Appendix VII

Java Code for Calculating Solid Angle Coverage

Marco Arosio

```
package ma.CosmicRayDetector;

import java.io.*;
import java.util.*;

public class SolidAngle {

public SolidAngle(double xvalue, double yvalue, double zvalue)
                                                    { x = xvalue; y = yvalue;
z=zvalue;}

// to generate random vectors:

    private double x;
    private double y;
     private double z;

    public double getxcomponent(){return x;}
    public double getycomponent(){return y;}
    public double getzcomponent(){return z;}

    public static SolidAngle vectorGenerator (){double theta
=Math.PI*Math.random();
                                                    double phi =
2*Math.PI*Math.random();
                                                    double ax =
Math.sin(theta)*Math.cos(phi);
                                                    double a y =
Math.sin(theta)*Math.sin(phi);
                                                    double az =Math.cos(theta);
                                                   SolidAngle randomvector= new
SolidAngle(ax,ay,az);
                                                    return randomvector;
    }


   // to generate parametric lines of the form r= P +At:

    static double d;
    static double px;
    static double py;

   public SolidAngle lineGenerator(){double t=d/vectorGenerator().getzcomponent();
                                        double rx = px +
vectorGenerator().getxcomponent()*t;
                                        double ry = py +
vectorGenerator().getycomponent()*t;
```

```
                                        SolidAngle randomline =new
SolidAngle(rx,ry,d);
                                        return randomline;
    }

// to check if the lines are intersecting the area of the top insulator:

   static double insulatorLength;
   static double insulatorWidth;

   public boolean intersection(){
                if((x < insulatorLength && x > 0)&&(y < insulatorWidth && y >
0)){return true;}
                else{return false;}
    }


  // to calculate the solid angle covered:

    static double solidAngleCovered(double totalNumberOfParticles, double
NoOfIntersections){
                        double solid
=4*Math.PI*(NoOfIntersections/totalNumberOfParticles);
                         return solid;
    }


    public static void main(String[] args) {

        System.out.println("SOLID ANGLE COVERAGE OF COSMIC RAY
DETECTOR");

        System.out.println("\n");


    String valuea="";
    final boolean enteringData=true;
    final double RETURN = 10;

    System.out.println("Enter insulator length: ");

    while(enteringData) {
    try{
       int length = System.in.read();
       if(length==RETURN) break;
       char charlength= (char) length;
       valuea +=charlength;}

      catch(java.io.IOException e){System.out.println("Error reading from
keyboard");}
```

```
}


  StringTokenizer st1 =new StringTokenizer(valuea);
  String s1 =st1.nextToken();
  Double d1 = new Double(s1);
  insulatorLength= d1.doubleValue();
System.out.println("the typed value is "+insulatorLength);
System.out.println("\n");

String valueb="";
System.out.println("Enter insulator width:");

while(enteringData) {
try{
    int width = System.in.read();
    if(width==RETURN) break;
    char charwidth = (char) width;
    valueb += charwidth;
  }

catch(java.io.IOException e){System.out.println("Error reading from keyboard");}
}

  StringTokenizer st2 =new StringTokenizer(valueb);
  String s2 =st2.nextToken();
  Double d2 = new Double(s2);
  insulatorWidth= d2.doubleValue();
  //System.out.println("the typed value is "+insulatorWidth);


System.out.println("\n");

String valuec="";
System.out.println("Enter insulator gap:");

while(enteringData) {
try{
    int gap = System.in.read();
    if(gap==RETURN) break;
    char chargap = (char) gap;
    valuec += chargap;
  }

catch(java.io.IOException e){System.out.println("Error reading from keyboard");}
}

  StringTokenizer st3 =new StringTokenizer(valuec);
  String s3 =st3.nextToken();
  Double d3 = new Double(s3);
```

```
      d = d3.doubleValue();
      //System.out.println("the typed value is "+d);


    System.out.println("\n");

    int j=0;
    int h=0;
    int k=0;

    for(px=0; px < insulatorLength; px = px+0.1) {
          for(py=0; py < insulatorWidth; py = py+0.1){
                        for(int m=0; m<100;m=m+1){
                                    j = j+1;
                                    SolidAngle myrandomvector =
vectorGenerator();
                                    SolidAngle myrandomline =
myrandomvector.lineGenerator();
                                    if(myrandomline.intersection()){h=h+1;}
                                    else{k=k+1;}

       }
    }


    }

   System.out.println("Number of scattered random particles: "+j);
    System.out.println(h+" particles went through the top insulator");
    System.out.println(k+" particles missed the top insulator");

   System.out.println("\n");

   System.out.println("The solid angle covered by these dimensions is "+
solidAngleCovered(j,h));
}
}
```

Appendix VIII
SLAC Data

Manuel Kurdian

The data shown below were used for the calculation of the solid angle subtended by the SLAC detector in section 4.2. The data readings are for 20 intervals of 27 minutes between Wed, 12 Feb 2003 09:00:00 and Wed, 12 Feb 2003 18:00:00 (PDT).
This data was acquired from the website:
http://www2.slac.stanford.edu/vvc/cosmicrays/crdatacenter.htm1

Table of Flux vs. Time

Interval Number	Date/Time	Flux (counts/minute/cm^2/sterad)		
		vertical (90^0)	45 degrees	horizontal (0^0)
1	Wed, 12 Feb 2003 09:13:00	0.601 +/- 0.028	0.283 +/- 0.019	0.018 +/- 0.005
2	Wed, 12 Feb 2003 09:40:00	0.576 +/- 0.027	0.262 +/- 0.018	0.026 +/- 0.006
3	Wed, 12 Feb 2003 10:07:00	0.606 +/- 0.028	0.264 +/- 0.018	0.037 +/- 0.007
4	Wed, 12 Feb 2003 10:34:00	0.655 +/- 0.029	0.268 +/- 0.019	0.037 +/- 0.007
5	Wed, 12 Feb 2003 11:01:00	0.525 +/- 0.026	0.273 +/- 0.019	0.031 +/- 0.006
6	Wed, 12 Feb 2003 11:28:00	0.556 +/- 0.027	0.270 +/- 0.019	0.025 +/- 0.006
7	Wed, 12 Feb 2003 11:55:00	0.585 +/- 0.028	0.238 +/- 0.018	0.035 +/- 0.007
8	Wed, 12 Feb 2003 12:22:00	0.589 +/- 0.028	0.279 +/- 0.019	0.045 +/- 0.008
9	Wed, 12 Feb 2003 12:49:00	0.613 +/- 0.028	0.257 +/- 0.018	0.025 +/- 0.006
10	Wed, 12 Feb 2003 13:16:00	0.543 +/- 0.026	0.258 +/- 0.018	0.031 +/- 0.006
11	Wed, 12 Feb 2003 13:43:00	0.531 +/- 0.026	0.291 +/- 0.019	0.035 +/- 0.007
12	Wed, 12 Feb 2003 14:10:00	0.548 +/- 0.027	0.262 +/- 0.018	0.039 +/- 0.007
13	Wed, 12 Feb 2003 14:37:00	0.606 +/- 0.028	0.304 +/- 0.020	0.034 +/- 0.007
14	Wed, 12 Feb 2003 15:04:00	0.601 +/- 0.028	0.277 +/- 0.019	0.034 +/- 0.007
15	Wed, 12 Feb 2003 15:31:00	0.562 +/- 0.027	0.264 +/- 0.018	0.045 +/- 0.008
16	Wed, 12 Feb 2003 15:58:00	0.532 +/- 0.026	0.256 +/- 0.018	0.041 +/- 0.007
17	Wed, 12 Feb 2003 16:25:00	0.616 +/- 0.028	0.278 +/- 0.019	0.028 +/- 0.006
18	Wed, 12 Feb 2003 16:52:00	0.587 +/- 0.028	0.293 +/- 0.019	0.032 +/- 0.006
19	Wed, 12 Feb 2003 17:19:00	0.549 +/- 0.027	0.279 +/- 0.019	0.039 +/- 0.007
20	Wed, 12 Feb 2003 17:46:00	0.598 +/- 0.028	0.258 +/- 0.018	0.040 +/- 0.007

Table of Rate vs Time

Interval Number	Date/Time	Rate (counts/minute)		
		vertical (90^0)	45 degrees	horizontal (0^0)
1	Wed, 12 Feb 2003 09:13:00	17.222 +/- 0.799	8.111 +/- 0.548	0.519 +/- 0.139
2	Wed, 12 Feb 2003 09:40:00	16.519 +/- 0.782	7.519 +/- 0.528	0.741 +/- 0.166
3	Wed, 12 Feb 2003 10:07:00	17.370 +/- 0.802	7.556 +/- 0.529	1.074 +/- 0.199
4	Wed, 12 Feb 2003 10:34:00	18.778 +/- 0.834	7.667 +/- 0.533	1.074 +/- 0.199
5	Wed, 12 Feb 2003 11:01:00	15.037 +/- 0.746	7.815 +/- 0.538	0.889 +/- 0.181
6	Wed, 12 Feb 2003 11:28:00	15.926 +/- 0.768	7.741 +/- 0.535	0.704 +/- 0.161
7	Wed, 12 Feb 2003 11:55:00	16.778 +/- 0.788	6.815 +/- 0.502	1.000 +/- 0.192
8	Wed, 12 Feb 2003 12:22:00	16.889 +/- 0.791	8.000 +/- 0.544	1.296 +/- 0.219
9	Wed, 12 Feb 2003 12:49:00	17.556 +/- 0.806	7.370 +/- 0.522	0.704 +/- 0.161
10	Wed, 12 Feb 2003 13:16:00	15.556 +/- 0.759	7.407 +/- 0.524	0.889 +/- 0.181
11	Wed, 12 Feb 2003 13:43:00	15.222 +/- 0.751	8.333 +/- 0.556	1.000 +/- 0.192
12	Wed, 12 Feb 2003 14:10:00	15.704 +/- 0.763	7.519 +/- 0.528	1.111 +/- 0.203
13	Wed, 12 Feb 2003 14:37:00	17.370 +/- 0.802	8.704 +/- 0.568	0.963 +/- 0.189
14	Wed, 12 Feb 2003 15:04:00	17.222 +/- 0.799	7.926 +/- 0.542	0.963 +/- 0.189
15	Wed, 12 Feb 2003 15:31:00	16.111 +/- 0.772	7.556 +/- 0.529	1.296 +/- 0.219
16	Wed, 12 Feb 2003 15:58:00	15.259 +/- 0.752	7.333 +/- 0.521	1.185 +/- 0.210
17	Wed, 12 Feb 2003 16:25:00	17.667 +/- 0.809	7.963 +/- 0.543	0.815 +/- 0.174
18	Wed, 12 Feb 2003 16:52:00	16.815 +/- 0.789	8.407 +/- 0.558	0.926 +/- 0.185
19	Wed, 12 Feb 2003 17:19:00	15.741 +/- 0.764	8.000 +/- 0.544	1.111 +/- 0.203
20	Wed, 12 Feb 2003 17:46:00	17.148 +/- 0.797	7.407 +/- 0.524	1.148 +/- 0.206

The tables show the interval number in the first column, the date and time corresponding to the (approximate) centre of the interval in the second column.