Numeric solution of ordinary differential equations

Many laws of the Physical Sciences are expressed in terms of *derivatives*. e.g. **Newton's Second Law** of (non-relativistic) dynamics:

mass x acceleration = vector sum of forces

In one dimension, *acceleration* is the *rate of change of velocity*, and *velocity* is the *rate of change of displacement*. With appropriate symbols we could write a set of **differential equations** to describe subsequent motion, given a knowledge of the **initial conditions.**

$t = 0$

$v = v_0$

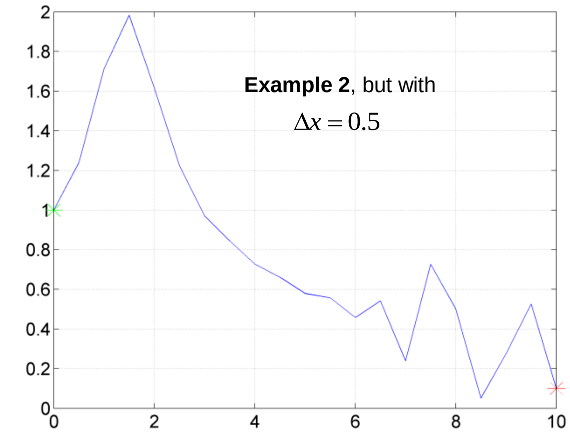
$x = x_0$

So a **second order differential equation** can often be written in terms of a **coupled set of first order equations**

$$\boxed{\begin{aligned} ma &= F(x,t) \\ a &= \frac{dv}{dt} = \frac{d^2x}{dt^2} \\ v &= \frac{dx}{dt} \end{aligned}}$$

Some differential equations can be solved analytically, i.e. in terms of an expression comprising of basic Mathematical functions. However, in general this is *not* possible. To make further progress we need a **numerical method** that we can apply using a computational tool such as Excel or MATLAB to evaluate in an *iterative* fashion. Most methods will assume a fixed, small time step (or x step) and 'solve' the equation *approximately*.

Making the finite difference (i.e. the x step) **smaller** introduces *less error* into the solver



Euler's Method is probably the simplest, but least precise method.

First order equations – i.e. in terms of a single derivative

For the Euler method, it can be shown that errors propagate in direct proportion to the size of Δx

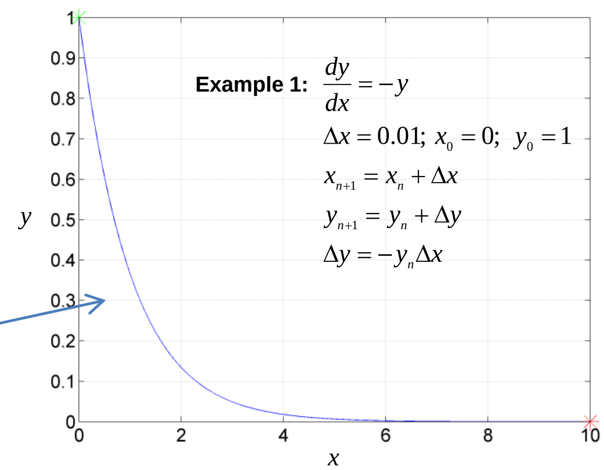
$$\boxed{\frac{dy}{dx} = f(x); \quad y = y_0 \text{ when } x = x_0}$$

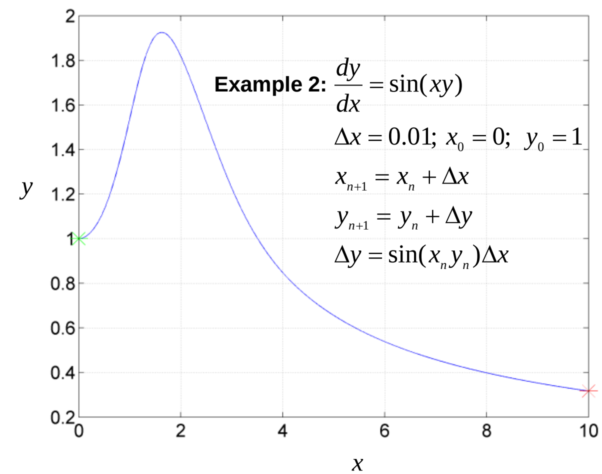
$$\boxed{\begin{aligned} x_{n+1} &= x_n + \Delta x \\ y_{n+1} &= y_n + \Delta y \\ \frac{dy}{dx} &\approx \frac{\Delta y}{\Delta x} \therefore \Delta y = f(x)\Delta x \\ \therefore y_{n+1} &= y_n + f(x_n)\Delta x \end{aligned}}$$

Define a finite x step and compute in an iterative fashion

$n = 0,1,2,3,4.....$

Approximate the derivative and hence find the y change between steps





Note this particular example has an *exact* solution

$$y = e^{-x}$$

Leonhard Euler
1707-1783

Euler's Method can be extended for equations **involving higher derivatives**, by creating a system of **coupled first order equations**

$$\frac{d^2y}{dx^2} = f\left(\frac{dy}{dx}, y, x\right); \quad y = y_0 \ \& \ \frac{dy}{dx} = m_0 \ \text{when} \ x = x_0$$

$$m = \frac{dy}{dx}$$

i.e. a coupled set of first order equations

$$\therefore \frac{dm}{dx} = f(m, y, x)$$

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

$$m_{n+1} = m_n + \Delta m$$

$$y_{n+1} = y_n + \Delta y$$

$$\Delta m = f(m_n, y_n, x_n)\Delta x$$

$$\Delta y = m_n \Delta x$$

Example using Euler's method:

$$\frac{d^2y}{dx^2} = -y - \left(\frac{dy}{dx}\right)^3$$

$$m = \frac{dy}{dx}$$

$$\therefore \frac{dm}{dx} = -y - m^3$$

$$\Delta x = 0.01; \ x_0 = 0; \ m_0 = 1; \ y_0 = 0$$

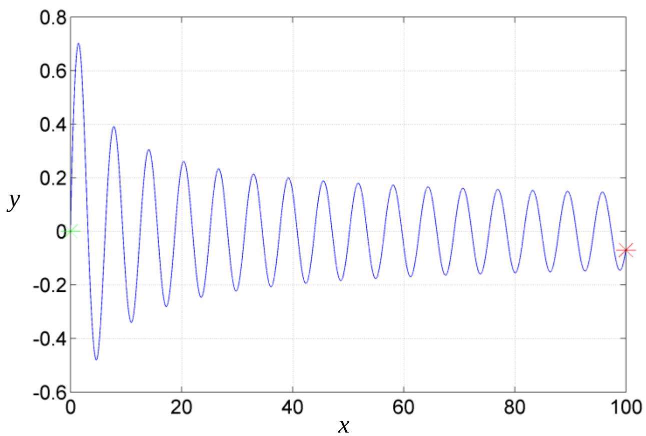
$$x_{n+1} = x_n + \Delta x$$

$$m_{n+1} = m_n + \Delta m$$

$$y_{n+1} = y_n + \Delta y$$

$$\Delta m = \left(-y_n + m_n^3\right)\Delta x$$

$$\Delta y = m_n \Delta x$$



We can readily extend the method to two, three or higher dimensional problems which are *parameterized* in terms of a single variable by using **vectors**.

$$\mathbf{r} = \begin{pmatrix} x \\ y \end{pmatrix}$$

$$\frac{d^2\mathbf{r}}{dt^2} = f\left(\frac{d\mathbf{r}}{dt}, \mathbf{r}, \mathrm{t}\right); \quad \mathbf{r} = \begin{pmatrix} x_0 \\ y_0 \end{pmatrix} \ \& \ \frac{d\mathbf{r}}{dt} = \begin{pmatrix} u_x \\ u_y \end{pmatrix} \ \text{when} \ t = 0$$

$$\mathbf{v} = \frac{d\mathbf{r}}{dt} = \begin{pmatrix} \frac{dx}{dt} \\ \frac{dy}{dt} \end{pmatrix}$$

$$\therefore \frac{d\mathbf{v}}{dt} = f(\mathbf{v}, \mathbf{r}, \mathrm{t})$$

$$t_{n+1} = t_n + \Delta t$$

$$\mathbf{v}_{n+1} = \mathbf{v}_n + \Delta\mathbf{v}$$

$$\mathbf{r}_{n+1} = \mathbf{r}_n + \Delta\mathbf{r}$$

$$\Delta\mathbf{v} = f(\mathbf{v}_n, \mathbf{r}_n, \mathrm{t}_n)\Delta t$$

$$\Delta\mathbf{r} = \mathbf{v}_n \Delta t$$

2D Example using Euler's method

$$\mathbf{r} = \begin{pmatrix} x \\ y \end{pmatrix}$$

$$\frac{d^2\mathbf{r}}{dt^2} = -g\begin{pmatrix} 0 \\ 1 \end{pmatrix} - k\mathbf{v}|\mathbf{v}|; \quad \mathbf{r} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \ \& \ \frac{d\mathbf{r}}{dt} = \tfrac{10}{\sqrt{2}}\begin{pmatrix} 1 \\ 1 \end{pmatrix} \ \text{when} \ t = 0$$

$$\mathbf{v} = \frac{d\mathbf{r}}{dt}$$

$$\therefore \frac{d\mathbf{v}}{dt} = -g\begin{pmatrix} 0 \\ 1 \end{pmatrix} - k\mathbf{v}|\mathbf{v}|$$

$$t_{n+1} = t_n + \Delta t$$

$$\mathbf{v}_{n+1} = \mathbf{v}_n + \Delta\mathbf{v}$$

$$\mathbf{r}_{n+1} = \mathbf{r}_n + \Delta\mathbf{r}$$

$$\Delta\mathbf{v} = \left(-g\begin{pmatrix} 0 \\ 1 \end{pmatrix} - k\mathbf{v}|\mathbf{v}|\right)\Delta t$$

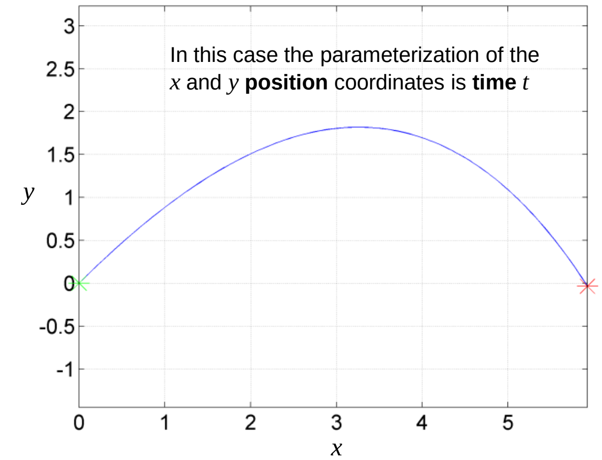
$$\Delta\mathbf{r} = \mathbf{v}_n \Delta t$$

This is essentially **projectile motion** with **air-resistance** proportional to the square of velocity

In the example below

$g = 9.81$

$k = 0.1$



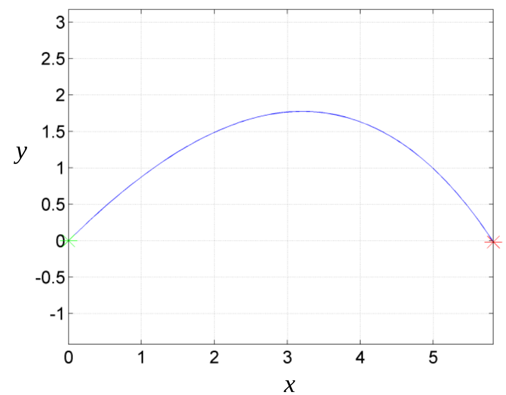
There are various methods which can offer greater precision given a fixed step size. The **Runge-Kutta** method below will extend to higher derivatives and vector equations in exactly the same way as described for the Euler method. The **Verlet** method is really specific to kinematic problems, but will naturally extend to two or three dimensions using vectors.

Verlet Method – assume 'constant acceleration motion' between time steps

$$v = \frac{dx}{dt}$$
$$\frac{dv}{dt} = a(x,v,t); \quad x = x_0 \;\&\; v = u \quad \text{when } t = 0$$
$$t_{n+1} = t_n + \Delta t$$
$$v_{n+1} = v_n + \Delta v$$
$$x_{n+1} = x_n + \Delta x$$
$$\Delta x = v_n \Delta t + \tfrac{1}{2} a(x_n, v_n, t_n)(\Delta t)^2$$
$$V = v_n + a(x_n, v_n, t_n)\Delta t$$
$$\Delta v = \tfrac{1}{2}\left\{ a(x_n, t_n) + a(x_{n+1}, V, t_{n+1}) \right\}\Delta t$$

This is a **second order** method
i.e. errors are proportional to $(\Delta x)^2$

Note if acceleration is functionally dependent on velocity we have to use a first order Euler method to evaluate the second acceleration term.



$$\mathbf{r} = \begin{pmatrix} x \\ y \end{pmatrix}$$

2D Example using the Verlet method

$$\frac{d^2\mathbf{r}}{dt^2} = -g\begin{pmatrix} 0 \\ 1 \end{pmatrix} - k\mathbf{v}|\mathbf{v}|; \quad \mathbf{r} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \;\&\; \frac{d\mathbf{r}}{dt} = \tfrac{10}{\sqrt{2}}\begin{pmatrix} 1 \\ 1 \end{pmatrix} \text{ when } t = 0$$
$$\mathbf{v} = \frac{d\mathbf{r}}{dt}$$
$$\therefore \frac{d\mathbf{v}}{dt} = -g\begin{pmatrix} 0 \\ 1 \end{pmatrix} - k\mathbf{v}|\mathbf{v}|$$
$$t_{n+1} = t_n + \Delta t$$
$$\mathbf{v}_{n+1} = \mathbf{v}_n + \Delta\mathbf{v}$$
$$\mathbf{r}_{n+1} = \mathbf{r}_n + \Delta\mathbf{r}$$
$$\Delta\mathbf{r} = v_n \Delta t + \tfrac{1}{2}\left(-g\begin{pmatrix} 0 \\ 1 \end{pmatrix} - k\mathbf{v}_n|\mathbf{v}_n| \right)(\Delta t)^2$$
$$\mathbf{V} = \mathbf{v}_n + \left(-g\begin{pmatrix} 0 \\ 1 \end{pmatrix} - k\mathbf{v}_n|\mathbf{v}_n| \right)\Delta t$$
$$\Delta\mathbf{v} = \tfrac{1}{2}\left(-g\begin{pmatrix} 0 \\ 1 \end{pmatrix} - k\mathbf{v}_n|\mathbf{v}_n| \right)\Delta t + \tfrac{1}{2}\left(-g\begin{pmatrix} 0 \\ 1 \end{pmatrix} - k\mathbf{V}|\mathbf{V}| \right)\Delta t$$

This is essentially **projectile motion** with **air-resistance** proportional to the square of velocity.

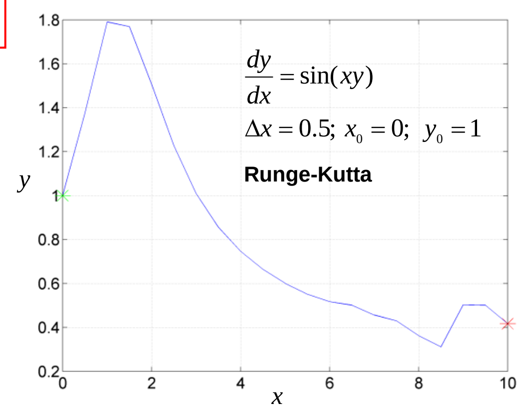
In this example: $g = 9.81$
$k = 0.1$

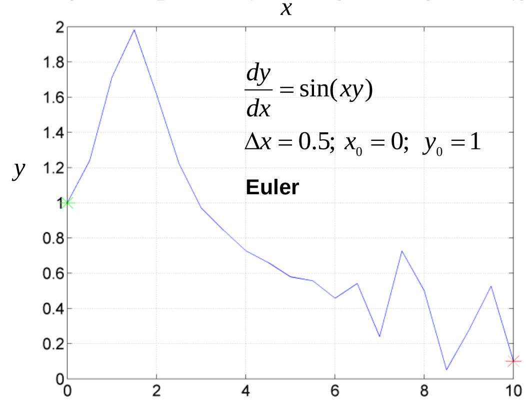
Runge-Kutta method

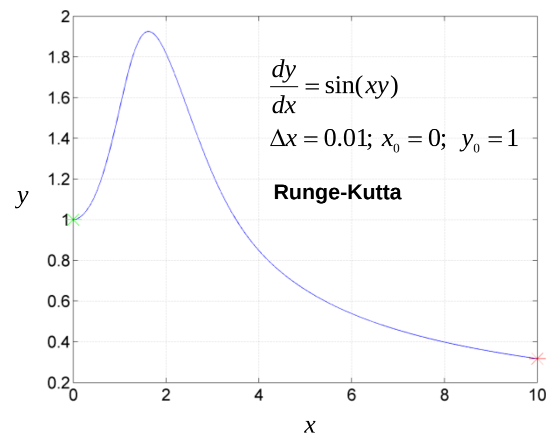
$$\frac{dy}{dx} = f(y,x); \quad y = y_0 \text{ when } x = x_0$$

$$x_{n+1} = x_n + \Delta x$$
$$y_{n+1} = y_n + \Delta y$$
$$k_1 = f(y_n, x_n)$$
$$k_2 = f\left(y_n + \tfrac{1}{2}\Delta x k_1,\; x_n + \tfrac{1}{2}\Delta x\right)$$
$$k_3 = f\left(y_n + \tfrac{1}{2}\Delta x k_2,\; x_n + \tfrac{1}{2}\Delta x\right)$$
$$k_4 = f\left(y_n + \Delta x k_3,\; x_n + \Delta x\right)$$
$$\Delta y = \tfrac{1}{6}\Delta x\left(k_1 + 2k_2 + 2k_3 + k_4\right)$$

This is a **fourth order** method
i.e. errors are proportional to $(\Delta x)^4$







Comparing the effect of reducing the finite x difference between the **Euler** and the **Runge-Kutta** solvers

Carl Runge
1856-1927

Martin Kutta
1867-1944

Solar system simulation using the Verlet method

$$\mathbf{r} = \begin{pmatrix} x \\ y \end{pmatrix}$$
$$m\frac{d\mathbf{v}}{dt} = -\frac{GMm}{|\mathbf{r}|^3}\mathbf{r}$$
$$\mathbf{v} = \frac{d\mathbf{r}}{dt}$$

Newton's law of Universal Gravitation. M is the mass of the Sun, which is assumed to be so massive compared to the other planets that the interaction of their gravity on it is negligible. i.e. it is fixed in position.

The gravitational interaction *between* the planets is also ignored.

Using **Kepler's Third law**

$$T^2 = \frac{4\pi^2}{GM}a^3$$
$$\therefore 1\text{Yr}^2 = \frac{4\pi^2}{GM}1\text{AU}^3$$
$$\therefore GM = 4\pi^2\frac{\text{AU}^3}{\text{Yr}^2}$$
$$\therefore \frac{d\mathbf{v}}{dt} = -\frac{4\pi^2}{|\mathbf{r}|^3}\mathbf{r}$$

T is the orbital period
a is the semi-major axis of the orbit (in general, bound orbits are *ellipses*)

Yr is an Earth year
AU is an astronomical unit
i.e. the average Earth-Sun separation

If lengths are measured in AU and times in Years. Note planet masses cancel and therefore are not required.

Can't readily **Runge-Kutta** since the left had side of the differential equation is functionally dependent on *neither* velocity or time

Solver using the **Verlet method**

$$t_{n+1} = t_n + \Delta t$$
$$\mathbf{r}_{n+1} = \mathbf{r}_n + \mathbf{v}_n\Delta t - \tfrac{1}{2}\frac{4\pi^2}{|\mathbf{r}_n|^3}\mathbf{r}_n(\Delta t)^2$$
$$\mathbf{v}_{n+1} = \mathbf{v}_n - 4\pi^2\Delta t\,\tfrac{1}{2}\left\{\frac{1}{|\mathbf{r}_n|^3}\mathbf{r}_n + \frac{1}{|\mathbf{r}_{n+1}|^3}\mathbf{r}_{n+1}\right\}$$

i.e. constant acceleration motion between time steps

Use the *average acceleration* between the time steps to work out the new velocity

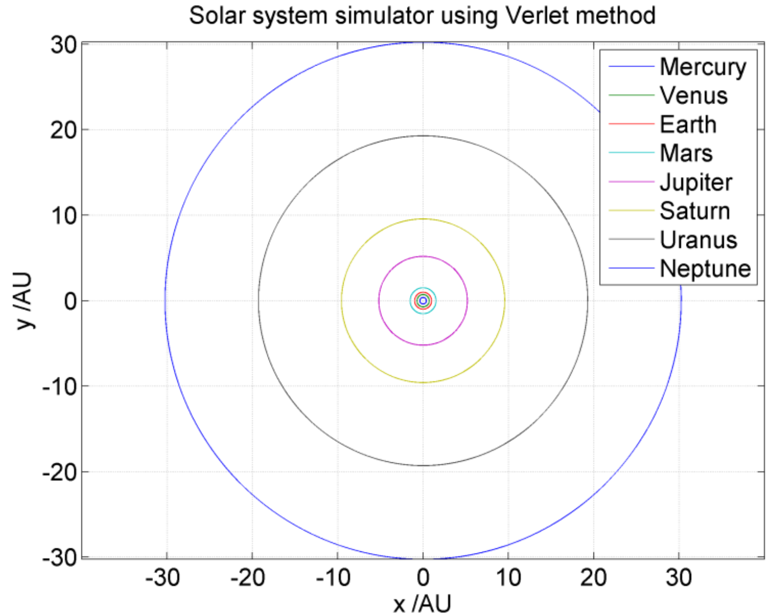
Initial conditions for circular orbits.

In (Yr, AU) units, Kepler III is:

$$\boxed{T^2 = a^3}$$

Circular Motion:

$$v = \frac{2\pi r}{T}$$
$$r = a$$
$$\therefore v = \frac{2\pi a}{a^{\frac{3}{2}}}$$
$$\therefore v = \frac{2\pi}{\sqrt{a}}$$
$$\therefore \mathbf{r}_0 = \begin{pmatrix} a \\ 0 \end{pmatrix}; \quad \mathbf{v}_0 = \begin{pmatrix} 0 \\ \frac{2\pi}{\sqrt{a}} \end{pmatrix}$$



$$M_\odot = 1.99\times10^{30}\,\text{kg}$$
$$G = 6.67\times10^{-11}\,\text{Nm}^2\text{kg}^{-2}$$
$$\text{AU} = 1.49597871\times10^{11}\,\text{m}$$
$$24\times3600\text{s} = 1\text{day}$$
$$M_\odot = 332{,}837 m_\oplus$$
$$m_\oplus = 5.972\times10^{24}\,\text{kg}$$

Planet	*T* / years	*r* / AU	*m* / Earth masses	Rotation period /days	Orbital eccentricity
Mercury	0.241	0.387	0.055	58.646	0.21
Venus	0.615	0.723	0.815	243.018	0.01
Earth	1.000	1.000	1.000	1.000	0.02
Mars	1.881	1.523	0.107	1.026	0.09
Jupiter	11.861	5.202	317.85	0.413	0.05
Saturn	29.628	9.576	95.159	0.444	0.06
Uranus	84.747	19.293	14.5	0.718	0.05
Neptune	166.344	30.246	17.204	0.671	0.01
Pluto	248.348	39.509	0.003	6.387	0.25