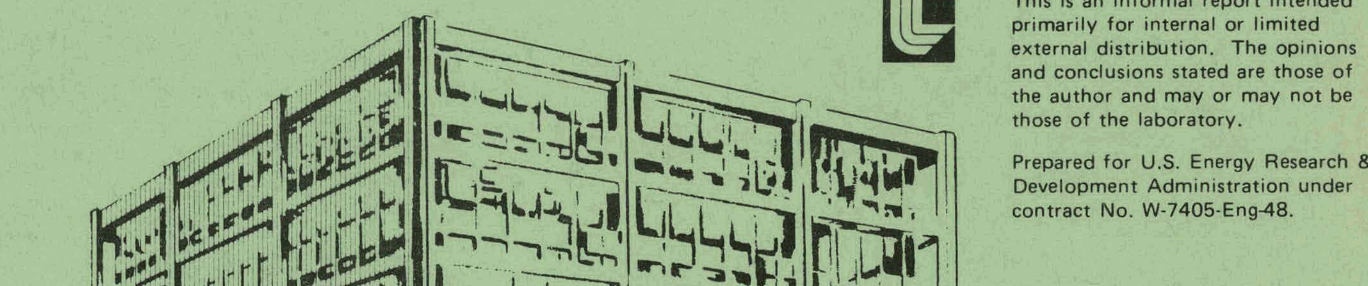
UCID- 17522

Lawrence Livermore Laboratory

VOLUME CALCULATION AND GEOMETRY CHECKING IN A MONTE CARLO TRANSPORT CODE

Paul F. Dubois

July 1977



Prepared for U.S. Energy Research & Development Administration under contract No. W-7405-Eng-48.

DISCLAIMER

VOLUME CALCULATION AND GEOMETRY CHECKING IN A MONTE CARLO TRANSPORT CODE

Paul F. Dubois

Numerical Mathematics Group
Computation Department
Lawrence Livermore Laboratory, Livermore, CA 94550

ABSTRACT

An algorithm for finding cross-sectional areas of zones in a Monte Carlo transport code is given. The method is well suited to problems involving thin shells, and includes the capability of checking for gaps and overlaps.

1. Introduction

Monte Carlo codes require a physicist to divide the space of a problem into "zones", in which different physical conditions hold. A water pipe being bombarded by particles might be described as three zones, consisting of the outer air, the pipe itself, and the space inside the pipe. These zones could be delineated by giving equations of the inner and outer surfaces of the pipe. In a real problem, the zones are much more complicated. There may be up to 200 zones delineated by up to 100 planes and 100 quadratic or other type surfaces. Various codes have different limits on the number and type of allowable surfaces.

TARTNP (4) is such a code. It is in wide use at Lawrence Livermore Laboratory, being about the fifth largest time consumer among Laboratory codes. We shall confine ourselves to the surfaces allowed in TARTNP, hoping that others may find the methods described a useful aid in attacking similar problems.

Describing the zones is a tremendous task. In TARTNP, the user may describe up to 100 planes $ax + by + cz = d$, and up to 150 conic surfaces of the form

$$a(x-x_0)^2 + b(y-y_0)^2 + c(z-z_0)^2 = k^2$$

Then, for each zone, of which there may be up to 200, a description is given in the form of "JPB's" which are pairs of signed integers. The second integer tells to which surface is referred (e.g. +6 = conic#6, -3 = plane #3), while the sign of the first integer gives the side of that surface on which the zone lies. The sign is denoted by J, and J is positive if, substituting the values (x,y,z) of a point in the zone into the relation

$$ax + by + cz \leq d \quad \text{(if the boundary is a plane)}$$

or $$a(x-x_0)^2 + b(y-y_0)^2 + c(z-z_0)^2 \leq k^2 \quad \text{(if the boundary is a conic)}$$

yields a true statement. Otherwise, J is negative.

The problem is to find the volume of each zone to within some tolerance, say 1%, and to detect whether any zones overlap or have gaps between them. Both of these tasks are directed toward helping the physicist be sure the zone descriptions are correct before spending hours of computer time on the run itself. The method must be of reasonable cost.

The method used in TARTNP has been a hit-or-miss Monte Carlo method for the general problem, or a direct analytic calculation if the conic surfaces have a common axis of revolution and the planes are perpendicular to this axis.

The analytic method is satisfactory, of course, but the Monte Carlo method is not. It has trouble with zones, such as a pipe, with small volume but large surface area.

The method tries to check for gaps and overlaps, which is important. As each point is chosen, the zone descriptions are checked. If the point satisfies none of these or more than one, an error message can be issued. Any competing method must have the ability to detect and analyse such problems.

An algorithm for calculating these volumes and detecting gaps and overlaps is described below. An experimental preprocessor and graphics package based on this idea has been developed and has proven satisfactory on some real test problems, and will now undergo further testing and refinement ([1]). Short of analytic solution, which we believe to be impractical, no method can solve all problems in a reasonable time. Therefore, our method requires that it be used by the problem designer in an intelligent way, with a knowledge of its methods and limitations. In this context, it seems to be a practical solution for most real problems.

2. Overview

Before getting into the details of the calculation, we describe the overall process. A box, $XL \leq x \leq XU$, $YL \leq y \leq YU$, $ZL \leq z \leq ZU$, is given, and it is desired to find the volumes of the zones described by "JPB"s within the box. Some given number (ZNUM) of z-values are chosen, either at random from (ZL,ZU) or by a stepping procedure. For each value $Z=Z_1$, the algorithm finds the intersection of each given surface with the plane area $Z = Z_1$, $XL \leq x \leq XU$, $YL \leq y \leq YU$. A value XNUM is given, and the segment (XL,XU) is divided by the points

$x_{k+1} = XL + (2k+1)\Delta X/2$, $k = 0,1,\ldots XNUM-1$, where $\Delta X=(XU-XL)/XNUM$.

For each x_k, the intersection of the volume "below" each surface (i.e. the J = positive region) with the line $Z = Z_1$, $X = x_k$, is found and described as a triple $(\pm 1, y_1, y_2$. Here we define for $-\infty \leq y_1 \leq y_2 \leq \infty$,

$$(1,y_1, y_2) \equiv (y_1,y_2)$$

$$(-1,y_1,y_2) \equiv (y_1,y_2) = (-\infty, y_1) \cup (y_2,\infty)$$

In computation, we use suitable machine numbers for $\pm\infty$. Whether one considers the intervals open or closed is irrelevant.

We start by assuming a zone occupies the entire segment (YL,YU) of the line $Z = Z_1$, $X = X_k$. Using the "JPB"s for the zone, we intersect the set described by a triple (s,y_1, y_2) with the current description of the zone until the "JPB"s are exhausted. The triple used is that (s',y_1,y_2) associated with the boundary in a JPB, with $s = s'*J$, where J is the sign (± 1) from the JPB. (This adjusts the triple for whether we want the "inside" or the "outside" of that surface.)

We are left with a description of the zone as a disjoint union $(y_1,y_2) \cup (y_3, y_4) \cup \ldots \cup (y_m,y_{m+1})$. The total length $\ell_I(k)$ of these intervals can then be used to estimate the cross-sectional area of the zone I with the plane $Z = Z_1$ by $C_I(Z_1) = \sum_k \ell_I(k)$.

Also, the descriptions of all the zones taken together allow us to check whether or not there are any gaps or overlaps in the line $X = x_k$, $Z = Z_1$, $YL \leq y \leq YU$. A simple test, which works for most errors, is to test whether

$$\left| \sum_I \ell_{I(k)} - (YU-YL) \right| \leq (YU-YL) * TOL$$

for some tolerance TOL. If not, a further investigation can detail the difficulty.

We present the algorithms below as if there was only one zone; in reality, most quantities are arrays indexed by zone number.

First, we give the algorithm for finding the triples associated with each surface and for updating these cheaply from x_k to x_{k+1}. Then we give the algorithm for intersecting these triples to find the zone description. Finally, we discuss accuracy and cost compared to the Monte Carlo method.

3. Finding and updating boundary triples

Let Z_1, XL,XU,YL,YU, and XNUM be given, and let Δx = (XU-XL)/XNUM.

Define $x_{k+1} = XL + (2k+1)\Delta x/2$, k = 0,1,...XNUM-1.

For a surface given by the equation

$$a(x-x_0)^2 + b(y-y_0)^2 + c(z-z_0)^2 = r^2$$

we want to associate a triple $(\pm 1, y_1, y_2)$ so that it describes the set

$$(\pm 1, y_1, y_2) = \{y \,|\, a(x_j-x_0)^2 + b(y-y_0)^2 + c(z_1-z_0)^2 < r^2\}$$

for a given x_j. First we assume j= 1. There are two cases, b=0 and b≠0.

First, if b=0, let $S_1 = r^2 - a(x_1-x_0)^2 - c(Z_1-Z_0)^2$.

The desired triple is $(-\text{sgn } S_1, y_0, y_0)$.

If b≠0, let $S_1 = b^{-1}(r^2 - a(x_1-x_0)^2 - c(Z_1-Z_0)^2)$.

The desired triple is $(\text{sgn } b, y_0-\Gamma, y_0+\Gamma)$, where $\Gamma = (\max\{0., S_1\})^{1/2}$.

Define $S_k = (r^2 - a(x_k-x_0)^2 - c(Z_1-Z_0)^2)/b^*$, where $b^* = b$ if b≠0 or $b^* = 1$ for b=0. Then

$$\begin{aligned} b^*(S_k - S_{k-1}) &= -a(\,(x_k-x_0)^2 - (x_{k-1}-x_0)^2) \\ &= -a(\,x_k^2 - x_{k-1}^2 - 2x_0(x_k - x_{k-1})\,) \\ &= -a(\,\Delta x(x_k + x_{k-1}) - 2x_0\,\Delta x) \\ &= -a(2(x_1-x_0)\Delta x + (2k-3)(\Delta x)^2) \end{aligned}$$

using the substitution $x_j = x_1 + (j-1)\,\Delta x$.

As we compute S_1, then, we compute and store the quantities

$$\Delta S_1 = -2a(x_1 - x_o)\ \Delta x/b^*$$

$$\Delta S_2 = -a(\Delta x)^2/b^* \quad .$$

Then we can update $S = S_j$ to $S = S_{j+1}$ by setting $S = S_1$, $\beta = -1$ initially and then performing

$$\beta: = \beta + 2$$

$$S: = S + \Delta S_1 + \beta \Delta S_2 \quad .$$

To update the triple for x_k to the one for x_{k+1} we need only check whether or not $b = 0$. The quantities a, c, x_o, y_o, z_o, r^2, and z_1 need not be re-fetched.

For plane surfaces, the process is similar. Let $\pm\infty$ be suitable machine numbers. Suppose the plane is given by

$$ax + by + cz = d \quad .$$

We seek $(s,\ y_1, y_2)$ so that

$$(s,\ y_1,\ y_2) = \{y \mid ax_k + by + cz_1 < d\} \quad .$$

Let b^* be as above and $T_1 = (d - ax_1 - cz_1)/b^*$. For $b \neq 0$, the desired triple is $(\text{sgn } b,\ -\infty,\ T_1)$. If $b = 0$, the triple we seek is $(\text{sgn } T_1,\ -\infty,\ \infty)$. For updating, let $\Delta T = -a\ \Delta x/b^*$. Then letting $T = T_1$ originally, we update T by

$$T: = T + \Delta T \quad .$$

4. The Intersection Algorithm

Suppose a region in the line segment (YL, YU) is described by a dijoint union $(y_1,y_2) \cup (y_3,y_4) \cup \ldots \cup (y_{2m-1}, y_{2m})$ of m intervals. We wish to find the intersection of this union with the set described by the triple (s, x, y). In the implementation of the algorithm, initially m = 1, y_1 = YL, y_2 = YU. As new JPB triples are intersected with this description, the number of pieces m may grow, leading to the problem just described. We suppose the values y_1, y_2, ... y_{2m} to be stored in an array with m stored separately. If this array is Y, we will use y_i for Y(i).

Case I: s = +1

For i = 1 to m do:

(1) $w_1 := \max(y_{2i-1},x)$

(2) $w_2 := \min(y_{2i},y)$

(3) $y_{2i-1} := w_1$

(4) $y_{2i} := \max(w_1,w_2)$.

It would be useful to test whether x = y before beginning this procedure. If so, set m = 0 and we are done — the region is empty. Other JPB's can be ignored for this zone.

Case II: s = -1

Let NEW: = 0

For i = 1 to m do:

(1) $w_1 := \max(y_{2i-1},y)$

(2) $w_2 := \min(y_{2i},x)$

(3) If $y_{2i-1} \geq w_2$, then $y_{2i-1}: = \min(w_1, y_{2i})$ and proceed to the next i.

(4) If $y_{2i} \leq w_1$, then $y_{2i}: = \max(w_2, y_{2i-1})$ and proceed to the next i.

(5) If neither (3) nor (4) pertains:

(a) If x = y, exit routine

(b) NEW: = NEW + 1

(c) $y_{2(m+NEW)-1}: = w_1$

(d) $y_{2(m+NEW)}: = y_{2i}$

(e) $y_{2i} = w_2$.

After completing this loop, m: = m + NEW. (a) is done to prevent wasted time and storage since $(-1, x, x) \equiv (-\infty, \infty)$.

When the process described in this section is completed for all JPBS, for a particular x_k, then we have $\ell_I(k) = \sum_{i=1}^{m} (y_{2i} - y_{2i-1})$.

Thus far, we have found m does not exceed 5 in practice.

5. Comparison with Monte Carlo

Let B be a zone contained in the box $XL \le x \le XU$, $YL \le y \le YU$, $ZL \le z \le ZU$. Let $XU - XL = a$, $YU - YL = b$, $ZU - XL = c$. Let Volume (B) $= p \cdot V$, $0 < p \le 1$, where $V = abc$.

Using the hit or miss Monte Carlo method, we choose N points in the box in a uniform random manner. We then estimate Volume (B) by

$$V_e = V \cdot N^*/N$$

where N^* is the number of points found to lie in B. V_e is a random variable with expected value $\langle V_e \rangle = V \langle N^*/N \rangle = Vp =$ Volume (B), and variance

(1) $$\text{var } V_e = V^2 p(1-p)/N \quad .$$

Let us examine an alternate procedure. Suppose we sample M points z_i from the line segment (ZL,ZU). For each z_i so chosen, suppose we can find the cross-sectional area $C(z_i)$ of $B \cap \{(x,y,z) | z = z_i\}$. Let $A(z) = C(z)/ab$, the proportion of $C(z)$ to the rectangular area ab. Define

$$V_c = (V/M) \sum_{i=1}^{M} A(z_i) \quad .$$

Then

$$\langle V_c \rangle = (V/M) \sum_{i=1}^{M} \langle A(z) \rangle = (V/M) \cdot M \cdot \int_{ZL}^{ZU} A(\xi)\ (1/c) d\xi$$

$$= V \cdot \int_{ZL}^{ZU} (C(\xi)/abc) d\xi = \int_{ZL}^{ZU} C(\xi)\ d\xi = \text{Volume (B)}.$$

We are assuming that B has been produced by the kinds of intersections described in this paper so that C(z) is at least piece-wise continuous.

We find the variance of V_c to be

$$(2) \qquad \operatorname{var} V_c = \operatorname{var}\left(V\cdot\left(\sum_{i=1}^{M} A(z_i)/M\right)\right)$$

$$= V^2 \operatorname{var}\left(\sum_{i=1}^{M} A(z_i)/M\right) = V^2 \operatorname{var} A(z)/M \quad .$$

Let us assume that the labor required per point for the two methods are L_1 and L_2 respectively. We would expect $L_1 \leq L_2$ since to find $A(z_i)$ by whatever method is presumably more work than discovering whether or not a point is in B.

Hammersley ([3]) defines the relative efficiencies of two estimators requiring labor TL_1 and TL_2 and producing variances in the estimator σ_1^2 and σ_2^2 to be $\rho = (TL_1)\sigma_1^2/(TL_2)\sigma_2^2$. Thus, for V_e and V_c,

$$\rho = \frac{(ML_2)\ V^2 \operatorname{var} A(z)/M}{(NL_1)\ V^2\ p(1-p)/N} = (L_2/L_1)(\operatorname{var} A(z)/p(1-p)) \quad .$$

<u>Theorem</u>. $\operatorname{var} A(z) \,/\, p(1-p) \leq 1$.

<u>Proof</u>. Note that

$$\langle A(z)\rangle = \int_{ZL}^{ZU} A(z)\,\frac{1}{c}\,dz = \frac{\text{Volume (B)}}{V} = p \quad .$$

$$\text{Thus, } \operatorname{var} A(z) - p(1-p) = \left[E((A(z))^2) - p^2\right] - (p-p^2)$$

$$= \int_{ZL}^{ZU} (A(z))^2\,\frac{1}{c}\,dz - p = \frac{1}{c}\int_{ZL}^{ZU} (A(z))^2 - A(z)dz$$

$$= \frac{1}{c}\int_{ZL}^{ZU} -A(z)(1 - A(z))dz \leq 0 \;, \quad \text{since } 0 \leq A(z) \leq 1 \;.$$

The result follows.

The table below shows the values of the ratio var A(z) / p(1-p) for various B contained in the box $-1 \leq x,y,z \leq 1$. The definitions of these volumes are:

B_1: A slab, $-p \leq z \leq p$, $0 < p \leq 1$

B_2: A sphere, $x^2 + y^2 + z^2 \leq 1$

B_3: A cylinder, $x^2 + y^2 \leq 1$, $-1 \leq z \leq 1$

$B_4(\delta)$: A shell of width δ, $(1-\delta)^2 \leq x^2 + y^2 + z^2 \leq 1$.

TABLE 1

Zone	p	var A(z)	var A(z) / p(1-p)
B_1	p	p(1-p)	1.00
B_2	.52	5.4×10^{-2}	.22
B_3	.79	0	0.00
$B_4(.5)$	.46	3.1×10^{-2}	.12
$B_4(.1)$	.14	6.7×10^{-4}	5.49×10^{-3}
$B_4(.01)$	1.55×10^{-2}	8.1×10^{-7}	5.26×10^{-5}

A value for L_2/L_1 is problem dependent and dependent on the method used to find A(z). For our algorithm, we are estimating A(z) so the above analysis is not exact. We estimate that for a typical TARTNP problem (L_2/L_1) is on the order of XNUM, the number of x slices. Note that if we are doing all the zones at once we are obliged to choose N and M large enough to get the most difficult zone's volume accurately. One thin shell is all it takes to make the second method a big winner.

Zone B_1 illustrates the virtues of making the sample from (ZL,ZU) a stratified sample. We divide the segment (ZL,ZU) into n pieces, and choose n_i points in the i^{th} piece, where $\sum_{i=1}^{n} n_i = M$. The estimates V_c are found in each piece separately and then summed to get a total volume. If the variance of A(z) within each piece is less than the variance of A(z) between the pieces, substantial increases in accuracy can be obtained. See [3; p.55]. Thus, if there are natural planes $z = z_i$ in the problem, we should break the problem into pieces at these boundaries.

For related reading, we refer the reader to [2].

REFERENCES

1. Dubois, Paul F., PRETART/TARTV User's Manual, UCID-17518 Lawrence Livermore Laboratory, Livermore, CA (1977).

2. Eastman, C.M. and C. Yessios, "An Efficient Algorithm for Finding the Union, Intersection, and Differences of Spatial Domains," available as Report AD-755958, NTIS, U.S. Dept. of Commerce, Springfield, VA (1972).

3. Hammersley, J.M. and D. Handscomb, Monte Carlo Methods, London: Methuen & Co., Ltd., 1964.

4. Plechaty, Ernest F. and John R. Kimlinger, "TARTNP: A Coupled Neutron-Photon Monte Carlo Transport Code," UCRL-50400 Vol. 14, Lawrence Livermore Laboratory, Livermore, CA (1976).

Printed in the United States of America
Available from
National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Road
Springfield, VA 22161
Price: Printed Copy $; Microfiche $3.00

Page Range	Domestic Price	Page Range	Domestic Price
001–025	$ 3.50	326–350	10.00
026–050	4.00	351–375	10.50
051 075	4.50	376–400	10.75
076–100	5.00	401–425	11.00
101–125	5.50	426–450	11.75
126–150	6.00	451–475	12.00
151–175	6.75	476–500	12.50
176–200	7.50	501–525	12.75
201–225	7.75	526–550	13.00
226–250	8.00	551–575	13.50
251–275	9.00	576–600	13.75
276–300	9.25	601–up	*
301–325	9.75		

*Add $2.50 for each additional 100 page increment from 601 to 1,000 pages; add $4.50 for each additional 100 page increment over 1,000 pages.

Technical Information Department
LAWRENCE LIVERMORE LABORATORY
University of California | Livermore, California | 94550