COMPUTATIONAL PHYSICS (SSP2122)

Improved Eular method

by

Abd. Khamim Ismail

Mohd Khalid Kasmin

Improved Euler method

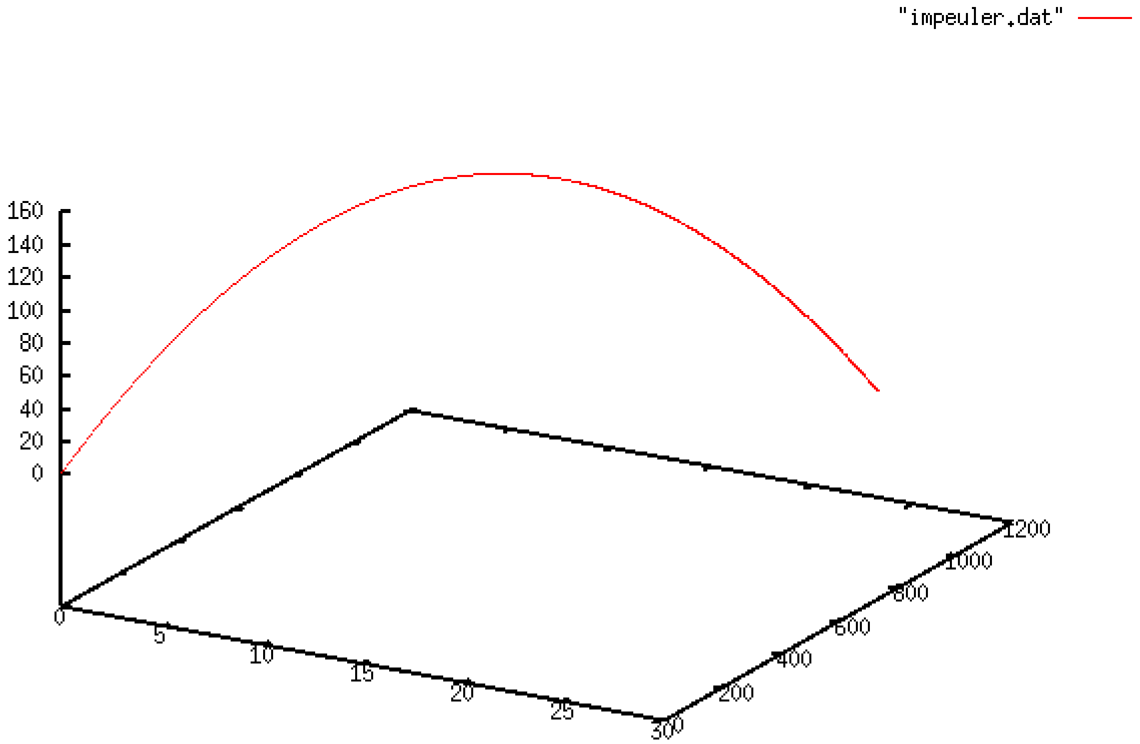
Comparison between analytical solution of F=ma with the Euler method, shows that although the method is able to find approximate solution, the error is quite significant if computational step is reduced. The error is also cumulative, i.e as the iteration of the calculation increases the error also gets bigger, hence value furthest away from initial condition has the largest error.

The Euler method gives us a way to approximate positions of object with respect to time, but requires an order of magnitude of computational iterations for each order of magnitude of accuracy. For large number of objects this method will certainly consume a great deal of computation time. Hence students are introduced to Initial Half-Step Method, an Euler method with a much improved computational effeciency.

Source code

```
/*
    Study of motion of an object with mass, m
    experiencing a constant force, F using
    Improved Euler Method (Initial half step method)

    In X and Y dimension
*/

import static java.lang.Math.*;
public class fma04 {
   public static void main(String args[]){

      double m, fx, fy, x, y, vo, vx, vy, ax, ay, t, delt;
      double pi, th, theta, ymax, tmax, xmax;

      pi = 4.0*atan(1.0);

      m = 0.1;     // in kg
      fy = -0.2;   // in N   (pulling down force)
      fx = 0.0;  // in N   (retarding force)

      ay = fy/m;
      ax = fx/m;
      vo = 50.0;    //in m/s
      theta = 30.0;  //in degrees

      th = theta*(pi/180.0); //degrees change to radian

      vx = vo * cos(th);  // in m/s
      vy = vo * sin(th);  // in m/s
      y = 1.2;            // in m
      x = 0.0;            // in m

      ymax=y; //initialize var ymax with initial value of y
      xmax=0.0;
      tmax=0.0;

      delt = 0.01;         // in s

      for(t=0.0; t<=30.0; t=t+delt){
         System.out.printf("%f   %f    %f\n",t,x,y);
         if (t==0.0){
            vx = vx + 0.5*ax*delt;
            vy = vy + 0.5*ay*delt;
            }
            else {
                vx = vx + ax*delt;
                vy = vy + ay*delt;
            }
            x = x + vx*delt;
            y = y + vy*delt;
            if (y>ymax) {
                ymax=y;
                xmax=x;
                tmax=t;
            }
            if (y<=0.0) break;
        }
        System.out.printf("#  Value of Y maximum is %f happened when t = %f \n",ymax,tmax);
        System.out.printf("#  at x = %f.\n",xmax);
        System.out.printf("#  The object reach the ground at x=%f and t=%f\n", x,t);
    }
}
```

gnuplot result:


"impeuler.dat"