Differential Equations

Analytical solutions

- Find a function y that satisfies the differential equation:

$$\frac{dy}{dx} = \frac{x^2 + 1}{2y}$$

  - Separation of variables

$$2y\, dy = (x^2 + 1)\, dx$$

$$\int 2y\, dy = \int (x^2 + 1)\, dx$$

$$y^2 + C_1 = \frac{1}{3}x^3 + x + C_2$$

$$y^2 = \frac{1}{3}x^3 + x + C$$

Boundary Conditions

- What is $y(x)$ if we know that $y = 3$ at $x = 0$?

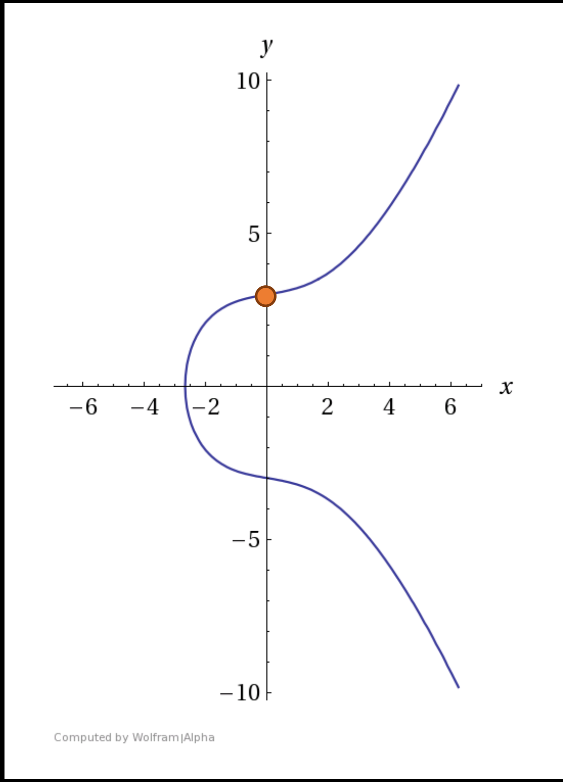
$$y^2 = \frac{1}{3}x^3 + x + C$$

$$3^2 = \frac{1}{3}0^3 + 0 + C$$

$$C = 9$$

$$y(x) = \sqrt{\frac{1}{3}x^3 + x + 9}$$

note that $y(x)$ is always positive



- This is generally known as an *initial value problem*

Analytical solutions

- Find a function that satisfies the following differential equation for the initial value $y(0) = \frac{\pi}{2}$

$$x\,dx + \sec x \sin y\;\,dy = 0$$

- Remember integration by parts:

$$\int f(x)g'(x)\,dx = f(x)g(x) - \int g(x)f'(x)\,dx$$

Analytical solutions

$$x\,dx + \sec x \sin y \; dy = 0$$

$$\sec x \sin y \; dy = -x\,dx$$

$$\sin y \; dy = -x \cos x \; dx$$

$$\int \sin y \; dy = -\int x \cos x \; dx$$

$$\cos y = -\left[x \sin x - \int \sin x \; dx\right]$$

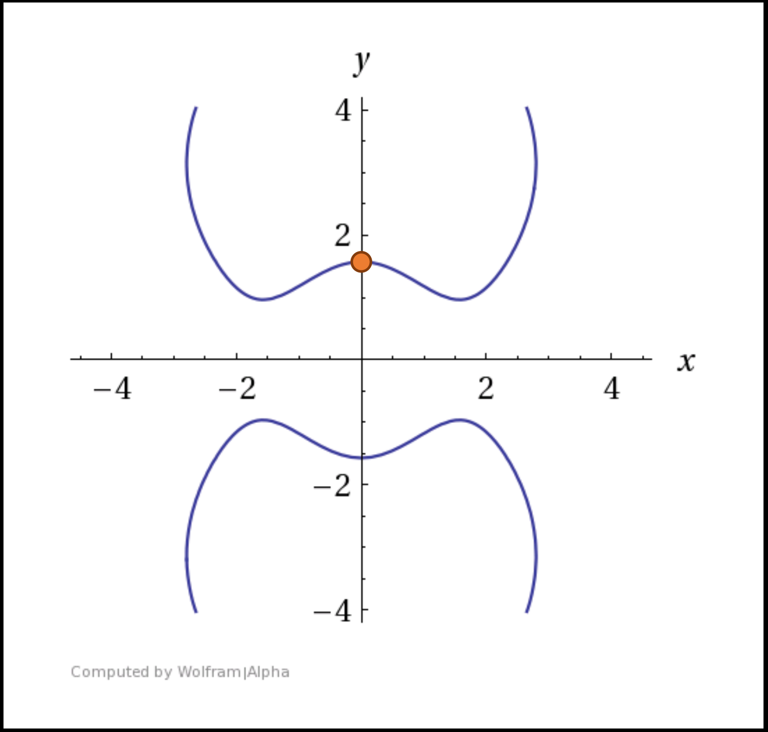
$$\cos y = x \sin x + \cos x + C$$

Analytical solutions

- calculate C given $\mathrm{y}(0) = \frac{\pi}{2}$:

$$\cos\frac{\pi}{2} = (0)\sin 0 + \cos 0 + C$$
$$C = -1$$

$\cos y = x\sin x + \cos x - 1$



Physical systems – exponential

- You own x_0 pounds of tribbles
- Assume that your $x = x_0$ pounds of pet tribbles multiply at a rate of α per day:

$$\frac{dx}{dt} = \alpha x$$

$\xrightarrow{?}$

Physical systems – exponential

- Find an equation $x(t)$ that satisfies $\frac{dx}{dt} = \alpha x$

$$\frac{dx}{dt} = \alpha x$$

$$\frac{1}{x}dx = \alpha\, dt$$

$$\int \frac{1}{x}dx = \int \alpha\, dt$$

$$\ln x + C_1 = \alpha t + C_2$$

$$\ln x = \alpha t + C$$

- Solve for C given the initial condition $x(0) = x_0$

$$\ln x_0 = \alpha(0) + C$$

$$C = \ln x_0$$

Physical systems – exponential

- Plug in the value of C to get $x(t)$

$$\ln x = \alpha t + \ln x_0$$

$$\ln x - \ln x_0 = \ln \frac{x}{x_0} = \alpha t$$

$$\frac{x}{x_0} = e^{\alpha t}$$

$$x(t) = x_0 e^{\alpha t}$$

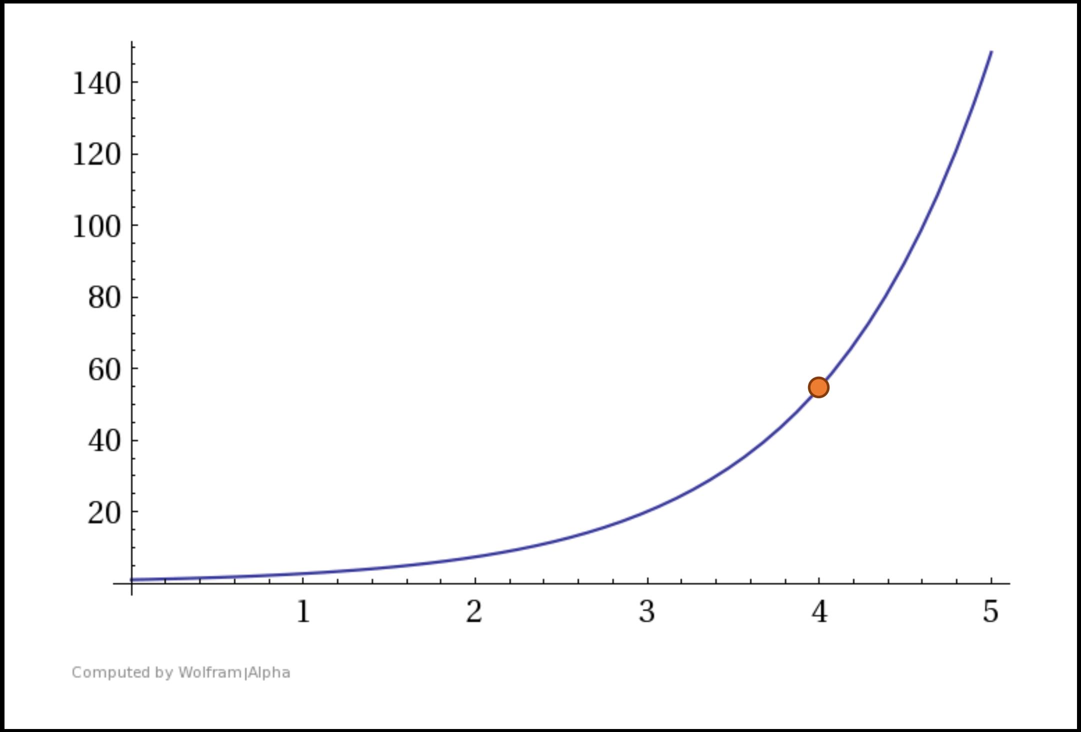
Physical systems – exponential

- Assuming we purchase $x_0 = 1$ pound of baby tribbles and they double every day: $\alpha = 1$

How long before a crew of 60 people can each have a pound of tribble?

$x(t) = e^t = 60$

$t = \ln 60 \approx 4.09$ days



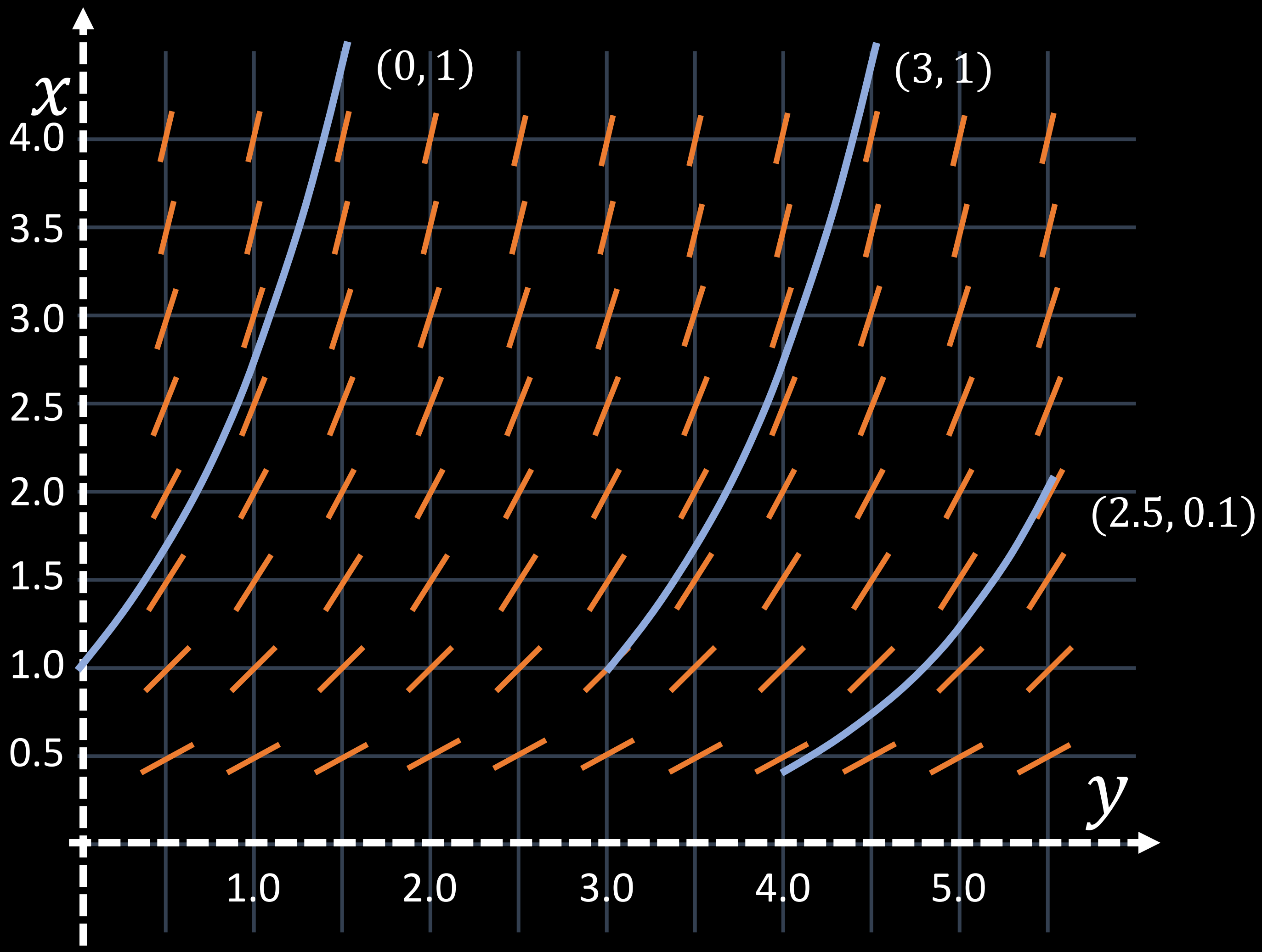
Analytical solutions

- Most differential equations cannot be solved analytically
- Solving differential equations that model physical systems are a primary application of numerical methods

- ***Slope fields*** – if we have a first-order differential equation, we can calculate the slope at each point (x, y) to understand the behavior of the function
- ***Explicit methods*** – if we know an initial value, we can approximate further values
- Look at numerical approaches for $x(t) = x_0 e^{\alpha t}$ as an example

Slope fields

Calculate $\frac{dy}{dx} = y$ for various initial values (x, y)



Explicit methods

- Given $\frac{dx}{dy}$, Start from the initial value (x, y) and increment by some discrete Δx:

$$y_{n+1} = y_n + \Delta x \cdot \frac{dx}{dy}$$

$$x_{n+1} = x_n + \Delta x$$

Euler method

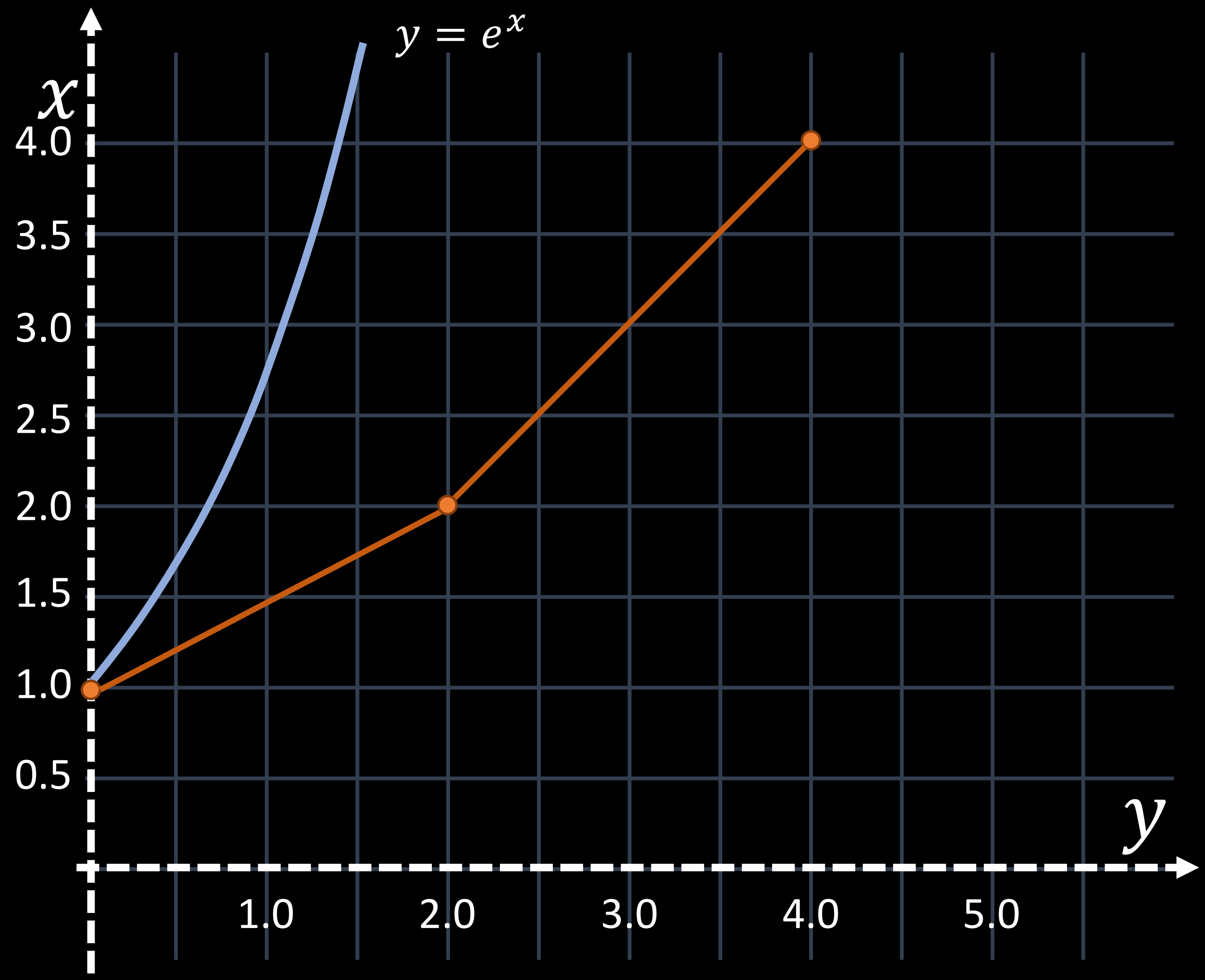
Calculate a function that satisfies $\frac{dy}{dx} = y$ for the initial value $(0, 1)$

$$y_{n+1} = y_n + \Delta x \cdot \frac{dy}{dx}$$

$$x_{n+1} = x_n + \Delta x$$

$$\Delta x = 2$$

$\boldsymbol{x}$ $\boldsymbol{\Delta x = 2}$	$\boldsymbol{y}$	$\boldsymbol{{dx}/{dy}}$
0	1	1
2	3	3
4	9	9
6	27	27



Euler method

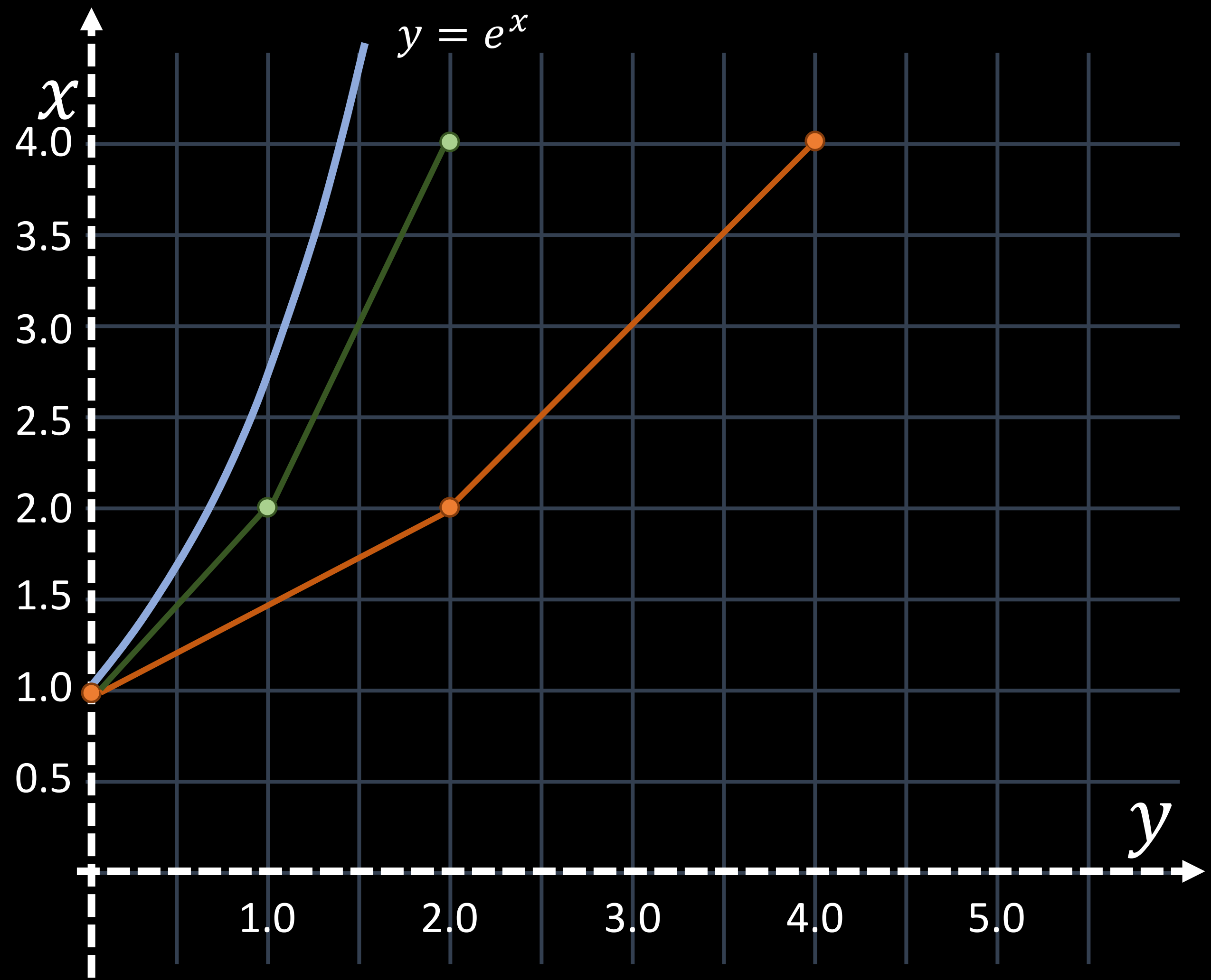
Calculate a function that satisfies $\frac{dy}{dx} = y$ for the initial value $(0, 1)$

$$y_{n+1} = y_n + \Delta x \cdot \frac{dy}{dx}$$

$$x_{n+1} = x_n + \Delta x$$

$$\Delta x = 1$$

x $\Delta x = 1$	y	dx/dy
0	1	1
1	2	2
2	4	4
3	8	



Euler method

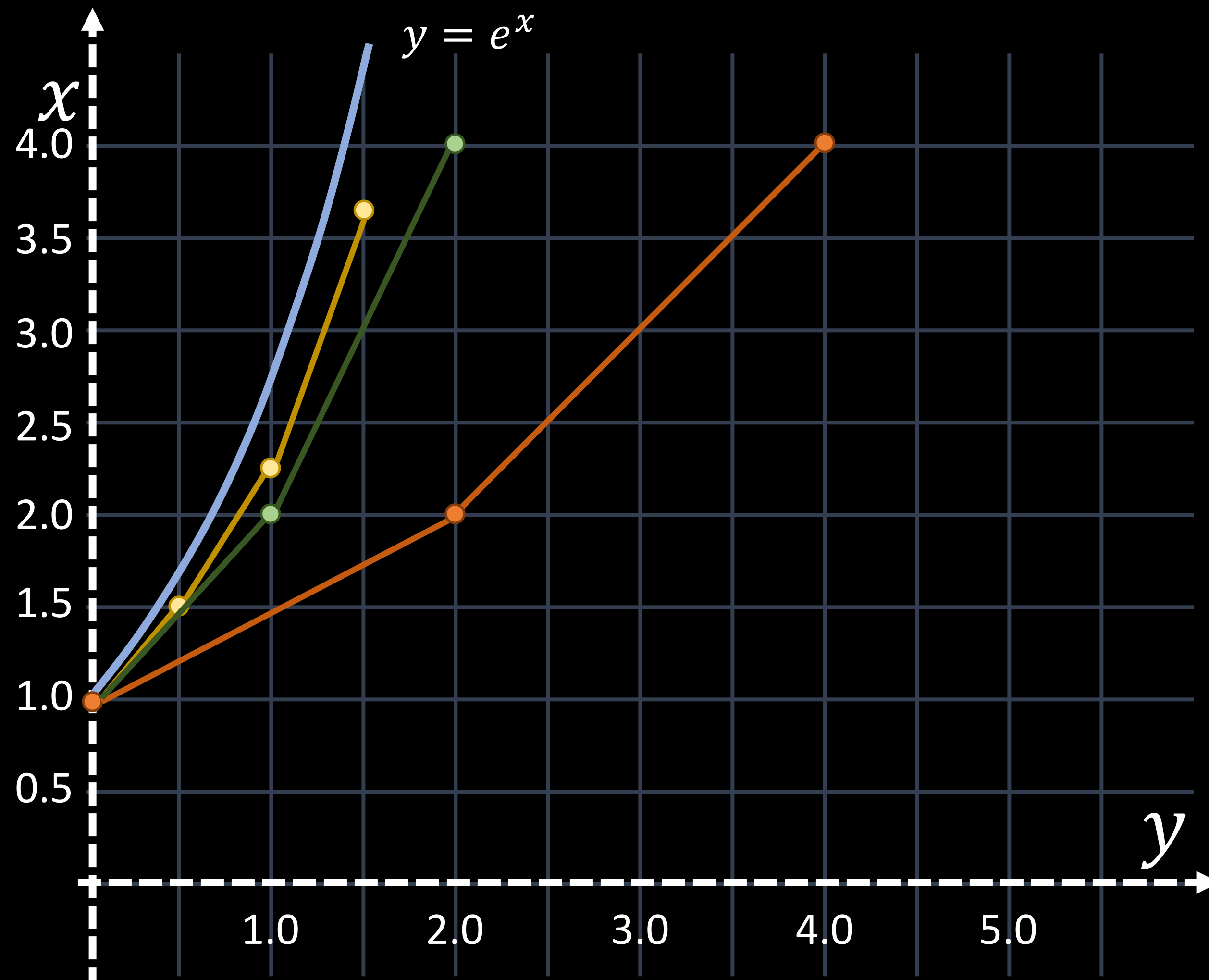
Calculate a function that satisfies $\frac{dy}{dx} = y$ for the initial value $(0, 1)$

$$y_{n+1} = y_n + \Delta x \cdot \frac{dy}{dx}$$

$$x_{n+1} = x_n + \Delta x$$

$$\Delta x = 0.5$$

x $\Delta x = 0.5$	y	$^{dx}/_{dy}$
0	1	1
0.5	1.5	1.5
1	2.25	2.25
1.5	3.36	3.36
2	5.04	



Explicit methods

- Euler method
  - given $\frac{dy}{dx}$, Start from the initial value (x, y) and increment by some discrete Δx:

$$y_{n+1} = y_n + \Delta x \cdot \frac{dy}{dx}$$

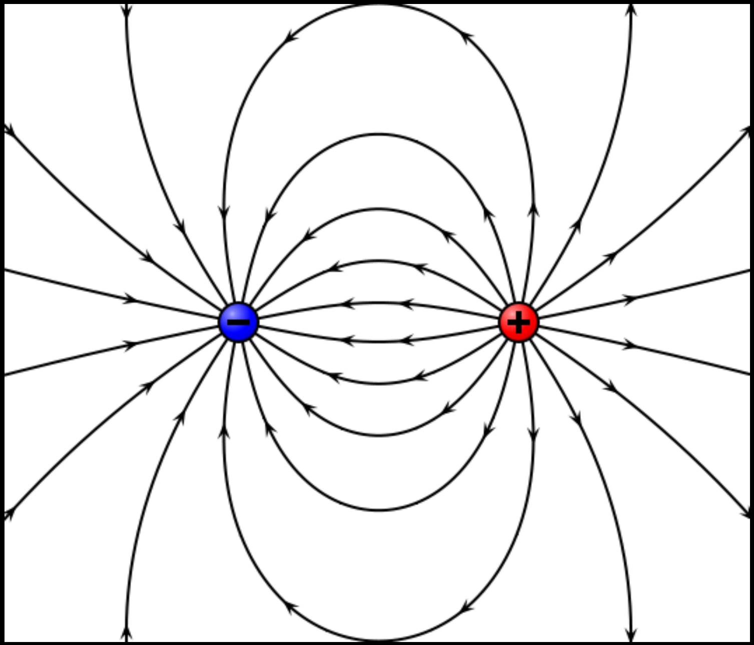
$$x_{n+1} = x_n + \Delta x$$

- What is the order of the error term using the Euler method?
  - Take the Taylor series expansion of y about x

$$y(x + \Delta x) = y(x) + \Delta x \cdot y'(x) + O(\Delta x^2)$$

Physical systems – fields

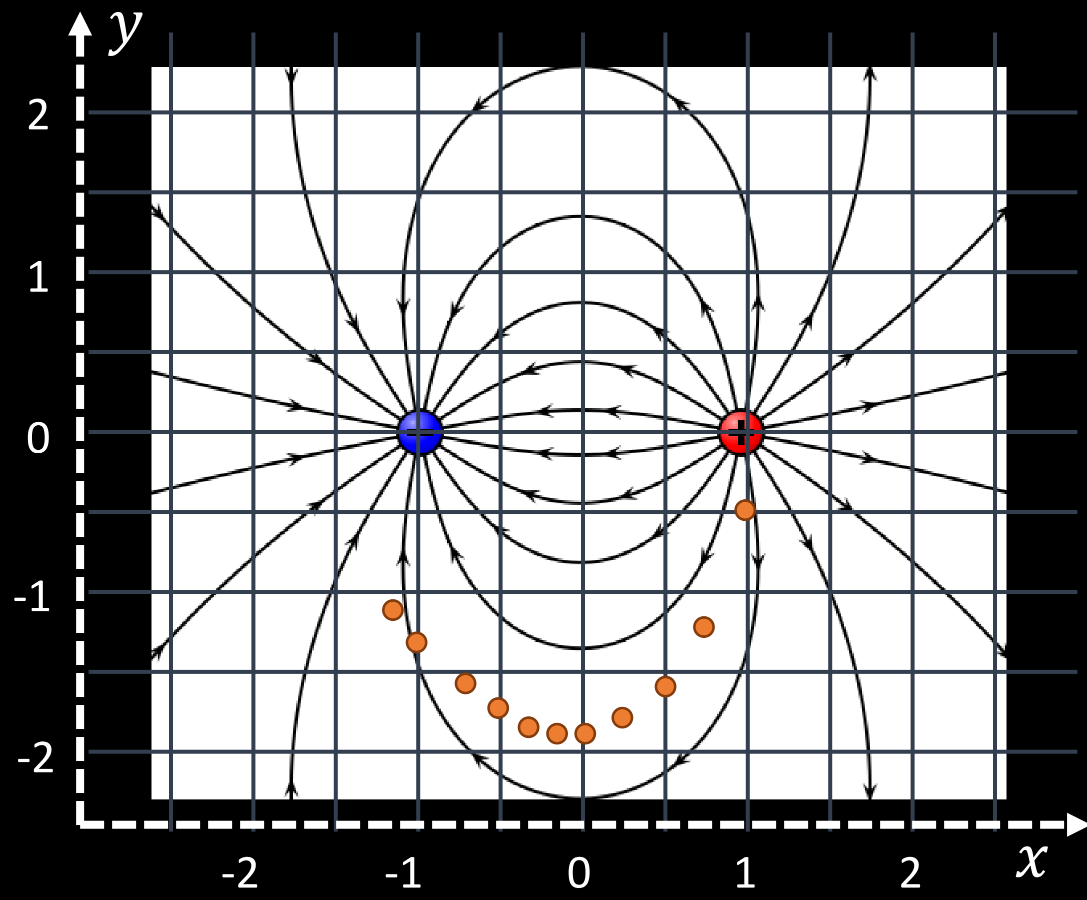
- Assuming a positive mass-less point charge at position $\boldsymbol{p}$
- The particle p is placed into an electric field produced by positive and negative charges at positions $\boldsymbol{e}_+$ and $\boldsymbol{e}_-$
  - The velocity of the particle due to the positive charge is
    $$\boldsymbol{v}_+ = \frac{\boldsymbol{e}_+ - \boldsymbol{p}}{|\boldsymbol{e}_+ - \boldsymbol{p}|^2}$$
  - The velocity of the particle due to the negative charge is
    $$\boldsymbol{v}_- = -\frac{\boldsymbol{e}_- - \boldsymbol{p}}{|\boldsymbol{e}_- - \boldsymbol{p}|^2}$$
  - And the net velocity is
    $$\boldsymbol{v} = \boldsymbol{v}_+ + \boldsymbol{v}_-$$
- Assuming a 2D system:
  $$\frac{dx}{dt} = v_x \text{ and } \frac{dy}{dt} = v_y$$
- The update function is given by:
  $$\boldsymbol{p}_{\boldsymbol{n+1}} = \mathbf{p_n} + \Delta t(\boldsymbol{v})$$



Physical systems – fields

- Using Euler's Method with $\Delta t = 0.5$ and $\boldsymbol{p_0} = (1, -0.5)$

x	y	dx/dt	dy/dt
1	-0.5	-0.47	-1.88
0.76	-1.44	-0.45	-0.4
0.54	-1.54	-0.46	-0.24
0.31	-1.76	-0.47	-0.13
0.08	-1.82	-0.46	-0.03
…	…	…	…
-1.16	-1.23		



Physical systems – Matlab code

```
clc;
clear;

%set the time step
dt = 0.05;

%set the number of time steps
T = 60;

%position of the negative charge
e_n = [-1, 0];

%position of the positive charge
e_p = [1, 0];

%start position of the point charge
p_s = [1, -0.5];

%allocate space for the list of points
p = zeros(T+1, 2);

%initialize the first point to the start position
p(1, :) = p_s;
```

```
%for each time step
for t = 2:T+1
    %calculate the velocity due to positive charge
    v_p = (p(t-1, :) - e_p) / norm(p(t-1,:) - e_p)^2;

    %calculate the velocity due to negative charge
    v_n = -(p(t-1,:) - e_n) / norm(p(t-1,:) - e_n)^2;

    v = (v_p + v_n);

    %calculate the new position of the particle
    p(t,:) = p(t-1,:) + dt * (v);

end

%plot the negative and positive charges
scatter(e_n(1), e_n(2), 'O');
hold on;
scatter(e_p(1), e_p(2), 'O');

%plot the particle path using a color map
scatter(p(:, 1), p(:, 2), [], 1:T+1, 'filled')

ylim([-2 2]);
xlim([-3 3]);
hold off
```

Euler method applications

- Use Matlab to calculate solutions to the following differential equations using the Euler method for varying time steps Δx

$$\frac{dy}{dx} = \sin(y + x^2)$$

$$\frac{dy}{dx} = y^2 - x$$

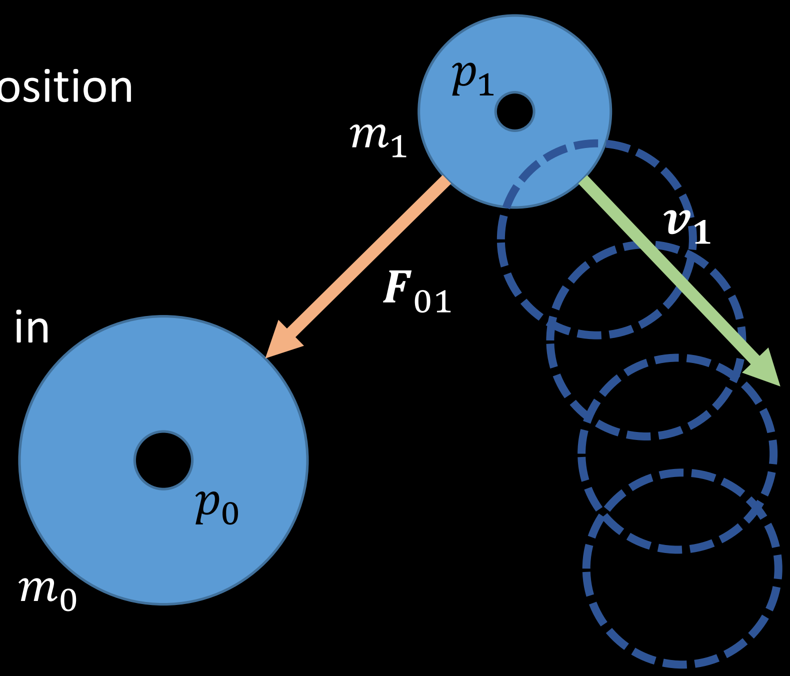
Two-body problem

- Determine the motion of two objects in a physical system
  - $\boldsymbol{p}_0$ and $\boldsymbol{p}_1$ are the positions of both objects
  - m_0 and m_1 are their corresponding masses
  - $\boldsymbol{v}_0$ and $\boldsymbol{v}_1$ are their corresponding velocities
  - $\boldsymbol{F}_{10}$ is the force applied to object 1 by object 0
- This system is governed by a set of differential equations
  - Velocity describes an object's change in position over time:

$$v = \frac{dp}{dt}$$

  - Acceleration describes an object's change in velocity over time:

$$a = \frac{dv}{dt}$$



Two-body problem

- This system is governed by a set of differential equations
  - Velocity describes an object's change in position over time:

$$v = \frac{dp}{dt}$$

  - Acceleration describes an object's change in velocity over time:

$$a = \frac{dv}{dt}$$

  - Newton's second law of motion – the sum of all forces on an object equals the product of the object's acceleration and mass

$$\sum F = ma$$

Two-body problem

- Find an object's position as a function of time $p(t)$
- Using explicit integration:

$$\begin{aligned} p(t_n) &= p(t_{n-1}) + \Delta t \cdot \frac{dp}{dt} \\ &= p(t_{n-1}) + \Delta t \cdot v(t_n) \end{aligned}$$

$$\begin{aligned} v(t_n) &= v(t_{n-1}) + \Delta t \cdot \frac{dv}{dt} \\ &= v(t_{n-1}) + \Delta t \cdot a(t_n) \end{aligned}$$

- The acceleration is given by $F = ma$

$$a(t_n) = \frac{1}{m} \sum F$$

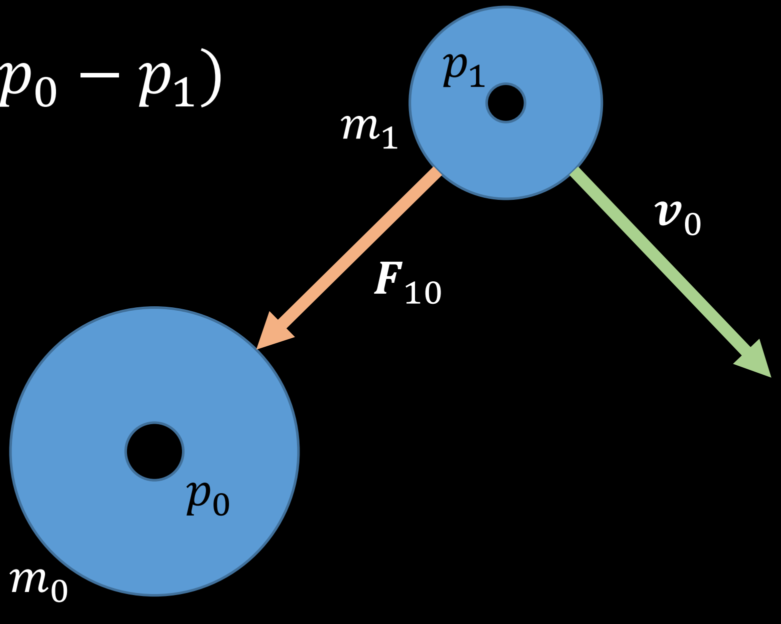
Two-body problem

- Newton's Law of Universal Gravitation

$$F_{ij} = -G\frac{m_i m_j}{\left|p_j - p_i\right|^3}(p_j - p_i)$$

- Since there are only two bodies (and therefore two forces)

$$F_{10} = -G\frac{m_1 m_0}{|p_0 - p_1|^3}(p_0 - p_1)$$



Satellite in orbit

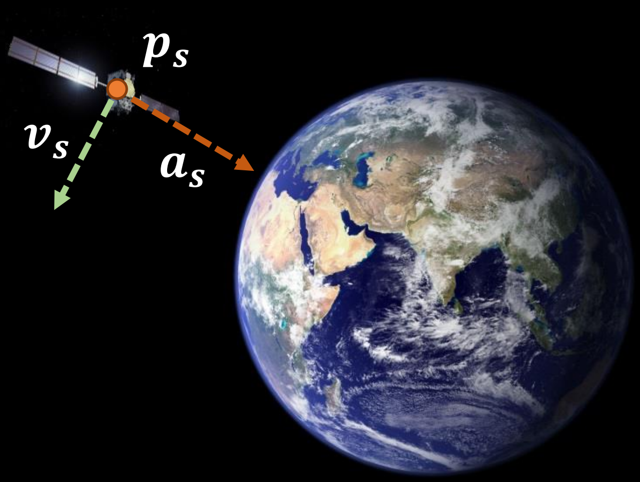
- The satellite has a position p_s, mass m_s, and velocity v_s
- The planet's gravity applies a force given by Newton:

$$F = -G\,\frac{m_s m_e}{|p_s - p_e|^3}\,(p_s - p_e)$$

where p_e and m_e are the position and mass of the planet

- This results in an acceleration dependent only on p_s:

$$\begin{aligned} a(t_n) &= \frac{1}{m}\sum F \\ &= -G\,\frac{m_e}{|p_s - p_e|^3}\,(p_s - p_e) \end{aligned}$$



- Using the Euler update equations:

$$v(t_n) = v(t_{n-1}) + \Delta t \cdot a(t_n)$$
$$p(t_n) = p(t_{n-1}) + \Delta t \cdot v(t_n)$$

Momentum – Matlab code

```
%set the time step
dt = 0.01;

%set the number of time steps
T = 500;

%starting position and velocity of the satellite
p_s = [0, 0];
v = [-3, 0];

%gravitational constant
G = 1;

%position and mass of the planet
p_g = [-5, -5];
m = 100;

%allocate space for the list of positions
p = zeros(T+1, 2);

%initialize the first point to the start position
p(1, :) = p_s;

%for each time step
for t = 2:T+1

    %calculate the force on the satellite

    %direction of the force
    r = p_g - p(t-1, :);

    %distance between satellite and planet
    r_mag = norm(r);

    %calculate the magnitude of the force
    f_mag = G * m/(r_mag^2);

    %calculate the force vector
    f = r/norm(r) * f_mag;

    %apply the force as a change in velocity
    v = v + dt * f;

    %apply the velocity as a change in position
    p(t, :) = p(t-1, :) + dt * v;

end

%display the image
```

Higher order approximations

- Given the differential equation:
$$x' = 1 + x^2 + t^3$$
find a 4th order approximation for x
- Generate a Taylor series expansion
$$x(t+h) \approx x(t) + hx'(t) + \frac{1}{2}h^2x''(t) + \frac{1}{6}h^3x'''(t) + \frac{1}{24}h^4x^{(4)}(t)$$
- Compute additional derivatives:
$$\begin{aligned} x' &= 1 + x^2 + t^3 \\ x'' &= 2xx' + 3t^2 \\ x''' &= 2(xx'' + x'x') + 6t \\ x^{(4)} &= 2xx''' + 6x'x'' + 6 \end{aligned}$$

Higher order approximations

- Generate a Taylor series expansion

$$x(t+h) \approx x(t) + hx'(t) + \frac{1}{2}h^2x''(t) + \frac{1}{6}h^3x'''(t) + \frac{1}{24}h^4x^{(4)}(t)$$

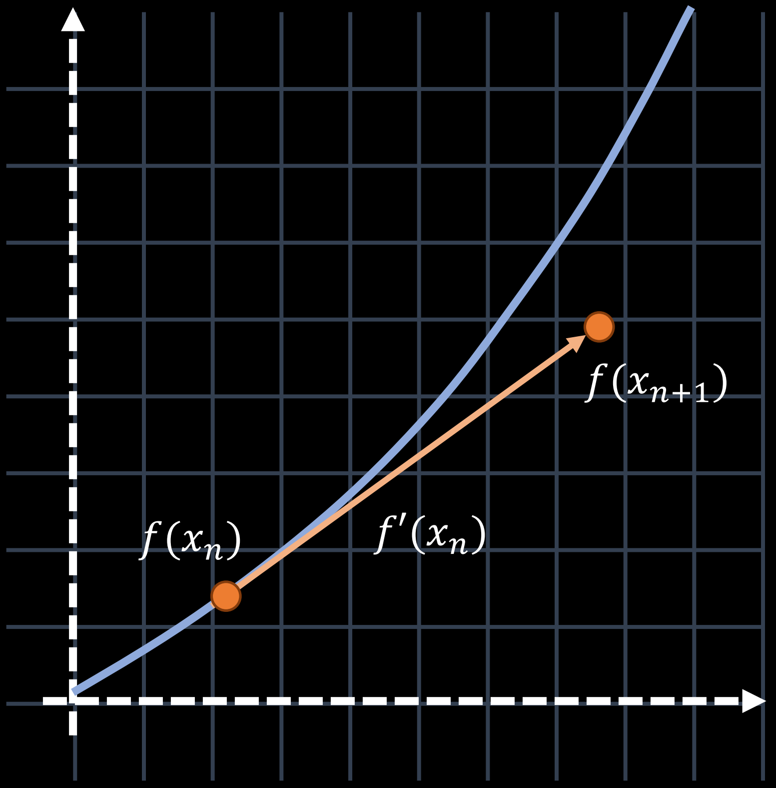
- Starting at some value $x(t_0) = x_0$, iterate using derivatives
  - This concept is an extension of the Euler method, which just relies on the given x'
  - What is the error for this approximation?

    $O(h^5)$ 4th order – error bound is $O(h^{n+1})$ where $n = 4$

- Differentiation is often easier than integration
- Requires an analytical expression

Predictor-Corrector Algorithms

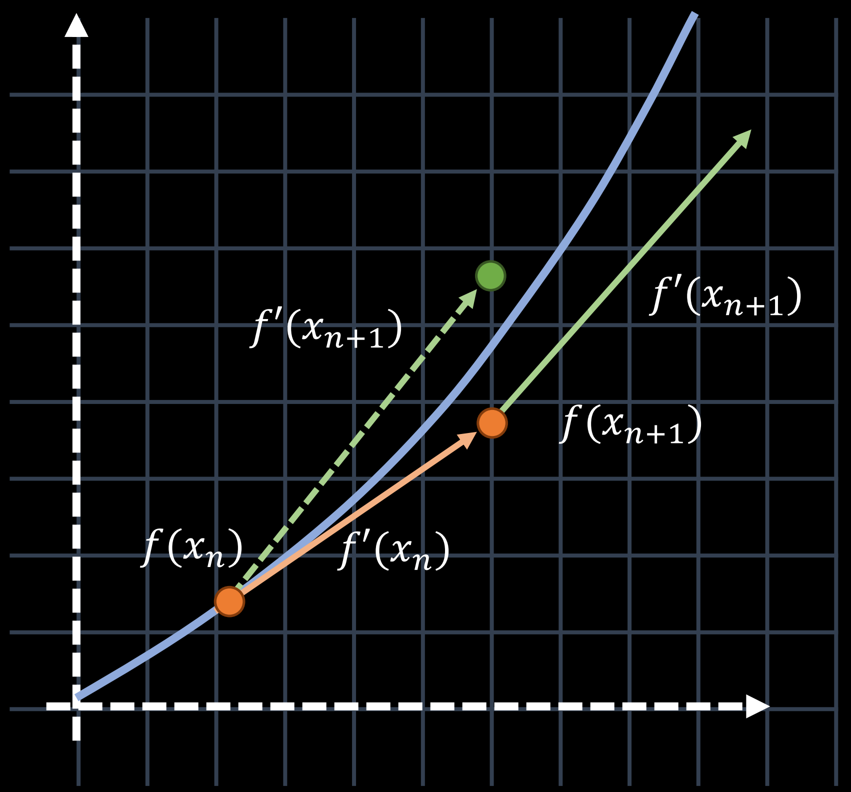
- Error in the Euler method is cumulative
- Also this error is consistent
  - If the function is "concave up"
    then $f(x)$ will be ***under*** estimated
  - If the function is "concave down"
    then $f(x)$ will be ***over*** estimated



Heun's Method

- Take one Euler step
  $f(x_{n+1}) \approx f(x_n) + \Delta x \cdot f'(x_n)$
- This provides a "prediction" for the next value $f(x_{n+1})$
- If the curve is concave up (positive curvature)
  - $f(x_{n+1})$ will be an under-estimate
  - $f'(x_{n+1}) > f'(x_n)$
- A more accurate result can be obtained using the ***average***

$$\widehat{f}'(x_{n+1}) = \frac{f'(x_n) + f'(x_{n+1})}{2}$$



Heun's Method

- Also known as the "Improved Euler Method"
- Update steps given $\frac{dy}{dx} = f(x, y)$:

$$x_{n+1} = x_n + \Delta x$$

$$\hat{y}_{n+1} = y_n + \Delta x \cdot f(x_n, y_n)$$

$$y_{n+1} = y_n + \frac{\Delta x}{2}\left[f(x_n, y_n) + f(x_{n+1}, \hat{y}_{n+1})\right]$$

- This can be thought of as a two-step algorithm:
  - Make a prediction based on $f(x_n, y_n)$
  - Make a correction based on f at the predicted position

Runge-Kutta method

- Taylor series method
  - Requires analytical differentiation – difficult to incorporate into software
- Runge-Kutta methods
  - Named after Carl Runge and Wilhelm Kutta
  - Approximates the higher-order Taylor series method
  - Eliminates the need for analytical differentiation

- Euler’s method is the first-order Runge-Kutta method
- Heun’s method is the second-order Runge-Kutta method
- The most commonly used method for integration in initial value problems is the fourth-order Runge-Kutta method
  - Generally just known as ***the*** Runge-Kutta method

Runge-Kutta method

- Update step:

$$x_{n+1} = x_n + \Delta x$$

$$y_{n+1} = y_n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

where k_1 to k_4 are given by:

$$k_1 = \Delta x \cdot f(x_n, y_n)$$

$$k_2 = \Delta x \cdot f\left(x_n + \tfrac{\Delta x}{2}, y_n + \tfrac{k_1}{2}\right)$$

$$k_3 = \Delta x \cdot f\left(x_n + \tfrac{\Delta x}{2}, y_n + \tfrac{k_2}{2}\right)$$

$$k_4 = \Delta x \cdot f(x_n + \Delta x, y_n + k_3)$$

Runge-Kutta method

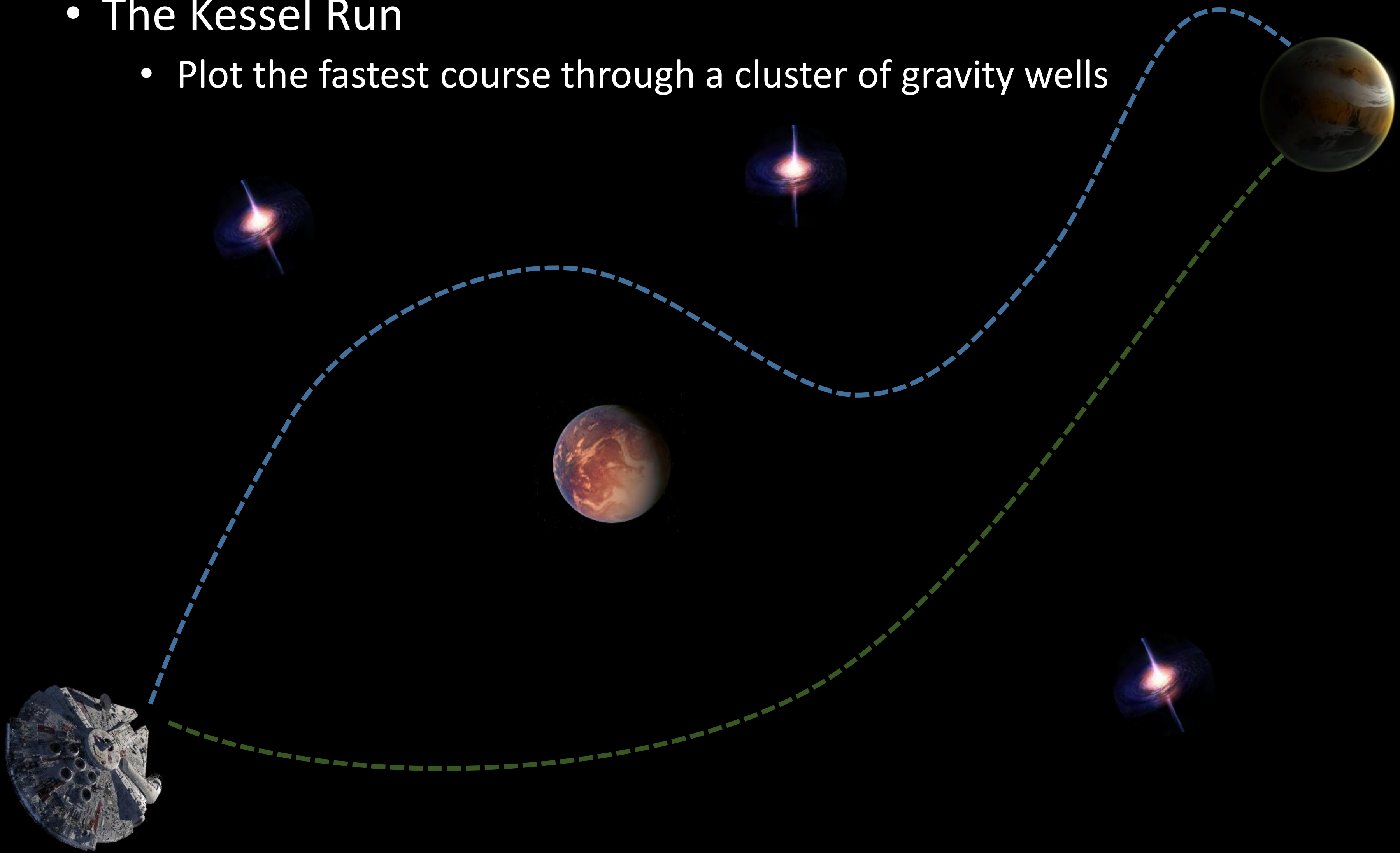
- Computes a weighted average of k_1 to k_4
- k_1 is simply Euler's method:

$$k_1 = \Delta x \cdot f(x_n, y_n)$$

- k_1 and k_4 estimate the slope at the end points
  - These two values effectively implement the steps in Heun's method
- k_2 and k_3 provide estimates of the slope at the mid points in the step Δx

Programming Assignment

- The Kessel Run
  - Plot the fastest course through a cluster of gravity wells



Physics

- You have control over:
    - Initial velocity
        - maximum velocity v_{max}
        - no “active” changes in velocity are allowed for the duration of the simulation – you’re “floating” until the end
    - Initial trajectory
    - Starting position
        - You will be allowed to start at any point on a given line

Start state

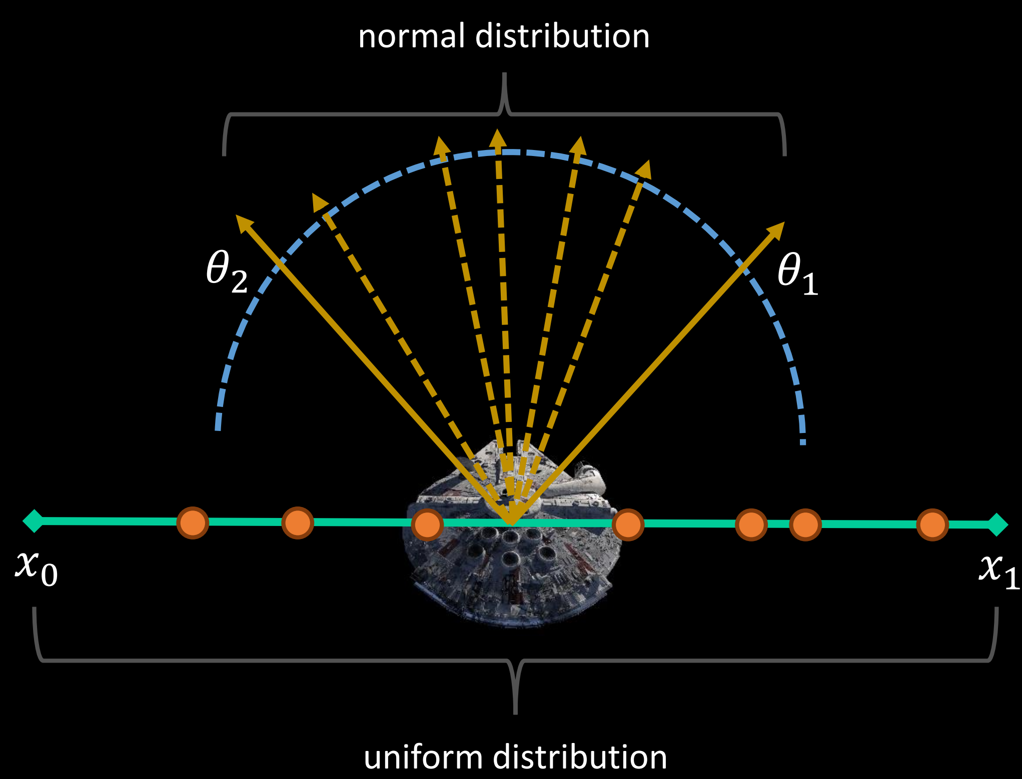
- 3D Monte-Carlo simulation
  - position $\quad \mathrm{p}_0 \in [x_1, x_2]$
  - trajectory $\quad \bar{v}_0 \in [\theta_1, \theta_2] \rightarrow (x, y)$
  - velocity magnitude $|v_0| \in [0, v_{max}]$

- p_0 and $|v_0|$ are selected from a uniform distribution

- the direction $\bar{v}_0$ is selected from a normal distribution

- The initial velocity is given by:

$$v = |v_0| \cdot \bar{v}_0$$



Physical simulation

- Integration for estimating the ship position $p(t)$
    - Euler integration
    - any ***net*** force above F_{max} will destroy your ship
    - the time step Δt should be selected such that a highly-accurate calculation of the path won't destroy your ship

Code

- Put all of your functions in one Matlab (----.m) file
- Implement a function that performs Euler integration
  - a start state (position, velocity, trajectory)
  - a list of gravity wells (positions and masses)
- Create separate functions that generate random values for the start state

- You may use any ‘built-in’ Matlab functions
  - No downloading code from Matlab Central

- Vectorize your code wherever possible to take advantage of Matlab’s speed
  - We’ll talk about this in detail