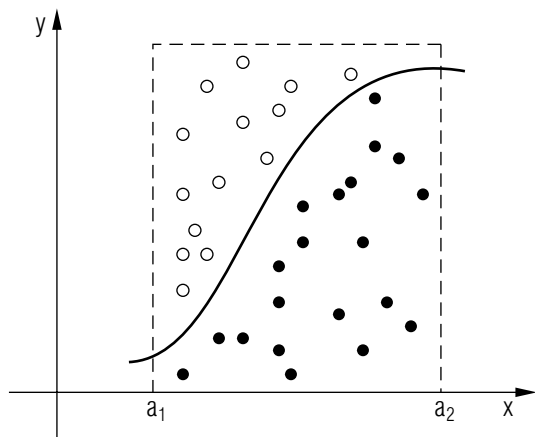# PHYSNET

MISN-0-355

MONTE CARLO EVALUATION OF DEFINITE INTEGRALS

Project PHYSNET Physics Bldg. Michigan State University East Lansing, MI

MONTE CARLO EVALUATION OF DEFINITE INTEGRALS

by

Robert Ehrlich

Title: **Monte Carlo Evaluation of Definite Integrals**

Author: R. Ehrlich, Physics Dept., George Mason Univ., Fairfax, VA 22030; (703)323-2303

Version: 3/20/2002   Evaluation: Stage 0

Length: 2 hr; 20 pages

Input Skills:

1. Vocabulary: Simpson's rule (MISN-0-349); center of mass coordinates (MISN-0-6); moment of inertia (MISN-0-41); probability density function (MISN-0-354).
2. Enter and use a computer program which generates random numbers (MISN-0-354).

Output Skills (Knowledge):

K1. Vocabulary: hypercube, hypersphere, moments of a probability distribution, Monte Carlo method, surface integral, volume integral.

K2. State the advantages and disadvantages of the Monte Carlo method.

Output Skills (Project):

P1. Run a program to calculate the mass and center of mass coordinates for a sphere with a hole through its center, and analyze how the results vary with changes in the N randomly located points, and with the size of N.

P2. Modify the program in P1 to find the mass and center of mass coordinates for other shaped objects.

P3. Modify the program in P1 to integrate two functions over the surface of a sphere of unit radius.

External Resources (Required):

1. A computer with FORTRAN or BASIC.

THIS IS A DEVELOPMENTAL-STAGE PUBLICATION
OF PROJECT PHYSNET

The goal of our project is to assist a network of educators and scientists in transferring physics from one person to another. We support manuscript processing and distribution, along with communication and information systems. We also work with employers to identify basic scientific skills as well as physics topics that are needed in science and technology. A number of our publications are aimed at assisting users in acquiring such skills.

Our publications are designed: (i) to be updated quickly in response to field tests and new scientific developments; (ii) to be used in both classroom and professional settings; (iii) to show the prerequisite dependencies existing among the various chunks of physics knowledge and skill, as a guide both to mental organization and to use of the materials; and (iv) to be adapted quickly to specific user needs ranging from single-skill instruction to complete custom textbooks.

New authors, reviewers and field testers are welcome.

PROJECT STAFF

Andrew Schnepp	Webmaster
Eugene Kales	Graphics
Peter Signell	Project Director

ADVISORY COMMITTEE

D. Alan Bromley	Yale University
E. Leonard Jossem	The Ohio State University
A. A. Strassenburg	S. U. N. Y., Stony Brook

MONTE CARLO EVALUATION
OF DEFINITE INTEGRALS

by

Robert Ehrlich

1. Integrals, 1-2 Dimensions

1a. Introduction. This module describes the Monte Carlo method for the evaluation of some definite integrals. The exotic name of the method arises from the use of random numbers in its execution. Briefly, the Monte Carlo method for evaluating a definite integral involves determining the fraction of a large number of randomly located points which lie under the curve of the function to be integrated. This idea is illustrated in Fig. 1(a) and (b), which is a graph of an arbitrary function $f(x)$ which we wish to integrate between the limits $x = a_1$, and $x = a_2$. The integral is equal to the shaded area under the curve in Fig. 1(a). The Monte Carlo approach is indicated in Fig. 1(b) where a number of random points are depicted inside the dotted rectangle. Let:

A = area of dotted rectangle
N = number of randomly located points inside the rectangle
n = number of random points which lie under curve
(shown blackened)

The Monte Carlo approximation to the area under the curve can then be obtained from:
$$a = fA$$

where $f = n/N$ is the fraction of all points that lie under the curve. Clearly the approximation improves as the number of random points increases. Typically the method is implemented on a computer by generating pairs of pseudorandom numbers for the x,y coordinates of each point.

1b. Area Inside a Closed Curve. The area inside a closed curve can likewise be approximated by enclosing the curve by a rectangle and computing the fraction of random points inside the rectangle which lie inside the curve. This area is in effect equal to the integral of the function $f_2(x) - f_1(x)$ where $f_1(x)$ and $f_2(x)$ are the upper and lower portions of the closed curve in Fig. 2.
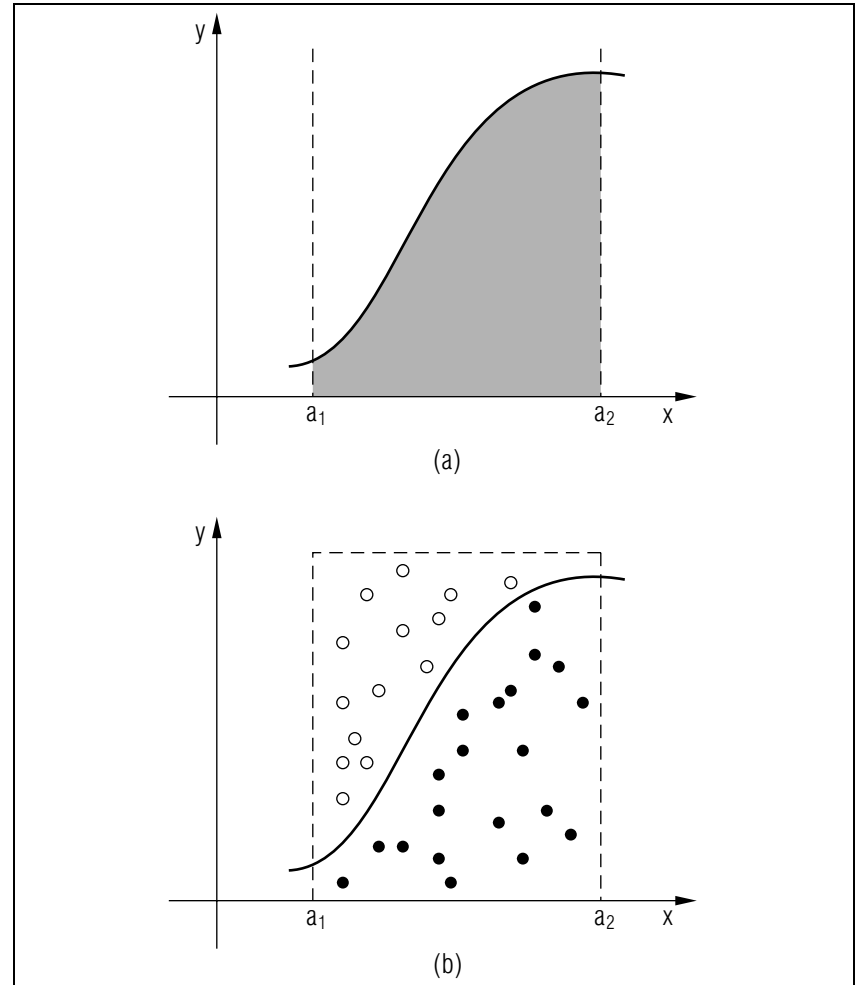
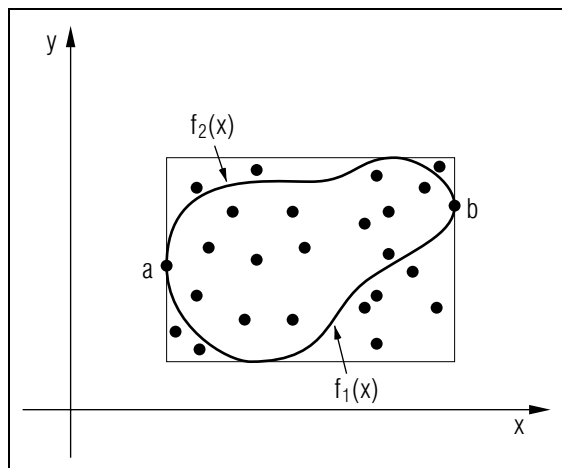**Figure 1.** Evaluating an integral using the Monte Carlo method.

Figure 2. Evaluating the area inside a closed curve using the Monte Carlo method.

1c. Surface Integrals In A Plane. The surface integral of a function of two variables $f(x, y)$ over an area S may be written as follows:

$$Q = \int_S f(x, y)\, dS. \tag{1}$$

Surface integrals, however, are two-dimensional integrals and actually involve successive integration over the two variables:

$$Q = \int \int_S f(x, y)\, dx\, dy. \tag{2}$$

We may approximate Eq. (1) by the finite sum:

$$Q = \sum_{j=1}^{N} f_j(x, y)\, \Delta S, \tag{3}$$

where the element of area is given by $\Delta S = S_0/N$, S_0 being the total area of the domain of integration (the rectangle in Fig. 2). To evaluate Eq. (3) using the Monte Carlo method we simply generate N randomly located points inside the rectangle in Fig. 2. Let $(x, y)_j$ be the coordinates of the jth randomly located point. If the point lies inside the closed curve we would add the contribution $f_j(x, y)\Delta S$ to the sum in Eq. (3); if the random point lies outside the closed curve no addition is made to the sum. In the special case: $f(x, y) = 1$, we would find that Eq. (3) yields the area inside the closed curve, exactly as was the case in the one-dimensional integral.

1d. Surface Integrals Over Non-Planar Surfaces. We may use the Monte Carlo method to approximate a surface integral over any surface S:

$$Q = \int_S f(x, y, z)\, ds. \tag{4}$$

As before Eq. (4) can be approximated by a finite sum:

$$Q = \sum_{j=1}^{N} f_j(x, y, z)\, \Delta S. \tag{5}$$

A difference from the integral over a plane surface is that now the integrand may be a function of three variables. In addition, the N random points need to be chosen at random locations over the surface of integration S which is non-planar. An example will help clarify the procedure. Suppose we wish to integrate a function over the surface of a sphere of radius R. The function to be integrated may be expressed either in rectangular coordinates x, y, z or polar coordinates R, θ, ϕ, where θ is the polar angle and ϕ the azimuth. The location of random points on the surface of a sphere can be most conveniently specified in terms θ and ϕ. In order that the random points be uniformly distributed over the surface we would need to use:

$$\phi = 2\pi r_1 \tag{6}$$

$$\cos\theta = 2\left(r_2 - \frac{1}{2}\right) \tag{7}$$

where r_1 and r_2 are two random numbers uniformly distributed in the interval $(0, 1)$. It is only by choosing $\cos\theta$ (not θ) from a uniform distribution that random points will be uniformly distributed over the surface of the sphere. To carry out the Monte Carlo integration, indicated in Eq. (5), we would need to generate N random points and evaluate $f(x, y)$ at each of these points.

1e. Precision Of The Monte Carlo Method. Like any method of numerical integration, the Monte Carlo method only can yield an approximate result. In other numerical integration methods, such as Simpson's rule,[1] the error typically decreases as the number of integration intervals, N, increases according to N^{-1}. The Monte Carlo method, converges on the exact result as the number of random points, N, increases. The convergence in the case of the Monte Carlo method is proportional to $N^{-1/2}$. This slower rate of convergence is a significant drawback to the Monte

---

[1] See "Numerical Integration" (MISN-0-349).

Carlo method, since it means that higher N values (more computer time) is needed to achieve the same precision. There are, nevertheless, several advantages to the Monte Carlo method over more conventional numerical integration methods that will become clearer in discussing integrals in more than two dimensions.

2. More Than Two Dimensions

2a.   Introduction. Suppose we wish to integrate some function $f(x, y, z)$ over the interior volume V of a closed surface of some specific shape:

$$M = \int_V f(x, y, z)\, dV \,. \tag{8}$$

This volume integral actually stands for the triple integral:

$$M = \int \int \int f(x, y, z)\, dx\, dy\, dz \,. \tag{9}$$

The limits on each successive integral are determined by the specific shape of the region V. This may result in difficult functions being introduced into the next integrand as successive integrations are carried out analytically. This source of difficulty is completely bypassed when an integral such as that in Eq. (8) is carried out in the Monte Carlo method. Equation (8) may be approximated by the finite sum:

$$M = \sum_{j=1}^{N} f_j(x, y, z) \Delta V \tag{10}$$

The surface enclosing the volume V which is the region of integration is itself encased in a rectangular box of volume V_0 (see Fig. 3). In order to evaluate Eq. (10) we need to choose N randomly located points inside the box, and calculate $f_j(x, y, z)$ at each of the points $j = 1, 2, ..., N$. The only contributions to the sum are from those points that lie inside the volume V. The size of the volume element ΔV in Eq. (10) is given by $\Delta V = V_0/N$.

2b. Moments Of A Probability Distribution.   One particular type of function $f(x, y, z)$ that often appears in a multiple integral is the probability density function $\rho(x, y, z)$. This is the probability that x, y, z take on values corresponding to the location of the volume element ΔV. The function $\rho(x, y, z)$ can also stand for the mass density or charge density
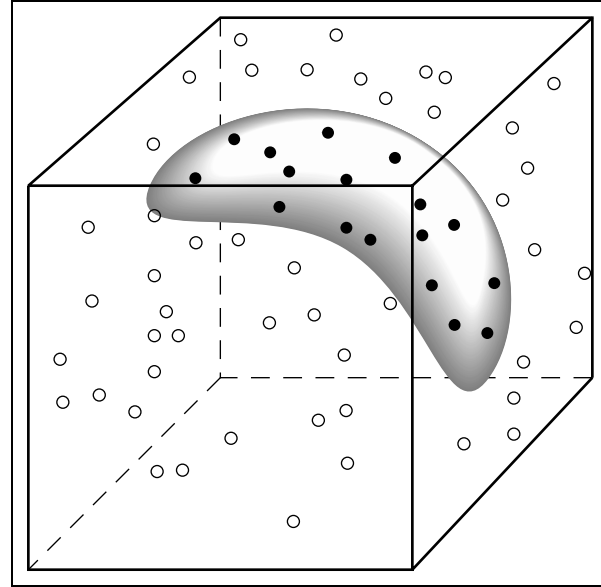
Figure 3. Monte Carlo method for doing a volume integral: N random points are chosen inside a box. Only those points which also lie inside the volume of integration (the shaded closed surface) contribute in Eq. (10).

as a function of position. We may define the nth moment of a three-dimensional probability distribution according to:

$$M_n = \int x^n \rho(x, y, z)\, dV \,. \tag{11}$$

This integral can be approximated by the finite sum:

$$M_n = \sum_{j=1}^{N} x_j^n \rho_j(x, y, z)\, \Delta V \,. \tag{12}$$

2c. Moments Of A Mass Distribution.   In the event that ρ represents a mass density rather than a probability density, then the product $\rho_0 \Delta V$ represents the mass Δm_j of one of N randomly located point masses. (A similar statement would hold for point charges with $\Delta q = \rho \Delta V$). Thus, Eq. (12) may be written:

$$M_n = \sum_{j=1}^{N} x_j^n \Delta m_j \tag{13}$$

The first three moments ($n = 0, 1, 2$) are of particular interest:

$$M_0 = \sum_{j=1}^{N} \Delta m_j \tag{14}$$

$$M_1 = \sum_{j=1}^{N} x_j \Delta m_j \tag{15}$$

$$M_2 = \sum_{j=1}^{N} x_j^2 \Delta m_j \tag{16}$$

From the Monte Carlo evaluation of Eq. (14) we can obtain the mass of a three-dimensional body of specified shape. Equation (15) yields the x-coordinate of the center of mass, and Eq. (16) can be used to find the moment of inertia. There are actually three moments of inertia (about each of the three axes). The three center of mass coordinates and moments of inertia are given by:

Center of Mass Coordinates:

$$x_{cm} = \frac{1}{M_0} \sum_j x_j \Delta m_j$$

$$y_{cm} = \frac{1}{M_0} \sum_j y_j \Delta m_j$$

$$z_{cm} = \frac{1}{M_0} \sum_j z_j \Delta m_j \tag{17}$$

Moments of Inertia:

$$I_{xx} = \sum_j (y_j^2 + z_j^2) \, \Delta m_j$$

$$I_{yy} = \sum_j (x_j^2 + z_j^2) \, \Delta m_j$$

$$I_{zz} = \sum_j (x_j^2 + y_j^2) \, \Delta m_j \tag{18}$$

2d. Illustration Of The Monte Carlo Method. A specific illustration may help explain the use of these equations in the Monte Carlo method. Assume that we wish to find the coordinate of the center of mass using Eq. (17). This would necessitate generating N randomly located points inside the rectangular box that encloses the object of interest (see Fig. 3). For each point a value of its mass Δm_j is calculated from $\Delta m_j = \rho_j(x, y, z) \, \Delta V$ where $\rho(x, y, z)$ is some specified mass density function (that might be a constant), and $\Delta V = V_0/N$ where V_0 is the volume of the rectangular box. Random points that fall outside the boundaries of the object are taken to have $\Delta m_j = 0$, when calculating the sum $\sum y_j \Delta m_j$. In the case of an object of constant density we may use:

$$\Delta m_j = \frac{M_0}{n}, \tag{19}$$

where M_0 is the mass of the object and n is the number of random points that fall inside its boundaries. It is interesting to recall that the representation of an object by a random collection of mass points is actually a closer approximation to the atomic nature of matter than the usual mathematical view of a continuous solid. The number of random points used in Monte Carlo calculations, however, never even comes close to the number of discrete atoms in a normal piece of matter, due to limitations of computer time.

2e. Integrals In More Than Three Dimensions. Although physical space only has three dimensions, mathematically we are sometimes faced with more than three variables. It is easy to do higher-dimensional integrals using the Monte Carlo method. For example, let us consider the case of a ten-dimensional sphere called a "hypersphere" of constant density. In three dimensions the interior of a sphere is defined by:

$$x^2 + y^2 + z^2 \leq R^2_0, \tag{20}$$

where R_0 is the radius. Likewise we may define the interior of a ten-dimensional hypersphere of radius R_0 from

$$x^2_1 + x^2_2 + ... + x^2_{10} \leq R^2_0. \tag{21}$$

Let us imagine the ten-dimensional hypersphere is encased in a unit side ten-dimensional "hypercube" having unit volume and density. It is quite easy to find the mass or volume of the ten-dimensional hypersphere by generating random points (of ten coordinates each) inside the hypercube. Each random point is then tested to see if it is inside the hypersphere
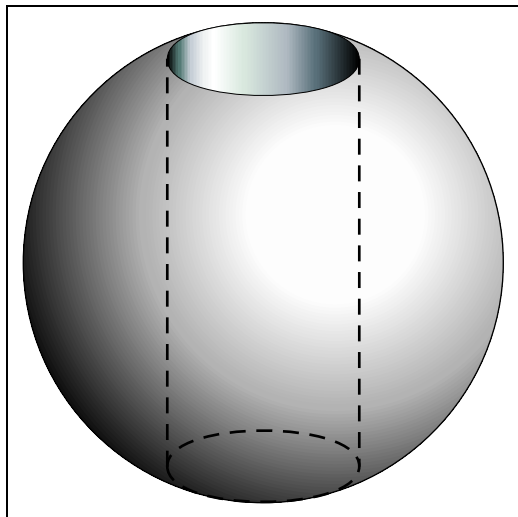
Figure 4. The object used in the present version of the program.

according to Eq. (21). Those points which are inside, contribute a mass $\Delta m = \rho \Delta V = M/N$ to the total, where M is the mass of the unit density hypercube, and N is the total number of random points. Outside random points contribute nothing. The total mass of the hypersphere is found from the sum over all point masses. The result is that the ratio of the volume of a hypersphere to that of the hypercube that just encloses it equals the fraction of the random points which lie inside the hypersphere, $f = n/N$. Doing a multiple integral by conventional numerical integration methods generally involves allowing each variable to take on a range of values. For a ten-dimensional integral if each variable takes on only 2 values we would need to calculate the integrand at 2^{10} terms. This represents one major advantage of the Monte Carlo method over conventional integration methods for multiple integrals in many dimensions.

3. Program to Calculate Integrals

3a. Introduction. A program has been written to calculate the mass and center of mass of an object, as described in Section 2. For the sake of definiteness, the object is chosen to have a specific shape: a sphere of radius $R = 0.5$ with a cylindrical hole of radius $r = 0.3$ through the center of the sphere (see Fig. 4). We assume that the sphere is enclosed by a cube one unit on a side centered at the origin. In that case, we may use triplets of pseudorandom numbers (r_1, r_2, r_3) to calculate the

coordinates (x, y, z) of random points according to:

$$x = r_1 - \frac{1}{2}$$

$$y = r_2 - \frac{1}{2}$$

$$z = r_3 - \frac{1}{2}$$

A random point would be considered to be inside the object provided that:

$$x^2 + y^2 + z^2 < 0.5^2 \qquad (22)$$

and

$$x^2 + y^2 > 0.3^2. \qquad (23)$$

Equation (22) checks that the point is inside the sphere, and Eq. (13) that it is not inside the hole. In the present version of the program the density of the object is considered to be constant. Random points inside are taken to have $\rho = 1.0$ and those outside $\rho = 0$. Since the volume of the encasing cube is also assumed to be unity, this means that the calculated mass of the object will be equal to f, the fraction of random points that lie inside. The program also calculates the x, y, and z coordinates of the center of mass as described in Paragraph 2c.

3b. Input. The program begins by reading numerical values for N and NRUNS, where

> N = the number of randomly located points to use in calculating the mass and center of mass coordinates, and
>
> NRUNS = the number of times to repeat the whole calculation using the same value of N.

The reason that the calculation is repeated some number of times (NRUNS) is that when random numbers are used the results depend to some extent on the particular random numbers chosen. It is, therefore, of interest to see how much variation occurs from run to run as a direct measure of the precision of the method. The output listed was obtained using the following input parameters:

> 1st case: N = 250 ,   NRUNS = 5
> 2nd case: N = 1000 ,   NRUNS = 5

3c. Sample Output.

```
N=250.00         NRUNS=5.00
RUN   1   MASS   =   .252
RUN   2   MASS   =   .264
RUN   3   MASS   =   .292
RUN   4   MASS   =   .248
RUN   5   MASS   =   .236
AVERAGE MASS .258
DEVIATION .019


C.M. COORD'S   X= .071   Y= .069   Z= .006
C.M. COORD'S   X=-.014   Y= .030   Z= .002
C.M. COORD'S   X=-.052   Y= .024   Z=-.008
C.M. COORD'S   X=-.007   Y= .006   Z= .002
C.M. COORD'S   X= .095   Y=-.019   Z=-.014
AVERAGE COORDINATES .019 .022 -0.002
DEVIATIONS .055 .029 .008
```

```
N=1000.00 NRUNS=5.00
RUN 1 MASS   =   .264
RUN 2 MASS   =   .287
RUN 3 MASS   =   .270
RUN 4 MASS   =   .281
RUN 5 MASS   =   .275
AVERAGE .275
DEVIATION .008

C.M. COORD'S   X=-.002   Y= 0.032   Z= .000
C.M. COORD'S   X= .007   Y=-0.010   Z=-.008
C.M. COORD'S   X= .035   Y=-0.003   Z= .010
C.M. COORD'S   X= .002   Y=-0.027   Z=-.012
C.M. COORD'S   X=-.007   Y=-0.010   Z= .004
AVERAGES .007 -0.004 -0.001
DEVIATIONS .015 .020 .008
```

We note that the x, y, z-coordinates of the center of mass are, as expected close to zero, due to the symmetrical shape of the object. The masses for each run are all reasonably close to the expected value 0.268. (The mass of a sphere of radius a, which has a cylindrical hole of radius c, is given

by

$$M = \frac{4}{3}\pi\,(a^2 - c^2)^{3/2}\rho\,;$$

$$a = 0.5, c = 0.3, \rho = 1.0.$$

The computed rms deviations for each series of runs is a measure of the errors in each quantity. The differences between the computed values and the expected values of the mass and the center of mass coordinates are, in most cases, less than the rms deviations. (The laws of statistics predict this should be the case about 68% of the time). The laws of statistics also predict that the rms deviations should decrease as the number of random points N increases, according to the formula $N^{-1/2}$. Thus, we should find the standard deviations for $N = 1000$ are about half those for $N = 250$. This is roughly borne out by the results. The agreement would generally be better if a larger number of runs was used in each case.

4. Procedures

4a. Calculated Mass And Center of Mass Coordinates. Run the program using $N = 100$ and NRUNS $= 50$. Make a histogram showing the number of runs in which the calculated mass lies in each of a number of equal-size intervals. Do the same for the calculated x and y center of mass coordinates, combining the values for both coordinates in one plot, instead of making two separate plots. Both histograms should have a shape which resembles a Gaussian distribution, and the latter histogram should be centered near zero, if the random number generator is working properly. The mass distribution should be centered at the expected value 0.2681.[2] For both distributions find the width which in each case includes roughly 68 percent of the cases on both sides of the expected value. The standard deviation σ is roughly half of that full width.

4b. Variation Of Results With N. Repeat the procedure suggested in the preceding problem using other values of N, say $N = 25$ and $N = 400$. Compare the distributions you obtain in these two cases with the original ones. In particular, see if the predicted $N^{-1/2}$ dependence of the error in each quantity (determined from the width of the distribution) appears to be valid.

4c. Objects Of Other Shapes. Modify the program so that it computes the mass and center of mass coordinates for objects of other shapes. For example, you can try the shapes shown in Fig. 5. Be sure to select

---

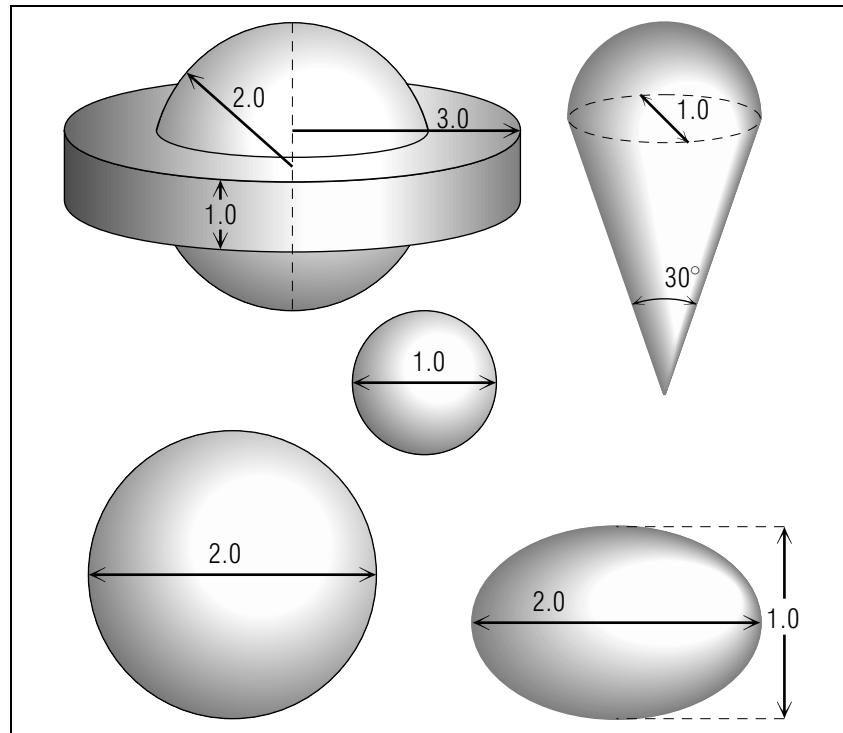[2]Volume $= (4/3)\pi(R^2 - r^2)^{3/2}$ where $R =$ sphere radius, $r =$ cylinder radius.

Figure 5. Sample objects for which to calculate the mass and center of mass.

random numbers for each coordinate between limits which define a rectangular block that just encloses the object. Use a large enough `N` to get a reasonably accurate result, and get an estimate of the error by using `NRUNS > 1`.

4d. Surface Integrals. Modify the program to evaluate an integral on the surface of a sphere, as described in Paragraph 1d. To do this you will need to generate N random points having angular coordinates θ and ϕ defined by Eqs. (6) and (7). Try integrating the function $f(\theta, \phi) = 1 + \cos^2 \theta$ over the surface of a sphere of unit radius using Eq. (5). The surface element ΔS in Eq. (5) is given by $\Delta S = S/N$ where S is the total surface area of the sphere and N is the number of random points. How close does your result come to the expected value of 1.5 using different values of N?

B. Fortran, Basic, C++ Programs

All programs are at

http://www.physnet.org/home/modules/support_programs

which can be navigated to from the home page at

http://www.physnet.org

by following the links: → `modules` → `support programs`, where the programs are:

m355p1f.for, Fortran;
m355p1b.bas, Basic;
m355p1c.cpp, C++;
lib351.h, needed Library for C++ program;

MODEL EXAM

1. See Output Skills K1-K2.

Examinee:

On your computer output sheet(s):

(i) Mark page numbers in the upper right corners of all sheets.

(ii) Label all output, including all axes on all graphs.

On your Exam Answer Sheet(s), for each of the following parts of items (below this box), show:

(i) a reference to your annotated output; and

(ii) a blank area for grader comments.

When finished, staple together your sheets as usual, but include the original of your annotated output sheets just behind the Exam Answer Sheet.

---

3. Submit your hand-annotated output for a Monte Carlo calculation of the mass and CM coordinates for a sphere with a hole through its center. Be sure it shows:

   a. a histogram showing the number of mass values that lie in each of a number of equal-sized intervals, with 50 values and 100 Monte Carlo points per value.

   b. a single histogram showing both the number of the x and the y CM coordinates that lie in a number of equal-sized intervals, with 50 values and 100 Monte Carlo points per value.

   c. for a) and b), a discussion of what distributions are expected about what points and a demonstration of how close your Monte Carlo values came to them.

   d. a comparison of the results from using three numbers of Monte Carlo points, N, everything else kept fixed, and the extent to which the error values have the predicted dependence on N.

4. Submit your hand-annotated output for the mass and CM coordinates for an object from Fig. 5 (or equivalent in difficulty). Be sure it shows a deduced estimate of error.

5. Submit your hand-annotated output showing Monte Carlo integration of a function over the surface of a unit sphere. Be sure it shows how close your result came to the exact value.

INSTRUCTIONS TO GRADER

If the student has submitted copies rather than originals of the computer output, state that on the exam answer sheet and **immediately stop grading the exam and give it a grade of zero.**