

ODE

Very simplified algorithm of
ODE integration by odeint() function

Initial conditions $t=0$ and $y=2.0$

User defined
function

Start

Initial conditions
 $t_0=0, y(t_0)=2$
and t_1 are given

Program steps

```

1  import numpy as np
   from scipy.integrate import odeint

3  def ode(y, t):
    y = y[0]
    k = 1.4e-3
    dydt = -k*y
    return [dydt]

2  t0 = 0
   y0 = 2
   t1 = 18
   results = odeint(ode, [y0], [t0, t1])

4  y = results[:,0]
   y1=y[1]
    
```

```

def ode(y, t):
    y = y[0]
    k = 1.4e-3
    dydt = -k*y
    return [dydt]
    
```

Compute dy/dt iteratively
for t goes from t_0 to t_1 .
Number of steps and other
details depend on the
integration method
implemented.

odeint() body

End

odeint() returns a vector
with the solution:

$[y(t_0), y(t_1)]$

The solution for $t=18$: $y=1.95022976479$

Calculations done by odeint():

tn=0.00000	yn=2.00000	dydt(tn, yn)=-0.00280
tn=0.00220	yn=1.99999	dydt(tn, yn)=-0.00280
tn=0.00220	yn=1.99999	dydt(tn, yn)=-0.00280
tn=0.00439	yn=1.99999	dydt(tn, yn)=-0.00280
tn=0.00439	yn=1.99999	dydt(tn, yn)=-0.00280
tn=3.28786	yn=1.99082	dydt(tn, yn)=-0.00279
tn=3.28786	yn=1.99082	dydt(tn, yn)=-0.00279
tn=6.57133	yn=1.98168	dydt(tn, yn)=-0.00277
tn=6.57133	yn=1.98168	dydt(tn, yn)=-0.00277
tn=9.85481	yn=1.97260	dydt(tn, yn)=-0.00276
tn=9.85481	yn=1.97260	dydt(tn, yn)=-0.00276
tn=23.66719	yn=1.93482	dydt(tn, yn)=-0.00271
tn=23.66719	yn=1.93482	dydt(tn, yn)=-0.00271
tn=20.20912	yn=1.94421	dydt(tn, yn)=-0.00272
tn=20.20912	yn=1.94421	dydt(tn, yn)=-0.00272