.76%
4	71.9304	33.1118	39.2400	36.4137	45.45%	9.97%
6	147.9462	42.0762	112.0674	46.2983	24.25%	10.03%
8	237.5104	46.9575	204.6640	50.1802	13.83%	6.86%
10	334.1782	49.4214	305.0244	51.3123	8.72%	3.83%

EXAMPLE 6 Repeat **Example 5** using the 4th order Runge-Kutta method (step size 2).

$$\frac{dx}{dt} = f_1(t, x, \upsilon) = \upsilon$$

$$\frac{d\upsilon}{dt} = f_2(t, x, \upsilon) = g - \frac{c_d}{m}\upsilon^2$$

k_1 values at t = 0

$$k_{1,1} = f_1(0, 0, 0) = 0$$

$$k_{1,2} = f_2(0, 0, 0) = 9.81 - \frac{0.25}{68.1}(0)^2 = 9.81$$

Function value estimates at t = 1 using k_1

$$x(1) = x(0) + k_{1,1}\frac{h}{2} = 0 + 0\frac{2}{2} = 0$$

$$\upsilon(1) = \upsilon(0) + k_{1,2}\frac{h}{2} = 0 + 9.81\frac{2}{2} = 9.81$$

k_2 values at t = 1

$$k_{2,1} = f_1(1, 0, 9.81) = 9.8100$$

$$k_{2,2} = f_2(1, 0, 9.81) = 9.4567$$

Function value estimates at *t* = 1 using k_2

$$x(1) = x(0) + k_{2,1}\frac{h}{2} = 0 + 9.8100\frac{2}{2} = 9.8100$$

$$\upsilon(1) = \upsilon(0) + k_{2,2}\frac{h}{2} = 0 + 9.4567\frac{2}{2} = 9.4567$$

k_3 values at *t* = 1

$$k_{3,1} = f_1(1, 9.8100, 9.4567) = 9.4567$$

$$k_{3,2} = f_2(1, 9.8100, 9.4567) = 9.4817$$

Function value estimates at *t* = 2 using k_3

$$x(2) = x(0) + k_{3,1}h = 0 + 9.4567(2) = 18.9134$$

$$\upsilon(2) = \upsilon(0) + k_{3,2}h = 0 + 9.4817(2) = 18.9634$$

k_4 values at *t* = 2

$$k_{4,1} = f_1(2, 18.9134, 18.9634) = 18.9634$$

$$k_{4,2} = f_2(2, 18.9134, 18.9634) = 8.4898$$

RK predicted values at *t* = 2

$$x(2) = 0 + \frac{1}{6}[0 + 2(9.8100 + 9.4567) + 18.9634]\,2 = 19.1656$$

$$\upsilon(2) = 0 + \frac{1}{6}[9.8100 + 2(9.4567 + 9.4817) + 8.4898]\,2 = 18.7256$$

t	x_{true}	v_{true}	x_{RK4}	v_{RK4}	$\varepsilon_t\ (x)$	$\varepsilon_t\ (v)$
0	0	0	0	0		
2	19.1663	18.7292	19.1656	18.7256	0.004%	0.019%
4	71.9304	33.1118	71.9311	33.0995	0.001%	0.037%
6	147.9462	42.0762	147.9521	42.0547	0.004%	0.051%
8	237.5104	46.9575	237.5104	46.9345	0.000%	0.049%
10	334.1782	49.4214	334.1626	49.4027	0.005%	0.038%

- Notice the results are much more accurate for the same step size compared with Euler’s method.
- Once again, accuracy can further be increased by taking smaller step sizes.

Converting Higher Order ODEs into a System

- We can convert higher order ODEs into a system of 1st order ODEs to solve using the methods presented.

- We make the substitution $x_{i+1} = y^{(i)}$.

EXAMPLE 7 Convert the following 3rd order ODE into a system of 3 ODEs then solve using 4th order Runge-Kutta for $0 < t < 4$ using a step size of 1.

$$y''' - 2y'' - 5y' + 6y = 0 \qquad y(0) = 0,\ y'(0) = 1,\ y''(0) = 2$$

Make substitution

$$x_1 = y \qquad x_2 = y' \qquad x_3 = y''$$

$$x_1' = x_2 = f_1(t, x_1, x_2, x_3)$$
$$x_2' = x_3 = f_2(t, x_1, x_2, x_3)$$
$$x_3' = 2x_3 + 5x_2 - 6x_1 = f_3(t, x_1, x_2, x_3)$$

$$x_1(0) = 0,\ x_2(0) = 1,\ x_3(0) = 2$$

k_1 values

$$\left.\begin{aligned} k_{1,1} &= f_1(0,0,1,2) = 1 \\ k_{1,2} &= f_2(0,0,1,2) = 2 \\ k_{1,3} &= f_3\,(0,0,1,2) = 9 \end{aligned}\right\}$$

$x(0.5)$ values

$$x_1(0.5) = x_1(0) + k_{1,1} \times 0.5 = 0.5$$
$$x_2(0.5) = x_2(0) + k_{1,2} \times 0.5 = 2$$
$$x_3(0.5) = x_3(0) + k_{1,2} \times 0.5 = 6.5$$

k_2 values

$$\left.\begin{aligned} k_{2,1} &= f_1(0.5,0.5,2,6.5) = 2 \\ k_{2,2} &= f_2(0.5,0.5,2,6.5) = 6.5 \\ k_{2,3} &= f_3\,(0.5,0.5,2,6.5) = 20 \end{aligned}\right\}$$

$x(0.5)$ values

$$x_1(0.5) = x_1(0) + k_{2,1} \times 0.5 = 1$$
$$x_2(0.5) = x_2(0) + k_{2,2} \times 0.5 = 4.25$$
$$x_3(0.5) = x_3(0) + k_{2,2} \times 0.5 = 12$$

k_3 values

$$\left.\begin{aligned} k_{3,1} &= f_1(0.5,1,4.25,12) = 4.25 \\ k_{3,2} &= f_2(0.5,1,4.25,12) = 12 \\ k_{3,3} &= f_3\,(0.5,1,4.25,12) = 39.25 \end{aligned}\right\}$$

$x(1)$ values

$$x_1(1) = x_1(0) + k_{3,1} \times 1 = 4.25$$
$$x_2(1) = x_2(0) + k_{3,2} \times 1 = 13$$
$$x_3(1) = x_3(0) + k_{3,2} \times 1 = 41.25$$

k_4 values

$$k_{4,1} = f_1(1, 4.25, 13, 41.25) = 13$$

$$k_{4,2} = f_2(1, 4.25, 13, 41.25) = 41.25$$

$$k_{4,3} = f_3\ (1, 4.25, 13, 41.25) = 122$$

$x(1)$ values from RK4 formula

$$x_1(1) = x_1(0) + \frac{1}{6}(k_{1,1} + 2k_{2,1} + 2k_{3,1} + k_{4,1}) \times 1 = 4.4167$$

$$x_2(1) = x_2(0) + \frac{1}{6}(k_{1,2} + 2k_{2,2} + 2k_{3,2} + k_{4,2}) \times 1 = 14.375$$

$$x_3(1) = x_3(0) + \frac{1}{6}(k_{1,3} + 2k_{2,3} + 2k_{3,3} + k_{4,3}) \times 1 = 43.5833$$

t	$y = x_1$	True y	ε_t
0	0	0	0%
1	4.4167	5.5907	20.1%
2	79.2049	119.7996	33.9%
3	1313.9249	2427.5779	45.9%
4	21560.8496	48817.3378	55.8%

- For this step size the solution is not accurate over the interval specified. A step size of 0.1 gives the following.

t	$y = x_1$	True y	ε_t
0	0	0	0%
1	5.5536	5.5907	0.66%
2	119.7564	119.7996	0.04%
3	2426.4264	2427.5779	0.05%
4	48786.5181	48817.3378	0.06%

When poll is active, respond at **PollEv.com/jsands601**

Which system of ODEs represents the equation:

$$3y^{(4)} + 2y''' + 4y'' - y' + 8y = \sin(t)?$$

1. [illegible]
2. [illegible]
3. [illegible]
4. [illegible]
5. $x_1 = y$, $x_2 = y'$, $x_3 = y''$, $x_4 = y'''$
6. [illegible]
7. [illegible]
8. [illegible]
9. [illegible]

To 0

Start the presentation to see live content. For screen share software, share the entire screen. Get help at **pollev.com/app**