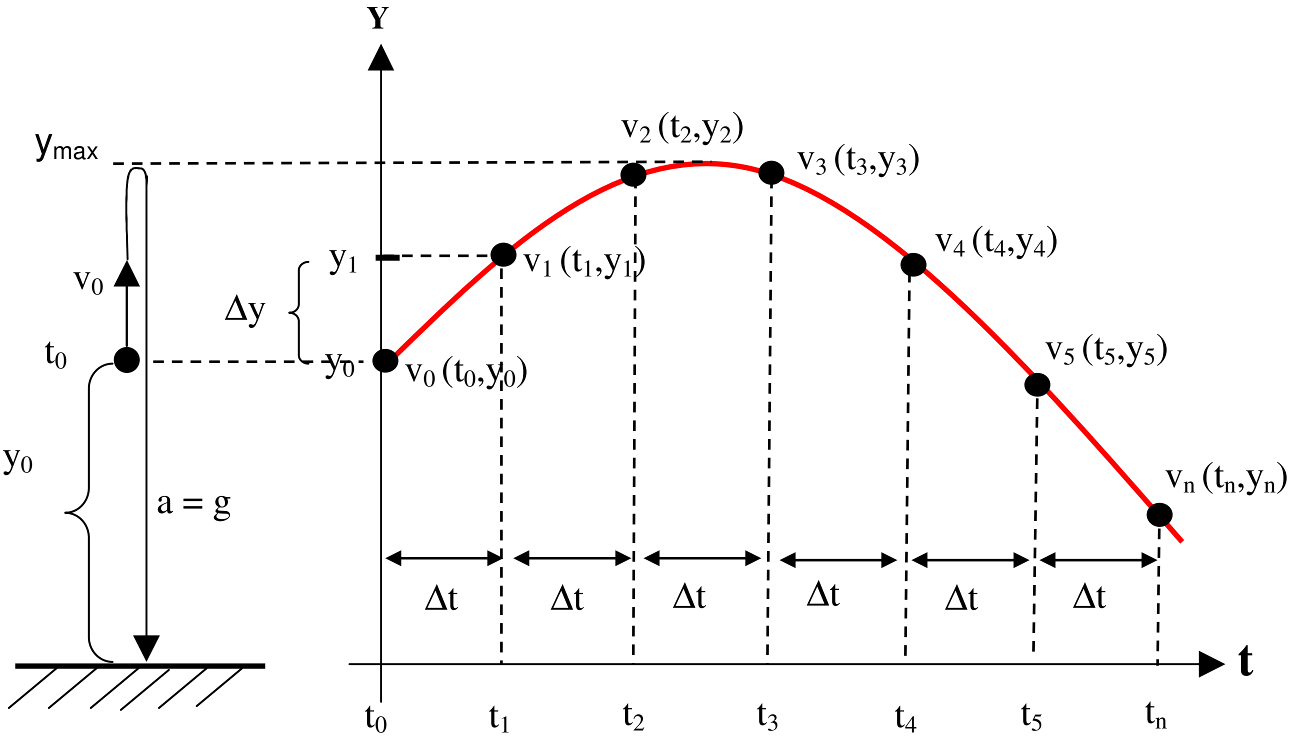
COMPUTATIONAL PHYSICS (SSP2122)

Eular method

by
Abd. Khamim Ismail
Mohd Khalid Kasmin

Free fall body problem

Consider an object thrown upward under gravitational force ($f=mg$) with an initial velocity, v_0. The object goes up and then stop at maximum *y* position, y_{max} before it start to fall down, back to the original position at y_0 and then to the ground level. (See figure).



$$F = ma = m\frac{dv}{dt} = m\frac{d^2y}{dt^2} = m\ddot{y}$$

Exact theoretical solution

$$y(t) = y_0 + v_0 t + \tfrac{1}{2} a t^2$$

The *y* projection of the exact solution can be plotted using the following java source code:

```
/* Java class for solving free fall object under gravity (g = 9.8 m/s/s)
   Exact solution : y = y0 + v0t + 1/2at^2
*/

public class freefallsol
{
  public static void main(String args[])
  {
  double a,v0,y,y0,delt,t;

  a  = -9.8;
  y0 = 1.0;
  v0 = 1.0;
  delt = 0.0001;
  for(t=0; t<=0.5; t=t+delt)
     {
      y   = y0  + v0*t + 0.5*a*t*t;
      System.out.printf("%8.4f  %8.4f\n",t,y);
     }
  }
}
```

Complile freefallsol.java to get the plot of the exact solution and then run and save the output into a file name as follows:

[khamim@setpar65 ~]$ java freefallsol >"ffsol.dat"

Euler Method

There is another way to see the projection of *y* position of the object using a numerical method called **"Euler method"**. See the following derivation:

$$v = \frac{\Delta y}{\Delta t}, \quad a = \frac{\Delta v}{\Delta t}$$

$$\Delta y = v\Delta t, \quad \Delta v = a\Delta t$$

$$v_1 = v_0 + \Delta v = v_0 + a\Delta t$$
$$y_1 = y_0 + \Delta y = y_0 + v_1\Delta t$$
$$v_2 = v_1 + \Delta v = v_1 + a\Delta t$$
$$y_2 = y_1 + \Delta y = y_1 + v_2\Delta t$$
$$\vdots$$
$$v_n = v_{n-1} + \Delta v = v_{n-1} + a\Delta t$$
$$y_n = y_n + \Delta y = y_{n-1} + v_n\Delta t$$

if **Δt $\rightarrow$ 0** then ***y*** will close to the exact solution ($y(t) = y_0 + v_0 t + \tfrac{1}{2} a t^2$)

The algorithm :

1. Declare the class/method to be used
2. Declare and assign all variables to be used; a,v0,v1,y,y0,y1,delt,t, and j.
3. Initialize value to all respective variables; a = -9.8, y_0 = 1.0, v_0 = 1.0;
4. Select appropriate value for Δt;
   $\Delta t = 0.005$
5. Start the loop to calculate the velocity and position for t=0
6. Calculate velocity; $v_1 = v_0 + \Delta v = v_0 + a\,\Delta t$
7. Calculate position; $y_1 = y_0 + \Delta y = y_0 + v_1\,\Delta t$
8. Print out values of t and y_1
9. Swap: $v_0 = v_1$ and $y_0 = y_1$;
10. Repeat from step 5 and increase the value of t by Δt (t =t + Δt) until stopping condition is satisfied at t=0.5

The java code :

```
/* Java class for solving free fall object under gravity (g = 9.8 m/s/s)
by numerical method called "Euler's Method"
*/
public class freefalleuler
{
  public static void main(String args[])
  {
  double a,v0,v1,y,y0,y1,delt,t;
  int j;

  a  = -9.8;
  y0 = 1.0;
  v0 = 1.0;
  delt = 0.005;

  for(t=0; t<=0.5; t =t + delt)
     {
      v1 = v0 + a*delt;
      y1 = y0 + v1*delt;
      System.out.printf("%8.4f %8.4f\n",t,y1);
      v0 = v1;
      y0 = y1;
     }
  }
}
```

Complile freefalleuler.java for different delt i.e 0.0005, 0.005, 0.05 and then run and save the output into a file name as follows:

```
[khamim@setpar65 ~]$ java freefalleuler >"ffeuler0.0005.dat"
[khamim@setpar65 ~]$ java freefalleuler >"ffeuler0.005.dat"
[khamim@setpar65 ~]$ java freefalleuler >"ffeuler0.05.dat"
```

Plotting the results using gnuplot :

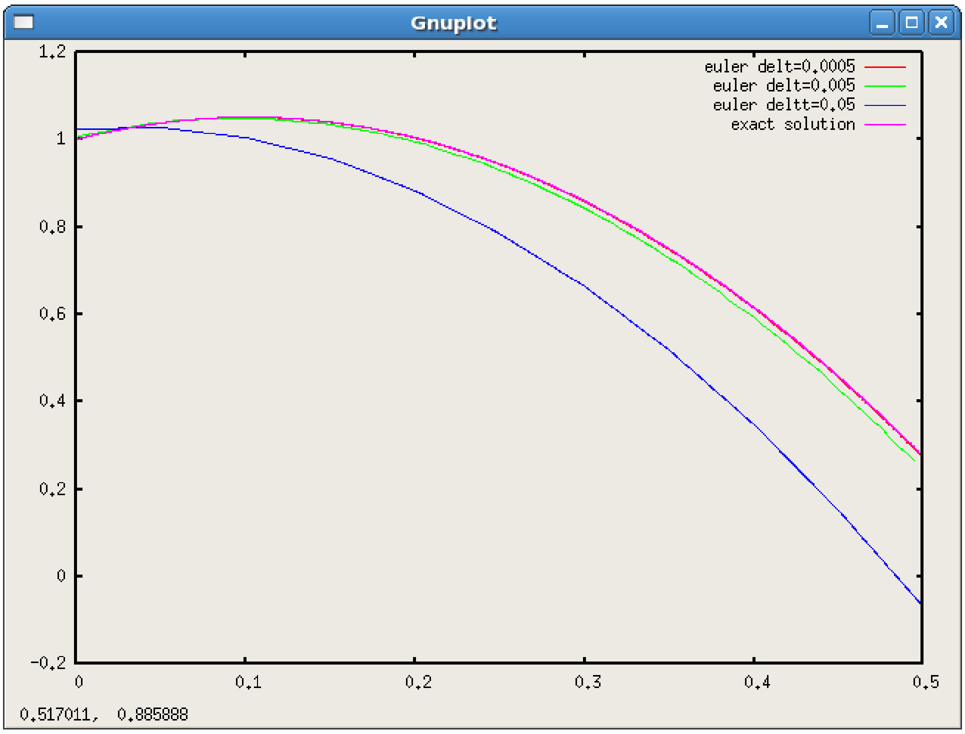
Using gnuplot, plot all the graphs using the following command and compare the results :

```
gnuplot> plot "ffeuler0.0005.dat" using 1:2 title "euler delt=0.0005" with    line,\
> "ffeuler0.005.dat" using 1:2 title "euler delt=0.005" with line,\
> "ffeuler0.05.dat" using 1:2 title "euler deltt=0.05" with line,\
> "ffsol.dat" using 1:2 title "exact solution" with line
```

In MS windows

```
gnuplot> plot "ffeuler0.0005.dat" using 1:2 title "euler delt=0.0005" with line,
"ffeuler0.005.dat" using 1:2 title "euler delt=0.005" with line,"ffeuler0.05.dat " using 1:2 title
"euler delt=0.05" with line, "ffsol.dat" using 1:2 title "exact solution" with line
```

Gnuplot



The smaller the delt the closer the curve to the exact solution.