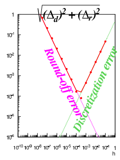
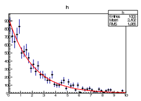
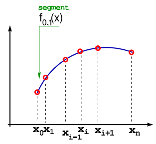
Computational Physics

numerical methods with C++ (and UNIX)







Fernando Barao
Instituto Superior Tecnico, Dep. Fisica
email: barao at lip.pt

Computational Physics

Physics problems

and Solutions

Fernando Barao, Phys Department IST (Lisbon)

Numerical methods

✔ Solving Ordinary Differential Equations

- ► Euler method
- ► Runge-Kutta method
- ► examples

Ordinary Differential Equations

✔ Ordinary Differential Equations involve only derivatives with respect to a single variable, usually time

$$\frac{dy}{dt} = f(t, y) \qquad \text{Ex:} \quad \frac{dy}{dt} + \alpha\, y = 0 \qquad \text{(decay equation)}$$

✔ Higher order differential equations

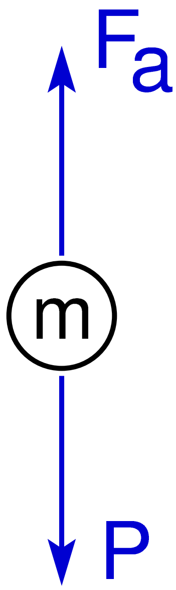
$$\frac{d^2y}{dt^2} + \lambda\frac{dy}{dt} = f(t, \frac{dy}{dt}, y) \qquad \text{Ex:} \quad \frac{d^2y}{dt^2} + \frac{\lambda}{m}\frac{dy}{dt} + \frac{k}{m}y = 0 \qquad \text{(damped harmonic osc)}$$

✔ Can be reduced to first-order by redefining dependent variables

$$\left\{ \begin{array}{l} \frac{dy}{dt} = v \equiv y^{(1)} \\ y \equiv y^{(0)} \end{array} \right. \Rightarrow \left\{ \begin{array}{l} \frac{dy^{(0)}}{dt} = y^{(1)}(t) \\ \frac{dy^{(1)}}{dt} = f(t, y^{(0)}, y^{(1)}) \end{array} \right. \Rightarrow \boxed{\frac{d\vec{y}}{d} = f(t, \vec{y})}$$

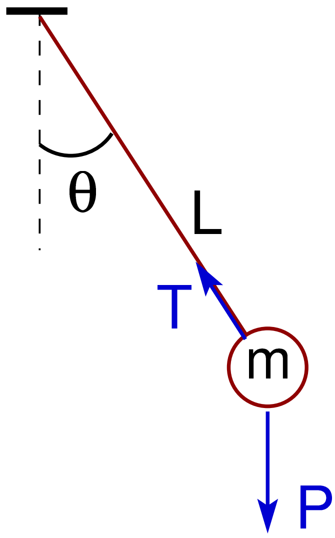
Examples

free fall with friction



$$\boxed{m\frac{d^2x}{dt^2} = mg - k\frac{dx}{dt}}$$

pendulum motion

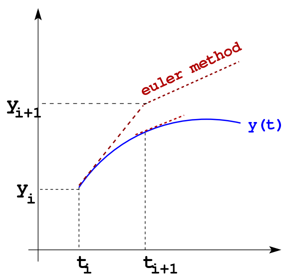


$$\boxed{\frac{d^2\theta}{dt^2} + \frac{g}{\ell}\sin\theta = 0}$$

1st order ODE: numerical solutions

✔ **Solution:**

$$\frac{d\vec{y}}{dt} = f(t,\vec{y}) \quad \Rightarrow \quad \vec{y}(t) = \vec{y}(t_0) + \int_{t_0}^{t} f(t',\vec{y}(t'))dt'$$



✔ **Euler method (1st order accurate)**

$$y_{n+1} = y_n + \delta t \left.\frac{dy}{dt}\right|_n + \mathcal{O}[(\delta t)^2]\cdots$$

using the forward difference approximation for the derivative:

$$\left.\frac{dy}{dt}\right|_n \simeq \frac{y_{n+1}-y_n}{\delta t}$$

the differential equation becomes:

$$\boxed{\frac{dy}{dt} = f[t,y(t)] \quad \Rightarrow \quad y_{n+1} = y_n + (\delta t)\ f[t_n,y(t_n)] + \mathcal{O}((\delta t)^2)}$$

✔ Stability

Suppose an error is introduced in the iteration value (δy) - like a round-off for instance - causing therefore a progressive deviation from the nominal numerical value

$$y_{n+1} + \delta y_{n+1} = y_n + \delta y_n - \delta t\left[f(t_n,y(t_n)) + \left.\frac{\partial y}{\partial t}\right|_n \delta y_n\right] \quad \Rightarrow \quad \boxed{\delta y_{n+1} = \delta y_n\left[1 - \delta t \left.\frac{\partial y}{\partial t}\right|_n\right]}$$

1st order ODE: numerical solutions

- ✔ Solution of the 1st-order equation

  $\frac{d\vec{y}}{dt} = f(t, \vec{y})$

- ✔ **Predictor-Corrector (Crank-Nicolson)**
  using the average of the two slopes at **i** and **i+1**:

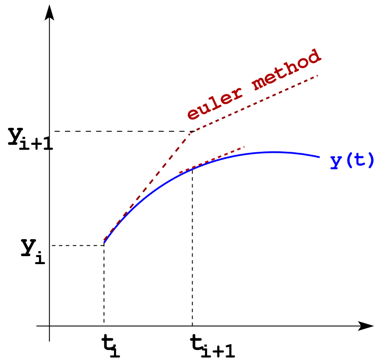
  $y(t_{i+1}) = y(t_i) + (\delta t)\frac{1}{2}\left[\left.\frac{dy}{dt}\right|_i + \left.\frac{dy}{dt}\right|_{i+1}\right]$

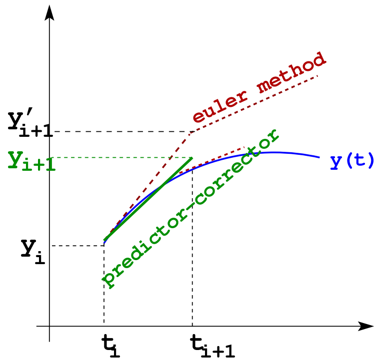
- ✔ Accuracy

  $O((\delta t)^3)$ $\Rightarrow$ second-order accurate

- ✔ algorithm

  - compute the slope at t_i : $f(t_i, y_i)$
  - user Euler approach to make a prediction for next slope value:
    $y'_{t+i} = y(t_i) + (\delta t) f(t_i, y_i) \quad \Rightarrow \quad f(t_{i+1}, y'_{i+1})$
  - average slopes and get next iteration:
    $y_{i+1} = y_i + \frac{\delta t}{2}\left[f(t_i, y_i) + f(t_{i+1}, y_{i+1})\right]$





1st order ODE: numerical solutions

- ✔ **Leap-Frog method (Stormer-Verlet)**
  using the centered difference approximation for the derivative:

  $\left.\frac{dy}{dt}\right|_n \simeq \frac{y_{n+1} - y_{n-1}}{2\delta t}$

  the differential equation becomes:

  $$\boxed{\left.\frac{dy}{dt}\right|_n = f[t_n, y(t_n)] \quad \Rightarrow \quad y_{n+1} = y_{n-1} + 2\,(\delta t)\; f[t_n, y(t_n)]}$$

- ✔ Accuracy

  $O((\delta t)^3)$ $\Rightarrow$ second-order accurate

- ✔ Stability

  $\delta y_{n+1} = \delta y_{n-1} - 2\;\delta t\;\left.\frac{\partial y}{\partial t}\right|_n\;\delta y_n$

- ✔ algorithm

  time: $\delta t = (t_f - t_0)/n$
  first iteration: $y_1 = y_0 + \delta t\; f(t_0, y_0) \quad ; \quad t_1 = t_0 + \delta t$
  following iterations (i=1,n-1): $y_{i+1} = y_{i-1} + 2\delta t\; f(t_i, y_i) \quad ; \quad t_{i+1} = t_0 + (i+1)\delta t$

ODE: numerical solution improvement?

Can we improve the numerical solution of $\frac{dy}{dt} = f(t, y)$?

✔ **Use more terms in the Taylor expansion of y_{n+1}**

$$\begin{aligned} y_{n+1} &= y_n + (\delta t) \left.\frac{dy}{dt}\right|_n + \frac{(\delta t)^2}{2} \left.\frac{d^2y}{dt^2}\right|_n + O(h^3) \\ &= y_n + (\delta t)\ f(t_n, y_n) + \frac{(\delta t)^2}{2} \frac{d}{dt}\left[f(t_n, y_n)\right] \\ &= y_n + (\delta t)\ f(t_n, y_n) + \frac{(\delta t)^2}{2} \left(\frac{\partial f}{\partial t} + \frac{\partial f}{\partial y}\frac{\partial y}{\partial t}\right) \end{aligned}$$

Interesting if analytic differentiation possible! Otherwise numerical derivatives...(errors)

✔ **Use intermediate points within one time step (Runge-Kutta methods)**

We have seen that the general solution for the 1st order differential equation was:

$$\frac{dy}{dt} = f(t, y) \quad \Rightarrow \quad y(t) = y(t_0) + \int_{t_0}^{t} f(t', y(t'))\, dt'$$

Considering a small interval $\delta t = t_{n+1} - t_n$, the solution comes:

$$\boxed{y_{n+1} = y_n + \int_{t_n}^{t_{n+1}} f\left[t', y(t')\right]\ dt'}$$

Runge-Kutta of second order (RK2)

✔ Let's use for the integrand $f(t, y)$ a Taylor expansion at 1st-order around an intermediate abcissa $t_{i+\frac{1}{2}} \equiv t_i + h/2$

$$\begin{aligned} f(t, y) &= f(t_{i+1/2}, y_{i+1/2}) + (t - t_{i+1/2} \left(\frac{df}{dt}\right)_{t_{i+1/2}, y_{i+1/2}} + \cdots \\ &= f(t_{i+1/2}, y_{i+1/2}) + (t - t_{i+1/2} \left(\frac{\partial f}{\partial t} + \frac{\partial f}{\partial y}\frac{dy}{dt}\right)_{t_{i+1/2}, y_{i+1/2}} + \cdots \\ &= f(t_{i+1/2}, y_{i+1/2}) + (t - t_{i+1/2} \left(\frac{\partial f}{\partial t} + \frac{\partial f}{\partial y} f(t, y)\right)_{t_{i+1/2}, y_{i+1/2}} + \cdots \end{aligned}$$

✔ The integration in the step interval:

$$\begin{aligned} \int_{t_n}^{t_{n+1}} f\left[t', y(t')\right]\ dt' \quad &= \quad f(t_{i+1/2}, y_{i+1/2}) \int_{t_n}^{t_{n+1}} dt' + \\ &\quad \left(\frac{\partial f}{\partial t} + \frac{\partial f}{\partial y} f(t, y)\right)_{t_{i+1/2}, y_{i+1/2}} \int_{t_n}^{t_{n+1}} (t - t_{i+1/2})\, dt' \\ &= \quad h\ f(t_{i+1/2}, y_{i+1/2})\ + O(h^3) \end{aligned}$$

$$\boxed{y_{i+1} = y_i + h\ f(t_{i+1/2}, y_{i+1/2})\ + O(h^3)}$$

RK2 (cont.)

✔ **algorithm**

- the derivative $f(t_{i+1/2}, y_{i+1/2})$ is computed using the Euler relation

$$\begin{aligned} t_{i+1/2} &= t_i + \tfrac{h}{2} \\ y_{i+1/2} &= y_i + \tfrac{h}{2}\ f(t_i, y_i) \quad \text{(euler relation)} \\ y_{i+1} &= y_i + h\ f(t_{i+1/2}, y_{i+1/2}) \end{aligned}$$

$$\left| \begin{aligned} Y_1 &= h\ f(t_i, y_i) \\ Y_2 &= h\ f\left(t_i + \tfrac{h}{2}, y_i + \tfrac{Y_1}{2}\right) \\ y_{i+1} &= y_i + Y_2 \end{aligned} \right.$$

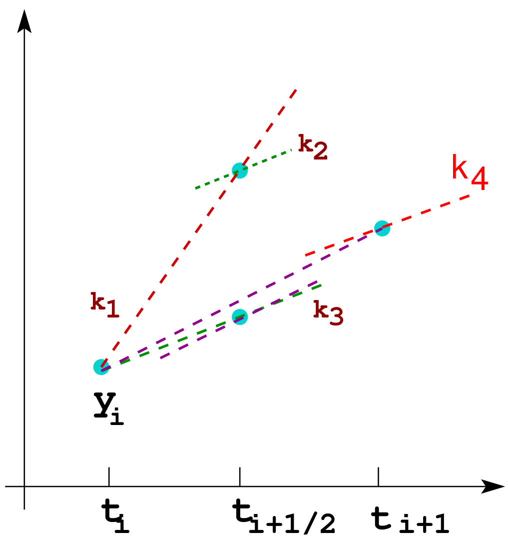
Runge-Kutta of 4th-order (RK4)

✔ Instead of approximating the integral with the midpoint rule we can now use the Simpson rule (2nd-deg polynomial) for the integration in the step interval: $[t_i,\ t_{i+1/2},\ t_{i+1}]$

$$\int_{t_n}^{t_{n+1}} f\left[t', y(t')\right]\ dt' = \frac{h}{6}\left[f(t_i, y_i) + 4\ f(t_{i+1/2}, y_{i+1/2}) + f(t_{i+1}, y_{i+1})\right] + O(h^5)$$

✔ **algorithm**

$$\left| \begin{aligned} Y_1 &= h\ f(t_i, y_i) \\ Y_2 &= h\ f\left(t_i + \tfrac{h}{2}, y_i + \tfrac{Y_1}{2}\right) \\ Y_3 &= h\ f\left(t_i + \tfrac{h}{2}, y_i + \tfrac{Y_2}{2}\right) \\ Y_4 &= h\ f\left(t_i + h, y_i + Y_3\right) \\ y_{i+1} &= y_i + \tfrac{1}{6}\left(Y_1 + 2Y_2 + 2Y_3 + Y_4\right) \end{aligned} \right.$$



adimensional equations

- In a differential equation inbvolving observables like masses, velocities and time for instance, the arbitrary choice of the units can cause problems for numerical calculations
- We shall write the equations invariants to unit changes, i.e. written as function of adimensional observables

  find the characteristic scales of the problem like for length L_c and velocity V_c and define the reduced variables:

  $v = V/V_c$

  $\ell = L/L_c$
- radioactive decay example:

  $dN = -p\ dt\ N$

  $p = 1/\tau$

Example: ODE 1st-order

- Let's solve numerically the 1st-order differential equation:

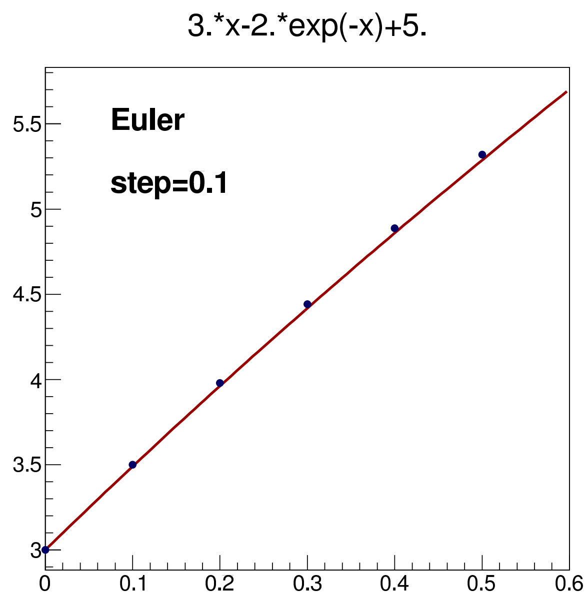
  $\frac{dy}{dx} = 3x - y + 8$

  $x \in [0.0, 0.5]$

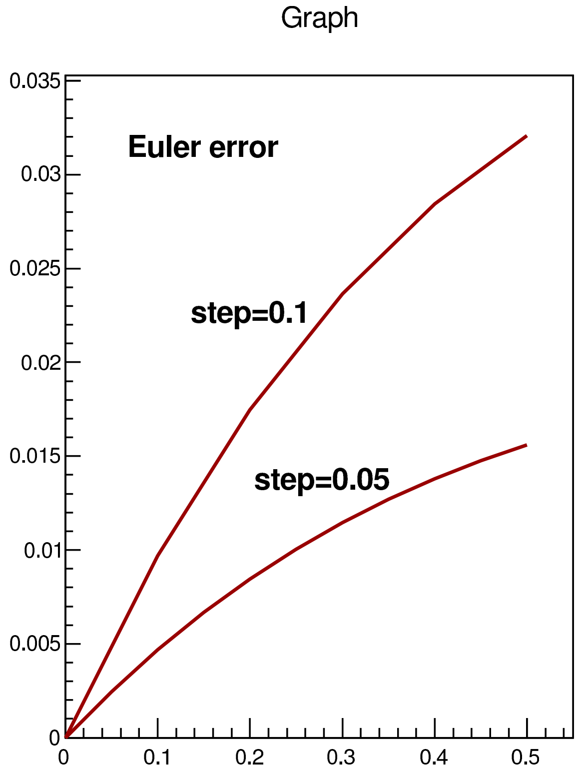
  $y(0) = 3.$
- analytical solution:

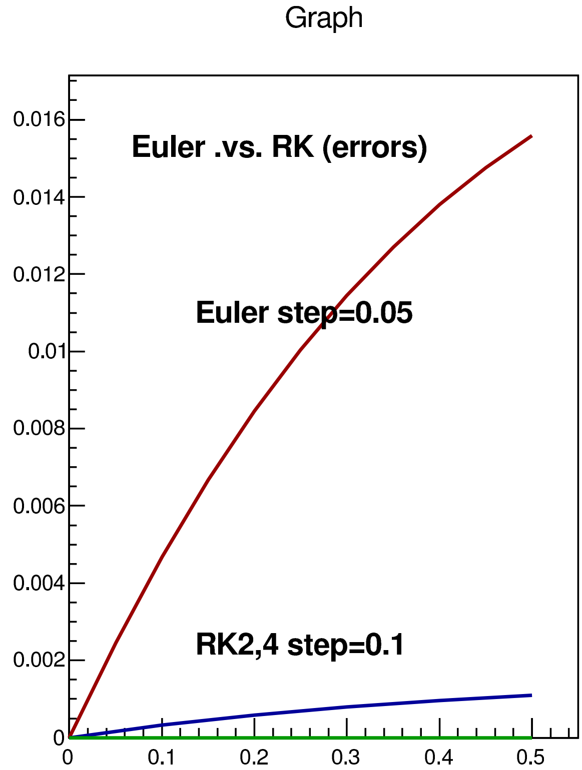
  $y(x) = 3x - 2e^{-x} + 5$
- Solving numerically with Euler method:

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



Example: ODE 1st-order (cont.)





Example: system of 1st-order ODEs

- ✔ Solve numerically the following system:

$$\frac{dw}{dx} = \sin(x) + y$$
$$\frac{dy}{dx} = \cos(x) - w$$

- ✔ Initial values:

$$w(0) = 0$$
$$y(0) = 0$$

Example: High-order equations

- High-order differential equations can be reduced to a systems of 1st-order equations
- These equations are very common in physics (dynamics):

$$m\frac{d^2\vec{r}}{dt^2} = \vec{F}(\vec{r}, t)$$

- The presence of frictional forces (or electromagnetic ones) introduce also a velocity dependence,

$$m\frac{d^2\vec{r}}{dt^2} = \vec{F}(\vec{r}, \dot{\vec{r}}, t)$$

- We can transform this equation into a set of 1st-order differential equations, in terms of variables $\vec{r}$ and $\vec{p}$:

$$\left\{\begin{array}{lcl} \frac{d\vec{r}}{dt} & = & \frac{\vec{p}(t)}{m} \\ \frac{d\vec{p}}{dt} & = & \vec{F}(t, \vec{r}, \dot{\vec{r}}) \end{array}\right. \Rightarrow \begin{bmatrix} \dot{r}_x \\ \dot{r}_y \\ \dot{r}_z \\ \dot{p}_x \\ \dot{p}_y \\ \dot{p}_z \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & \frac{1}{m} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{m} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{m} \\ \cdots & & & & & \\ \cdots & & & & & \\ \cdots & & & & & \end{bmatrix} \begin{bmatrix} r_x \\ r_y \\ r_z \\ p_x \\ p_y \\ p_z \end{bmatrix}$$

2nd order ODE: numerical solutions

- **Taylor method (2nd order) - Stormer-Verlet**
  using the numerical approximation for the 2nd-order derivative:

$$\left.\frac{d^2y}{dt^2}\right|_n \simeq \frac{y_{n+1} - 2y_n + y_{n-1}}{(\delta t)^2}$$

the differential equation becomes:

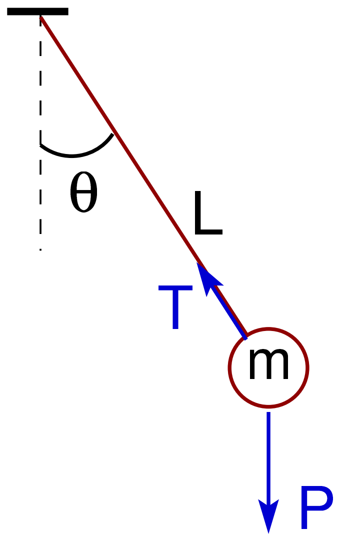
$$\boxed{\left.\frac{d^2y}{dt^2}\right|_n = f[t_n, y(t_n)] \quad \Rightarrow \quad y_{n+1} = -y_{n-1} + 2y_n + (\delta t)^2\ f[t_n, y(t_n)]}$$

- algorithm

time:	$\delta t = (t_f - t_0)/n$
initial conditions:	$t(0) \equiv t_0$
	$y(0) \equiv y_0$
	$\left.\frac{dy}{dt}\right\vert_{t=0} \equiv \left(\frac{dy}{dt}\right)_0$
first iteration:	$y_1 = y_0 + \delta t\ y'(t_0) + \frac{(\delta t)^2}{2}\ \underbrace{y''(t_0)}_{f(t_0, y(t_0))}$
following iterations (i=1,n-1):	$y_{i+1} = -y_{i-1} + 2y_i + (\delta t)^2\ f(t_i, y_i)$
	$t_{i+1} = t_0 + (i+1)\delta t$

simple pendulum

pendulum motion



Lagrangian: $\mathcal{L} = \frac{1}{2} m\ell^2 \dot{\theta}^2 + mg\ell\cos\theta$

eq. of motion: $\frac{d^2\theta}{dt^2} + \frac{g}{\ell}\sin\theta = 0$

initial conditions: $\theta(0) = \theta_0 \quad \dot{\theta}(0) = \omega_0$

simplify equation

characteristic time: $t_c = \sqrt{\frac{\ell}{g}}$

variable change: $\tau = \frac{t}{t_c}$

$$\boxed{\frac{d^2\theta}{d\tau^2} + \sin\theta = 0}$$

reduce to a system of 1st-order diff equations

$$\begin{cases} \frac{d\theta}{dt} = \omega \\ \frac{d\omega}{dt} = -\sin(\theta) \end{cases} \Rightarrow \begin{cases} \frac{dx_1}{dt} = & x_2 = f_1(t, x_1, x_2) \\ \frac{dx_2}{dt} = & -\sin(x_1) = f_2(t, x_1, x_2) \end{cases}$$

Solution methods:

- ☐ Euler
- ☐ Euler-Cromer
- ☐ Euler-Verlet
- ☐ Runge-Kutta

simple pendulum: solutions

Euler method

velocity is computed at the beginning of interval $(t, t + \Delta t)$

$$\begin{cases} x_{2,n+1} = x_{2,n} + (\Delta t)\ f_2(x_{1,n}) \\ x_{1,n+1} = x_{1,n} + (\Delta t)\ f_1(x_{2,n}) \end{cases}$$

Euler-Cromer method

coordinate uses velocity computed at the end of the interval $(t, t + \Delta t)$ improving behavior

$$\begin{cases} x_{2,n+1} = x_{2,n} + (\Delta t)\ f_2(x_{1,n}) \\ x_{1,n+1} = x_{1,n} + (\Delta t)\ f_1(x_{2,\mathbf{n+1}}) \end{cases}$$

Euler-Verlet method

it uses the 2nd-derivative operator $\ddot{y} = (\Delta t)^{-2}\ (y_{n+1} - 2y_n + y_{n-1}) + \mathcal{O}[(\Delta t)^2]$

$$\begin{cases} \theta_{n+1} = 2\ \theta_n - \theta_{n-1} + (\Delta t)^2\ f(\theta_n) \\ \omega_n = \frac{\theta_{n+1} - \theta_{n-1}}{2\ (\Delta t)} \end{cases}$$

first iteration $n = 0$
$$\begin{cases} \theta_1 = \theta_0 - (\Delta t)\ \omega_0 + \frac{(\Delta t)^2}{2}\ f(\theta_0) \\ \omega_0 \end{cases}$$

next iterations $n = 1, ..., N-1$
$$\begin{cases} \theta_2 = 2\theta_1 - \theta_0 + (\Delta t)^2\ f(\theta_1) \\ \omega_1 = \frac{\theta_2 - \theta_0}{2(\Delta t)} \end{cases}$$

simple pendulum: solutions (cont.)

Runge-Kutta method (4th order)

$$\left|\begin{aligned} K_{11} &= f_1(t_n,\ x_{1,n},\ x_{2,n}) \\ K_{12} &= f_2(t_n,\ x_{1,n},\ x_{2,n}) \end{aligned}\right.$$

$$\left|\begin{aligned} K_{21} &= f_1\left(t_n + \tfrac{h}{2},\ x_{1,n} + \tfrac{h}{2}K_{11},\ x_{2,n} + \tfrac{h}{2}K_{12}\right) \\ K_{21} &= f_2\left(t_n + \tfrac{h}{2},\ x_{1,n} + \tfrac{h}{2}K_{11},\ x_{2,n} + \tfrac{h}{2}K_{12}\right) \end{aligned}\right.$$

$$\left|\begin{aligned} K_{31} &= f_1\left(t_n + \tfrac{h}{2},\ x_{1,n} + \tfrac{h}{2}K_{21},\ x_{2,n} + \tfrac{h}{2}K_{22}\right) \\ K_{31} &= f_2\left(t_n + \tfrac{h}{2},\ x_{1,n} + \tfrac{h}{2}K_{21},\ x_{2,n} + \tfrac{h}{2}K_{22}\right) \end{aligned}\right.$$

$$\left|\begin{aligned} K_{41} &= f_1\left(t_n + h,\ x_{1,n} + hK_{31},\ x_{2,n} + hK_{32}\right) \\ K_{41} &= f_2\left(t_n + h,\ x_{1,n} + hK_{31},\ x_{2,n} + hK_{32}\right) \end{aligned}\right.$$

$$\boxed{\begin{aligned} x_{1,n+1} &= x_{1,n} + \tfrac{h}{6}\left(K_{11} + 2K_{21} + 2K_{31} + K_{41}\right) \\ x_{2,n+1} &= x_{2,n} + \tfrac{h}{6}\left(K_{12} + 2K_{22} + 2K_{32} + K_{42}\right) \end{aligned}}$$

projectile motion: multi-dimensions

- ✔ equation of motion

$$\frac{d^2\vec{r}}{dt^2} = -mg\vec{e}_y$$

- ✔ in terms of coordinates

$$\begin{aligned} \frac{d^2x}{dt^2} &= 0 \\ \frac{d^2y}{dt^2} &= -mg \end{aligned}$$

- ✔ system of 1st-order differential equations

$$\begin{aligned} \frac{dx}{dt} &= v_x \\ \frac{dy}{dt} &= v_y \\ \frac{dv_x}{dt} &= 0 \\ \frac{dv_y}{dt} &= -mg \end{aligned}$$

renaming variables

$$x, y, v_x, v_y \ \rightarrow\ x_1, x_2, x_3, x_4$$

$$\begin{aligned} \frac{dx_1}{dt} &= x_3 = & f_1(t, x_1, x_2, x_3, x_4) \\ \frac{dx_2}{dt} &= x_4 = & f_2(t, x_1, x_2, x_3, x_4) \\ \frac{dx_3}{dt} &= 0 = & f_3(t, x_1, x_2, x_3, x_4) \\ \frac{dx_4}{dt} &= -mg = & f_4(t, x_1, x_2, x_3, x_4) \end{aligned}$$

ODEsolver

- Runge–Kutta
- Euler
- Euler–Verlet

needs
- number variables
- functions

Library?

Physics Problem

eqs of motion

user problem

Electrostatic field lines

Let's consider N electric charges of charge Q_i located at fixed positions given by ther position vector $\vec{r}_i$, $i = 1, 2, ..., N$

The electric field produced at point a point P:

$$\vec{E}(\vec{r}) = \frac{1}{4\pi\varepsilon_0} \sum_{i=1}^{N} \frac{Q_i}{|\vec{r}-\vec{r}_i|^2} \vec{e}_{r_i}$$

The electric field components:

$$\vec{E}_x = \frac{1}{4\pi\varepsilon_0} \sum_{i=1}^{N} \frac{Q_i(x-x_i)}{[(x-x_i)^2+(y-y_i)^2]^{3/2}}$$
$$\vec{E}_y = \frac{1}{4\pi\varepsilon_0} \sum_{i=1}^{N} \frac{Q_i(y-x_i)}{[(x-x_i)^2+(y-y_i)^2]^{3/2}}$$

The field lines are curves whose tangent lines at every point are parallel to the electric field at the point. The discretized segments for step $\Delta\ell$:

$$\Delta_x = \Delta\ell \frac{E_x}{E}$$
$$\Delta_y = \Delta\ell \frac{E_y}{E}$$