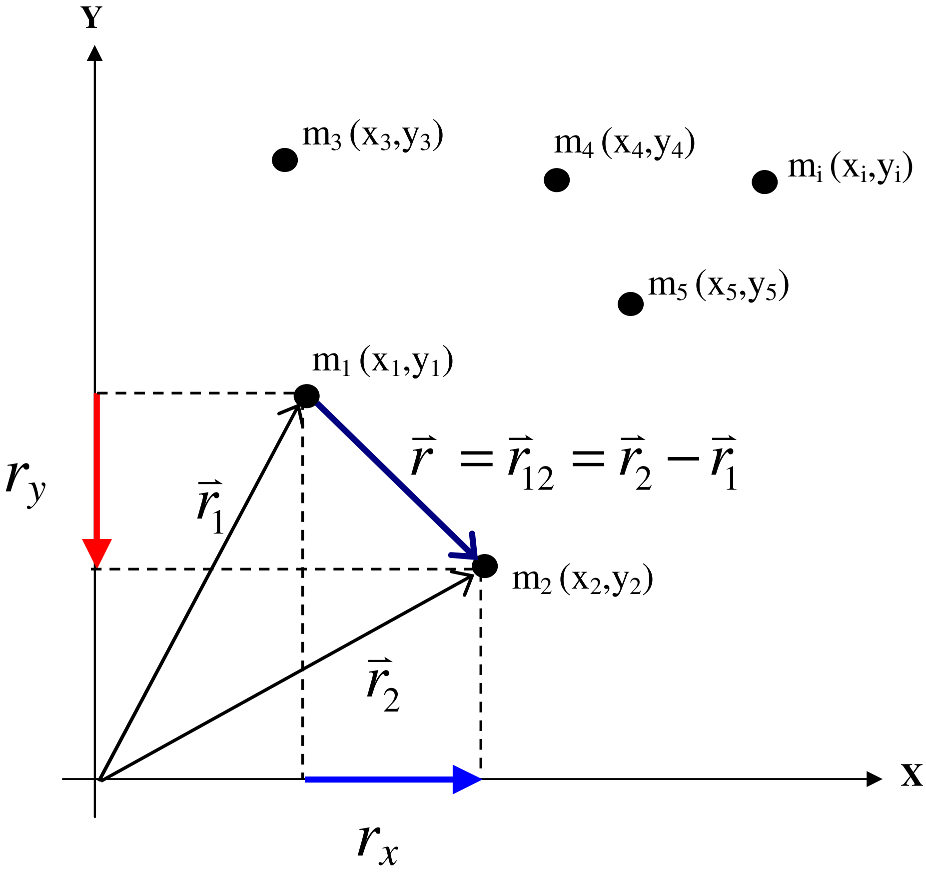
COMPUTATIONAL PHYSICS (SSP2122)

Calculation of gravitational force using array variable

by
Abd. Khamim Ismail
Mohd Khalid Kasmin

Example of usage of array variable. Calculation of gravitational force on an obejct due to many other neighboring objects. Array of m[ ] and r[ ]. Summations of F on m[i].



$$\vec{r}_1 = x_1\hat{x} + y_1\hat{y}$$

$$\vec{r}_2 = x_2\hat{x} + y_2\hat{y}$$

$$\vec{r} = \vec{r}_2 - \vec{r}_1 = (x_2 - x_1)\hat{x} + (y_2 - y_1)\hat{y}$$

$$r_x = (x_2 - x_1)$$

$$r_y = (y_2 - y_1)$$

$$\vec{r} = r_x\hat{x} + r_y\hat{y}$$

$$r = \sqrt{r_x^2 + r_y^2}$$

$$\vec{F}_{12} = \frac{Gm_1 m_2}{r^2}\hat{r}$$,

$$\hat{r} = \frac{\vec{r}}{r}$$, is a unit vector

$$\vec{F}_{12} = \frac{Gm_1 m_2}{r^3}\vec{r}$$

$$\vec{F}_{12} = \frac{Gm_1 m_2}{r^3}\left(r_x\hat{x} + r_y\hat{y}\right)$$

$$\vec{F}_{12} = \left(\frac{Gm_1 m_2}{r^3}r_x\right)\hat{x} + \left(\frac{Gm_1 m_2}{r^3}r_y\right)\hat{y}$$

$$\vec{F}_{12} = \left(\frac{Gm_1 m_2}{\left(r_x^2 + r_y^2\right)^{3/2}}r_x\right)\hat{x} + \left(\frac{Gm_1 m_2}{\left(r_x^2 + r_y^2\right)^{3/2}}r_y\right)\hat{y}$$

$$r_p = r^3 = \left(\sqrt{r_x^2 + r_y^2}\right)^3 = \left(r_x^2 + r_y^2\right)^{3/2}$$

$$\text{co} = Gm_1 m_2$$

$$\vec{F}_{12} = \left(\frac{\text{co}}{r_p}r_x\right)\hat{x} + \left(\frac{\text{co}}{r_p}r_y\right)\hat{y}$$

Components of vector;

$$F_{x_{12}} = \left(\frac{\text{co}}{r_p} r_x \right), \qquad F_{y12} = \left(\frac{\text{co}}{r_p} r_y \right)$$

Magnitude of $\vec{F}_{12}$;

$$F_{12} = \sqrt{\left(F_{x_{12}}\right)^2 + \left(F_{y12}\right)^2}$$

Direction;

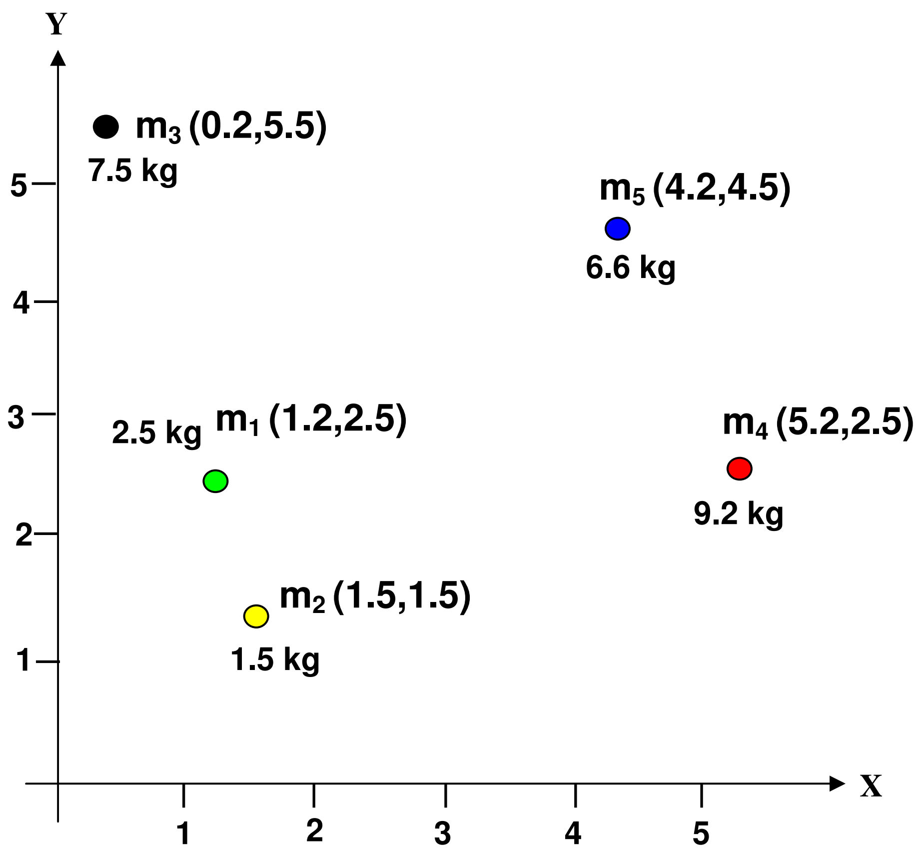
$$\theta = \tan^{-1} \frac{F_{y12}}{F_{x_{12}}}$$

For ***n*** total number of particles;

$$\vec{F}_{1_{net}} = \vec{F}_{12} + \vec{F}_{13} + \vec{F}_{14} + \vec{F}_{15} + \ldots\ldots\ldots\ldots \vec{F}_{1n}$$

$$\vec{F}_{1_{net}} = \sum_{i=2}^{n} \vec{F}_{1i}$$

$$F_{x_{1net}} = \sum_{i=2}^{n} F_{x_{1i}}, \qquad F_{y_{1net}} = \sum_{i=2}^{n} F_{y_{1i}}$$



Given m_1 to m_5 and their coordinates are as above. Calculate the gravitational force at m_1 due to m_2, m_3, m_4 and m_5.

```
m[1]=2.5; x[1]=1.2; y[1]=2.5;
m[2]=1.5; x[2]=1.5; y[2]=1.5;
m[3]=7.5; x[3]=0.2; y[3]=5.5;
m[4]=9.2; x[4]=5.2; y[4]=2.5;
m[5]=6.6; x[5]=4.2; y[5]=4.5;
```

The algorithm:

1. Declare the class/method to be used
2. Declare and assign all variables to be used
3. Initialize value to all respective variables
4. Decide on which mass the gravitational force to be calculated (e.g on m_j where j=1)
5. Set f_x=0 and f_y=0
6. Start the loop from j=2 to j=5 to calculate the summation of f_x and f_y
7. Resolve the gravitational force into unit vector (2D Cartesian coordinate)
8. Calculate the mass vector position: $$\vec{r} = \vec{r}_2 - \vec{r}_1 = (x_2 - x_1)\hat{x} + (y_2 - y_1)\hat{y}$$

   Therefore: $r_x = x[j]-x[1]$ and $r_y = y[j]-y[1]$
9. Define/assign all respective equations to be used:

$$\vec{F} = \left(\frac{Gm_1 m_2}{\left(r_x^2 + r_y^2\right)^{3/2}} r_x \right)\hat{x} + \left(\frac{Gm_1 m_2}{\left(r_x^2 + r_y^2\right)^{3/2}} r_y \right)\hat{y}$$

$$r_p = r^3 = \left(\sqrt{r_x^2 + r_y^2}\right)^3 = \left(r_x^2 + r_y^2\right)^{3/2}$$

$$\mathrm{co} = Gm_1 m_2$$

$$\vec{F} = \left(\frac{\mathrm{co}}{r_p} r_x \right)\hat{x} + \left(\frac{\mathrm{co}}{r_p} r_y \right)\hat{y}$$

10. Calculate the summation of f_x and f_y until m_j (j=2 to j=5) :

    $f_x = f_x + \mathrm{co}\ (r_x/r_p)$

    $f_y = f_y + \mathrm{co}(r_y/r_p)$

11. Calculate the magnitude of the gravitational force: $F = \sqrt{(F_x)^2 + (F_y)^2}$

12. Calculate the direction of F; $\theta = \tan^{-1} \dfrac{F_y}{F_x}$

13. Print the results

Source code GravityAttract05a.java

```
/*     Java class for calculating the resultant force on an object m1
       due to for other objects m2, m3, m4, and m5
  */

  import static java.lang.Math.*;

  public class GravityAttract05a
  {
     public static void main(String args[])
     {
        double[] m = new double[6]; // mass of object
        double[] x = new double[6]; // x-coor of object
        double[] y = new double[6]; // y-coor of object
        double f;                   // resultant force
        double fx,fy;                // x & y component of force
        double rx, ry, rp, co, G;
        int j;

        G  = 6.67e-11;
        m[1]=2.5; x[1]=1.2; y[1]=2.5;
        m[2]=1.5; x[2]=1.5; y[2]=1.5;
        m[3]=7.5; x[3]=0.2; y[3]=5.5;
        m[4]=9.2; x[4]=5.2; y[4]=2.5;
        m[5]=6.6; x[5]=4.2; y[5]=4.5;

        fx = 0.0;
        fy = 0.0;
        for(j=2; j<=5; j++)
        {
           rx = x[j]-x[1];
           ry = y[j]-y[1];
           rp = pow((rx*rx+ry*ry),1.5); // (rx^2 + ry^2) raised to the power of 3/2
           co = G*m[1]*m[j];
           fx = fx + co*rx/rp;
           fy = fy + co*ry/rp;
        }
        f = sqrt(fx*fx + fy*fy);
        System.out.printf("\n\tThe force on m[1] due to the other masses is %e N\n\n",f);
     }
  }
```

Source code GravityAttract06a.java

```
/*     Java class for calculating the resultant force on each object
       (on m1 through m5)
  */

  import static java.lang.Math.*;

  public class GravityAttract06a
  {
     public static void main(String args[])
     {
        double[] m = new double[6]; // mass of object
        double[] x = new double[6]; // x-coor of object
        double[] y = new double[6]; // y-coor of object
        double f;                   // resultant force
        double fx,fy;               // x & y component of force
        double rx, ry, rp, co, G, dirf,angdirf;
        int j, mi;

        G  = 6.67e-11;
        m[1]=2.5; x[1]=1.2; y[1]=2.5;
        m[2]=1.5; x[2]=1.5; y[2]=1.5;
        m[3]=7.5; x[3]=0.2; y[3]=5.5;
        m[4]=9.2; x[4]=5.2; y[4]=2.5;
        m[5]=6.6; x[5]=4.2; y[5]=4.5;

        for (mi=1; mi<=5; mi++)
           {
           fx = 0.0;
           fy = 0.0;
           for(j=1; j<=5; j++)
           {
              if (j!=mi)
               {
               rx = x[j]-x[mi];
               ry = y[j]-y[mi];
               rp = pow((rx*rx+ry*ry),1.5); // (rx^2 + ry^2) raised to the power of 3/2
               co = G*m[mi]*m[j];
               fx = fx + co*rx/rp;
               fy = fy + co*ry/rp;
               }
           }
           f = sqrt(fx*fx + fy*fy);
           dirf = atan(fy/fx);
           angdirf = (dirf/(22.0/7.0))*180;
           System.out.printf("The force on m["+ mi +"] due to the other masses is %eN, %fdeg\n",f,angdirf);
          }
     }
  }
```

Source code GravityAttract07a.java

```
/*     Java class for calculating the resultant force on any object
       (i.e either on m1,m2, m3, m4, or m5)
  */

  import static java.lang.Math.*;
  import java.io.*;
  public class GravityAttract07a
  {
     public static void main(String args[])
     {
        double[] m = new double[6]; // mass of object
        double[] x = new double[6]; // x-coor of object
        double[] y = new double[6]; // y-coor of object
        double f;                    // resultant force
        double fx,fy;                // x & y component of force
        double rx, ry, rp, co, G, dirf,angdirf;
        int j ;

        G  = 6.67e-11;
        m[1]=2.5; x[1]=1.2; y[1]=2.5;
        m[2]=1.5; x[2]=1.5; y[2]=1.5;
        m[3]=7.5; x[3]=0.2; y[3]=5.5;
        m[4]=9.2; x[4]=5.2; y[4]=2.5;
        m[5]=6.6; x[5]=4.2; y[5]=4.5;
        BufferedReader dataIn=new BufferedReader(new InputStreamReader(System.in));
        String mass="";
        System.out.println("On which mass you want to calculate the net force(1,2,3,4 or 5)?");
        try { mass=dataIn.readLine();}
        catch(IOException e){System.out.println("Not a valid input value!!");}
        int mi = Integer.parseInt(mass);
        fx = 0.0;
        fy = 0.0;
           for(j=1; j<=5; j++)
           {
              if (j!=mi)
               {
               rx = x[j]-x[mi];
               ry = y[j]-y[mi];
               rp = pow((rx*rx+ry*ry),1.5); // (rx^2 + ry^2) raised to the power of 3/2
               co = G*m[mi]*m[j];
               fx = fx + co*rx/rp;
               fy = fy + co*ry/rp;
               }
           }
         f = sqrt(fx*fx + fy*fy);
         dirf = toDegrees(atan(fy/fx));
         System.out.printf("The force on m["+ mi +"] due to the other masses is %eN, %fdeg\n",f,dirf);

     }
  }
```