Introduction to Programming and Computational Physics

Lesson n.11

Ordinary Differential Equations:

- Euler’s method

- Runge-Kutta method

Differential equations in physics

Physics laws are often expressed by differential equations.

Some of them can be analytically solved, but the majority has no exact solution.

For this reason, we often resort to a numerical approach.

Simple pendulum equation

$$\frac{d^2\theta}{dt^2} + \frac{g}{l}\sin\theta = 0$$

was solved with the approximation

$$\sin\theta \cong \theta$$

working for *small* angles

NB: Taylor series

$$\sin\theta = \theta - \frac{\theta^3}{3!} + \frac{\theta^5}{5!} - \frac{\theta^7}{7!} + \ldots$$

Ordinary differential equations

An ordinary differential equation (ODE) is a differential equation in which the unknown function is a function of a *unique* independent variable

Let y(x) be an unknown function, an n-th order ODE is

$$y^{n}(x) = f(x, y(x), y'(x), \ldots, y^{n-1}(x))$$

Problems involving ODEs can always be reduced to the study of a sets of first-order differential equations.
The equation of a small body in the gravitational potential of a body of mass M:

$$\frac{d^2\vec{r}}{dt^2} = -\frac{GM}{r^3}\vec{r} \quad \Longrightarrow \quad \begin{aligned} \frac{d\vec{r}}{dt} &= \vec{v}(t) \\ \frac{d\vec{v}}{dt} &= -\frac{GM}{r^3}\vec{r} \end{aligned}$$

The generic problem of solving an ordinary differential equation is then reduced to solving a set of N coupled first order differential equations for the functions

$$y_i \quad i = 1,2,...,N$$

having the general form

$$\frac{dy_i(x)}{dx} = f_i(x, y_1, y_2,..., y_N) \quad i = 1,2,...,N$$

where the functions f_i on the right-hand side are known

Boundary conditions

Initial value problems

All the y_i are given at some starting value x_s and it is requested to find the values for y_i at some final point x_f or at a some discrete list of points

Boundary value problems

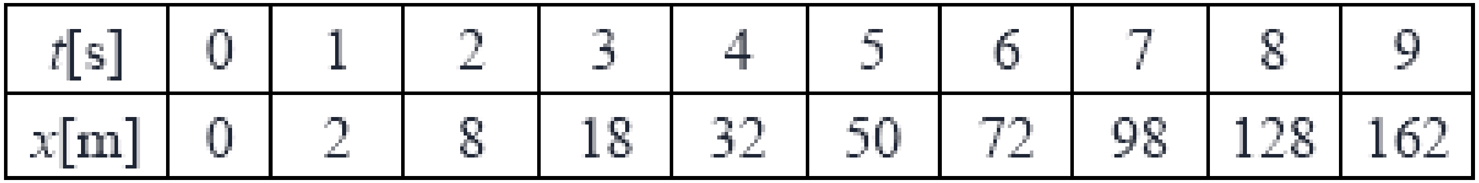
Boundary conditions are specified at more then one x.
Typically some will be defined at x_s and some at x_f

We will deal only with initial value problems

Numerical solutions

If it’s not possible to find an analytical solution, we need to resort to a numerical one.

Numerical solutions are not functions expressed in analytical form, but as a set of points (as close as we desire)



t[s]	0	1	2	3	4	5	6	7	8	9
x[m]	0	2	8	18	32	50	72	98	128	162

Numerical solutions are always approximated. The degree of approximation, in general, can be always improved, provided that adequate computational instruments are available.

Numerical solutions do not contain all the information contained in an analytical solution (roots, maxima/minima….)

Numerical methods

Given the first order differential equation $y' = f(x, y)$

and initial condition $y(x_0) = y_0$ we will fix a *stepsize* h

and then we will propagate the solution to the point $x_1 = x_0 + h$

and use $y(x_1) = y_1$ as initial condition for the next step and so on

Multi-step methods

The information from previous steps is used in order to obtain the value at the new step.

Multi-stage methods

Intermediate stages between the current and the next step are computed in order to advance the solution. Previous steps are not used and once the step has been completed, the intermediate stages are not used as well.

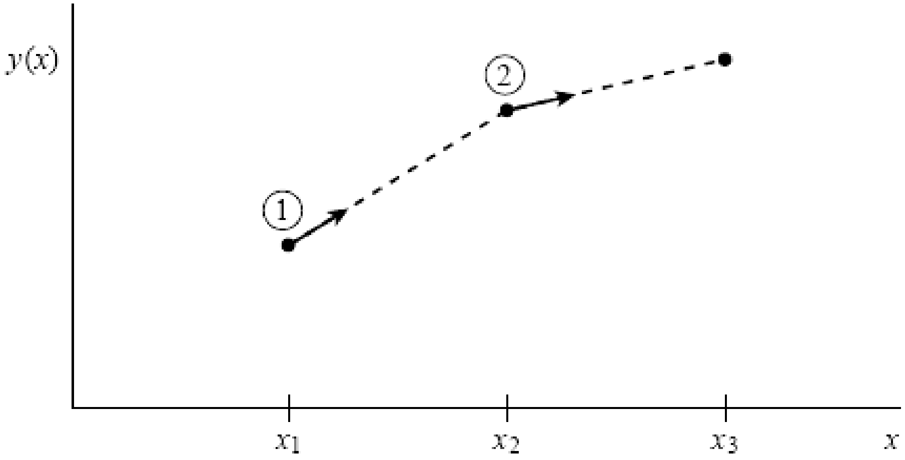
Euler’s method

It is probably the simplest (and less accurate) method. The derivative at the starting point of each interval is extrapolated to find the next function value.

The Euler’s method formula is:

$$y_{n+1} = y_n + hf(x_n, y_n) \qquad f = dy/dx$$

which advances a solution from x_n to $x_{n+1} \equiv x_n + h$



Euler’s method

The formula is asymmetric: it advances the solution through the interval h using derivative information only at the beginning of the interval.

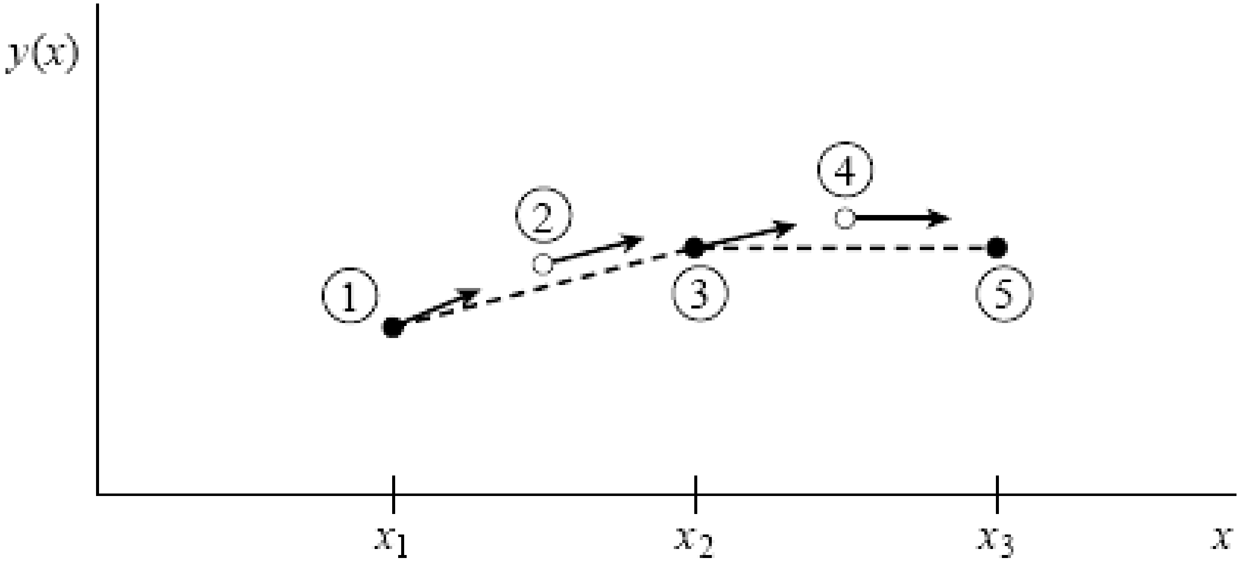
The error is $O(h^2)$ (can be demonstrated by Taylor expansion)

⇨ in order to reduce the error we have to reduce the stepsize h

Euler’s method is not recommended for any practical use. Anyway, the other methods come from it’s general idea: add small increments to the function corresponding to derivatives multiplied by step-size

Runge-Kutta second order method

(also known as midpoint method)



The initial derivative at each step is used to find a point halfway across the interval. Then the derivative at this point is used across the full width of the interval.

The method is O(h^3) accurate.

The propagation of the solution from (x_n, y_n) to (x_{n+1}, y_{n+1})

is obtained using the formulas:

$$k_1 = hf(x_n, y_n)$$

$$k_2 = hf(x_n + \tfrac{1}{2}h, y_n + \tfrac{1}{2}k_1)$$

$$y_{n+1} = y_n + k_2 + O(h^3)$$

Exemple:

Given the differential equation $y' = -x^2 y^2$

with $(x_0, y_0) = (0,1)$ and $h = 0.2$

First step:

$$k_1 = hf(x_0, y_0) = hf(0,1) = 0.2 * 0 = 0$$

$$k_2 = hf(x_0 + \tfrac{1}{2}h, y_0 + \tfrac{1}{2}k_1) = hf(0.1,1) = 0.2 * (-0.01) = -0.002$$

$$y_1 = y_0 + k_2 = 0.998$$

first step (0,1)→(0.2,0.998)

Second step:

$$k_1 = hf(x_1, y_1) = hf(0.2, 0.998) = \ldots$$

$$k_2 = hf(x_1 + \tfrac{1}{2}h, y_1 + \tfrac{1}{2}k_1) = hf(0.3, 0.998 + \tfrac{1}{2}k_1) = \ldots$$

$$y_2 = 0.998 + k_2$$

and so on…

An application: viscosity

Let's apply the two methods to a case where we know the analytical solution in order to see which degree of approximation we reach.

$$F(v(t)) = -\lambda v(t) = mv'(t)$$

$$v'(t) = f(t, v(t)) = -\frac{\lambda}{m} v(t) \quad \rightarrow \quad f(t_n, v_n) = -\frac{\lambda}{m} v_n$$

Exact solution:

$$v(t) = v_0 e^{-\frac{\lambda}{m} t}$$

Euler's method:

$$v_{n+1} \cong v_n + f(t_n, v_n)\Delta t = v_n - \tfrac{\lambda}{m} v_n \Delta t = v_n (1 - \tfrac{\lambda}{m} \Delta t)$$

Runge-Kutta second order method:

$$k_1 = f(t_n, v_n)\Delta t = -\tfrac{\lambda}{m} v_n \Delta t$$

$$k_2 = f(t_n + \tfrac{\Delta t}{2}, v_n + \tfrac{k_1}{2})\Delta t = -\tfrac{\lambda}{m}(v_n - \tfrac{\lambda}{2m} v_n \Delta t)\Delta t$$

$$v_{n+1} \cong v_n + k_2 = v_n \left(1 - \tfrac{\lambda}{m} \Delta t + \tfrac{\lambda^2}{2m^2} \Delta t^2\right)$$

If we assume: m = 1 g, v_0=10 m/s, λ = 20 g/s

t[s]	v[m/s]			
	Euler Δt =0.01s	**Euler Δt =0.001s**	**Runge-Kutta Δt =0.01s**	**Exact**
0.00	10.00	10.00	10.00	10.00
0.01	8.00	8.17	8.20	8.19
0.02	6.40	6.68	6.72	6.70
0.03	5.12	5.45	5.51	5.49
0.04	4.10	4.46	4.52	4.49
0.05	3.28	3.64	3.71	3.68
0.06	2.62	2.98	3.04	3.01
0.07	2.10	2.43	2.49	2.47
0.08	1.68	1.99	2.04	2.02
0.09	1.34	1.62	1.67	1.65

Runge-Kutta fourth order method

$$k_1 = hf(x_n, y_n)$$

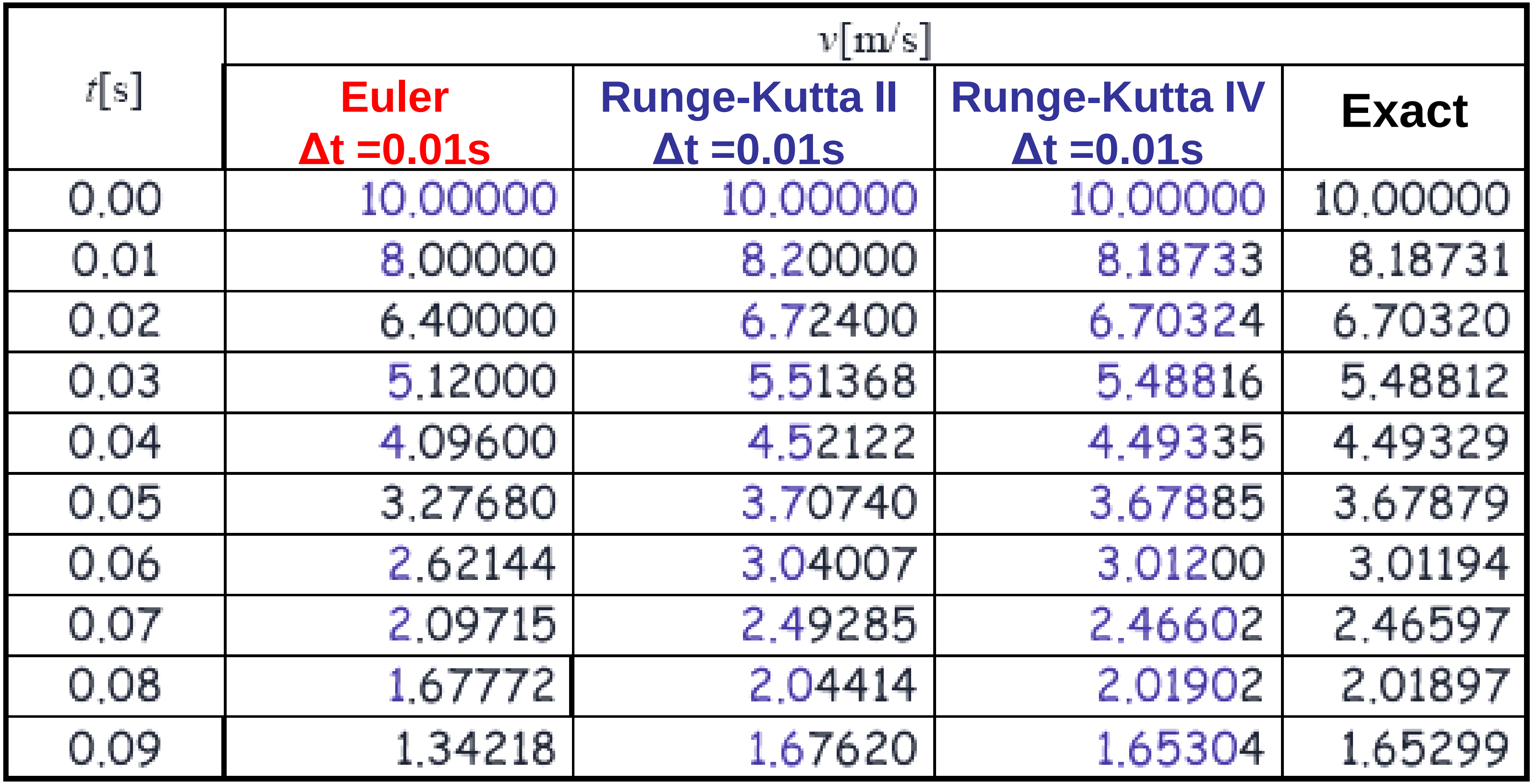
$$k_2 = hf(x_n + \tfrac{h}{2}, y_n + \tfrac{k_1}{2})$$

$$k_3 = hf(x_n + \tfrac{h}{2}, y_n + \tfrac{k_2}{2})$$

$$k_4 = hf(x_n + h, y_n + k_3)$$

$$y_{n+1} = y_n + \frac{k_1}{6} + \frac{k_2}{3} + \frac{k_3}{3} + \frac{k_4}{6} + O(h^5)$$

It’s by far the most often used , but computation time is longer



t[s]	v[m/s]			
	Euler Δt =0.01s	Runge-Kutta II Δt =0.01s	Runge-Kutta IV Δt =0.01s	Exact
0.00	10.00000	10.00000	10.00000	10.00000
0.01	8.00000	8.20000	8.18733	8.18731
0.02	6.40000	6.72400	6.70324	6.70320
0.03	5.12000	5.51368	5.48816	5.48812
0.04	4.09600	4.52122	4.49335	4.49329
0.05	3.27680	3.70740	3.67885	3.67879
0.06	2.62144	3.04007	3.01200	3.01194
0.07	2.09715	2.49285	2.46602	2.46597
0.08	1.67772	2.04414	2.01902	2.01897
0.09	1.34218	1.67620	1.65304	1.65299

Second order equations

$$y' = f(x, y, z)$$
$$z' = g(x, y, z)$$

x is the indipendent variable, functions y and z are unknown

initial conditions

$$y(x_0) = y_0$$
$$z(x_0) = z_0$$

Runge-Kutta II order method

$$k_1 = hf(x_0, y_0, z_0)$$
$$l_1 = hg(x_0, y_0, z_0)$$
$$k_2 = hf(x_0 + \tfrac{1}{2}h, y_0 + \tfrac{1}{2}k_1, z_0 + \tfrac{1}{2}l_1)$$
$$l_2 = hg(x_0 + \tfrac{1}{2}h, y_0 + \tfrac{1}{2}k_1, z_0 + \tfrac{1}{2}l_1)$$
$$y_1 = y_0 + k_2 + O(h^3)$$
$$g_1 = g_0 + l_2 + O(h^3)$$

Remarks

Higher order does not necessarily imply better precision (discontinuities, …)

We introduced “fixed stepsize” formulas, but often an adaptive stepsize works better