AD-A034 513 MATHEMATICAL APPLICATIONS GROUP INC ELMSFORD N Y F/G 12/1
SOLUTION OF TIME-DEPENDENT HEAT CONDUCTION EQUATION IN COMPLEX --ETC(U)
DEC 76 E S TROUBETZKOY, M H KALOS DAAD05-73-C-0501

UNCLASSIFIED BRL-CR-325 NL

| OF |
AD
A034513

END
DATE
FILMED
2-77

BRL CR 325

BRL

12 b.s.

AD

CONTRACT REPORT NO. 325

SOLUTION OF TIME-DEPENDENT HEAT CONDUCTION EQUATION IN COMPLEX GEOMETRY BY THE MONTE CARLO METHOD

Prepared by

Mathematical Applications Group, Inc.
3 Westchester Plaza
Elmsford, NY 10523

December 1976

USA BALLISTIC RESEARCH LABORATORIES
ABERDEEN PROVING GROUND, MARYLAND

Destroy this report when it is no longer needed.
Do not return it to the originator.

Secondary distribution of this report by originating or sponsoring activity is prohibited.

Additional copies of this report may be obtained from the National Technical Information Service, U.S. Department of Commerce, Springfield, Virginia 22151.

The findings in this report are not to be construed as an official Department of the Army position, unless so designated by other authorized documents.

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

REPORT DOCUMENTATION PAGE 19		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER CONTRACT REPORT NO. 325 ✓ 18	2. GOVT ACCESSION NO. BRL-CR-325	3. RECIPIENT'S CATALOG NUMBER
4. TITLE *(and Subtitle)* 6 Solution of Time-Dependent Heat Conduction Equation in Complex Geometry by the Monte Carlo Method.		5. TYPE OF REPORT & PERIOD COVERED 9 Final rept. 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) 10 Eugene S. /Troubetzkoy, Malvin H. /Kalos, Herbert A. /Steinberg, Norman E. /Banks, George J. /Klem		8. CONTRACT OR GRANT NUMBER(s) 15 DAAD05-73-C-0501 New
9. PERFORMING ORGANIZATION NAME AND ADDRESS Mathematical Applications Group, Inc. 3 Westchester Plaza Elmsford, N~~ew~~ Y~~ork~~ 10523 16		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 1W161102AH43
11. CONTROLLING OFFICE NAME AND ADDRESS USA Ballistics Research Laboratories Aberdeen Proving Ground, MD 21005		11 12. REPORT DATE DEC[redacted]76 13. NUMBER OF PAGES 80
14. MONITORING AGENCY NAME & ADDRESS *(if different from Controlling Office)* USA Materiel Development & Readiness Command 5001 Eisenhower Avenue Alexandria, VA 22333 12 79p.		15. SECURITY CLASS. *(of this report)* Unclassified 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT *(of this Report)*

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT *(of the abstract entered in Block 20, if different from Report)*

18. SUPPLEMENTARY NOTES

19. KEY WORDS *(Continue on reverse side if necessary and identify by block number)*

3-D Heat Transfer
Heat Conduction
Heat Diffusion
Monte Carlo Methods
Green's Functions
Homogeneous Boundary Conditions
Inhomogeneous Boundary Conditions

20. ABSTRACT *(Continue on reverse side if necessary and identify by block number)*

A Monte Carlo method has been developed to solve the time-dependent heat conduction or diffusion equation. This method is a generalization of the "floating sphere" method to rectangular parallelepipeds. Green's functions are developed which satisfy the appropriate boundary conditions. Probability distributions based on these Green's functions are determined and random walk techniques are used to estimate a solution. The method is capable of handling various boundary conditions and complex three-dimensional configurations.

Unclassified 390 389

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

TABLE OF CONTENTS

I. INTRODUCTION

A Monte Carlo method has been developed to solve the time-dependent heat conduction or diffusion equation. The method has been implemented in a versatile computer program, and applied to obtain the solution of heat conduction problems in complex geometry. Other possible applications include particle diffusion and neutron slowing down.

The Monte Carlo method is a generalization of the "floating sphere" method developed by Haji-Sheikh[1] and by Muller[2]. It is shown that the solution can be estimated by constructing a random walk based on the selection of position and time using probability distributions based on known Green's functions. These Green's functions satisfy appropriate boundary conditions on the surface of arbitrary volumes. They can be obtained for a wide class of volumes[3]. The form of the known solution simplifies if the volume is wholly contained within a homogeneous medium, but Haji-Sheikh's restriction of the volume to a sphere is not necessary. Our approach is to specialize the class of volumes to rectangular parallelepipeds of arbitrary size. This leads to an exact solution of the heat transfer problem if all the boundaries of the configuration are planar, and to a solution with any arbitrarily preset degree of accuracy if curved boundaries are involved.

The method provides the possibility of solving time-dependent heat conduction problems with internal heat sources and a variety of boundary conditions. The current program treats the various boundary conditions, but not internal heat sources. An exact treatment of the linear heat conduction problem is obtained. If the conduction properties depend on the local temperature, we linearize the problem by breaking up the calculation into small time steps, assuming no temperature dependence of the parameters during the time step.

Complex three-dimensional configurations can be treated. The geometrical description is of the Combinatorial Geometry[4] type. Geometrical bodies are defined in terms of intersections of quadratic surfaces. Geometrical regions are defined in terms of intersections of bodies.

II. THEORY

1. The Heat Conduction Equation

Let us consider the heat conduction equation:

$$-\frac{1}{\rho c}\nabla\cdot K\nabla T(x,t) + \frac{\partial T(x,t)}{\partial t} = \frac{1}{\rho c}Q(x,t) \qquad (1)$$

$T(x,t)$ is the temperature at point $\vec{x}$, time t. K is the thermal conductivity, ρ is the density, c the specific heat, and Q the heat source density. The problem is defined for $x\varepsilon\Omega$ where Ω is a volume surrounded by a surface Σ. K,ρ,c, are continuous functions of position, except across specified internal boundaries. The boundary conditions on T are specified as follows:

- Internal boundaries:

$$\left.\begin{array}{l} T \text{ continuous} \\ Kn\cdot\nabla T \text{ continuous} \end{array}\right\} \qquad (2)$$

- External boundary Σ

The external boundaries conditions we will consider are either:

$T(x,t)$ known for $x\varepsilon\Sigma$, $0\le t\le t_0$ (3a)

$$\text{or: } \left.\begin{array}{l} \vec{n}\cdot K\nabla T(x,t) = h(T_G(x,t) - T(x,t)) \\ h,\ T_G(x,t) \text{ known} \end{array}\right\} \text{for } x\varepsilon\Sigma,\ 0\le t\le t_0 \qquad (3b)$$

where $\vec{n}$ is the outward normal to the external boundary at point x, and h is the coefficient of surface heat transfer.

- Initial conditions:

$T(x,0)$ known for $x\varepsilon\Omega$ (4)

2. The Green's Function Method

The problem we consider consists of finding the temperature $T(x_0,t_0)$ at a point $x_0 \, \varepsilon\Omega$, $t_0>0$.

Let us surround the point x_0 by a volume $V\varepsilon\Omega$, the volume V being bounded by a surface S, and define a Green's function $G(x_0-x,t_0-t)$ for $x\varepsilon V$, $t\leq t_0$. It satisfies the differential equation

$$\nabla\cdot K\nabla \frac{1}{\rho c} G(x_0-x,t_0-t) + \frac{\partial G(x_0-x,t_0-t)}{\partial t} = 0 \, , \tag{5}$$

the boundary conditions are as follows:

- Internal boundaries - or portions (if any) of the internal boundaries of the full configuration which are within V:

$$\left.\begin{aligned} &\frac{1}{\rho c}\, G \text{ continuous} \\ &K\nabla \frac{1}{\rho c}\, G \text{ continuous} \end{aligned}\right\} \tag{6}$$

- External boundary S

$$G(x_0-x,t_0-t) = 0 \text{ for } x\varepsilon S \;,\; 0\leq t\leq t_0 \tag{7}$$

- Initial condition

$$G(x_0-x,0) = \delta(x-x_0) \tag{8}$$

A particular linear combination of Equations (1) and (5) can be written in the form

$$\begin{aligned} \int_0^{t_0} dt \int_V &\left\{ -T(x,t)\ \nabla K\nabla \frac{1}{\rho c}\, G(x_0-x,t_0-t) \right. \\ &\left. + G(x_0-x,t_0-t)\, \frac{1}{\rho c}\, \nabla K\nabla T(x,t) \right\} dV_x \\ &+ \int_0^{t_0} dt \int_V G(x_0-x,t_0-t)\, \frac{1}{\rho c}\, Q(x,t)\ dV_x \\ &= \int_0^{t_0} dt \int_V \left\{ T(x,t)\, \frac{\partial G(x_0-x,t_0-t)}{\partial t} \right. \\ &\left. + G(x_0-x,t_0-t)\, \frac{\partial T(x,t)}{\partial t} \right\} dV_x \end{aligned} \tag{9}$$

The first volume integral of the left hand side can be reduced to a surface integral by applying a generalized Green's theorem. The time integration of the right hand side can be performed. One obtains:

$$\int_0^{t_0} dt \int_S \left\{ -T(x,t)\, K\nabla \frac{1}{\rho c}\, G(x_0-x,t_0-t) \right.$$

$$\left. + G(x_0-x,t_0-t)\, \frac{1}{\rho c}\, K\nabla T(x,t) \right\} \cdot \vec{d}S_x$$

$$+ \int_0^{t_0} dt \int_V G(x_0-x,t_0-t)\, \frac{1}{\rho c}\, Q(x,t)\, dV_x \qquad (10)$$

$$= \int_V T(x,t_0)\, G(x_0-x,0)\, dV_x$$

$$- \int_V T(x,0)\, G\,(x_0-x,t_0)\, dV_x$$

The second term of the surface integral vanishes because of the boundary condition (7). The first volume integral of the right hand side can be evaluated taking the initial condition (8) into account. One therefore obtains, after rearranging the terms, and substituting $t=t_0-\tau$:

$$T(x_0,t_0) = \int_0^{t_0} d\tau \int_V G(x_0-x,\tau)\, \frac{1}{\rho c}\, Q(x,t_0-\tau)\, dV_x$$

$$+ \int_V T(x,0)\, G\,(x_0-x,t_0)\, dV_x \qquad (11)$$

$$+ \int_0^{t_0} d\tau \int_S -T(x,t_0-\tau)\, K\nabla \frac{1}{\rho c}\, G(x_0-x,\tau) \cdot \vec{d}S_x$$

Equation (11) can be considered as an integral equation for $T(x_0,t_0)$.

The choice of the volume V is arbitrary, provided $V\varepsilon\Omega$. If $V=\Omega$ and the corresponding G is known, the problem of finding $T(x_0,t_0)$ is reduced to quadratures (assuming that T is known on the external boundary $\Sigma=S$). In practice, the choice of V is limited to volumes for which the Green's function G is known, or readily computable, such as spheres, rectangular parallelepipeds, etc.

3. The Monte Carlo Method

A conceptually simple Monte Carlo technique can be constructed for the solution of Equation (11). Before describing it, a lemma has to be proven.

Let us consider an arbitrary volume V, with no internal heat sources ($Q=0$), and boundary conditions $T(x,t)=1$ for $x\varepsilon S$, $t>0$ and $T(x,0)=1$ for $x\varepsilon V$. The solution is then $T(x,t)=1$ for any internal point x, at any time $t>0$, in particular at x_0,t_0.

Substituting this into Equation (11) we obtain:

$$1 = \int_V G(x_0-x,t_0)dV_x + \int_0^{t_0} d\tau \int_S -K\nabla \frac{1}{\rho c} G(x_0-x,\tau)\cdot d\vec{S}_x \tag{12}$$

Letting $t_0\to\infty$ in Equation (12) we get:

$$1 = \int_0^\infty d\tau \int_S -K\nabla \frac{1}{\rho c} G(x_0-x,\tau)\cdot d\vec{S}_x$$

as $\lim_{t_0\to\infty} \left\{ \int_V G(x_0-x,t_0)dV_x \right\} \to 0$

Substituting that result into Equation (12) we obtain:

$$\int_{t_0}^\infty d\tau \int_S -K\nabla \frac{1}{\rho c} G(x_0-x,\tau)\cdot d\vec{S}_x = \int_V G(x_0-x,t_0)\ dV_x \tag{13}$$

A Monte Carlo algorithm for the solution of Equation (11) is then as follows.

First estimate the time-volume integral involving internal sources (if any). For estimating the remaining two terms, first select a time $\tau>0$ from

$$p(\tau)d\tau = \int_S -K\nabla \frac{1}{\rho c} G(x_0-x,\tau)\cdot d\vec{S}\ d\tau \tag{14}$$

If $\tau>t_0$ (which, according to 13, happens with probability $\int_V G(x_0-x,t_0)dV_x$), the second term of Equation (11) has to be sampled: sample a point x in V with density proportional to $G(x_0-x,t_0)$, and score the (known) T(x,0) as a contribution to $T(x_0,t_0)$ - the history terminates. If $\tau<t_0$, the last term of Equation (11) has to be sampled. Sample $-K\nabla \frac{1}{\rho c} G(x_0-x,\tau)\cdot\vec{n}$ for a point x on S. ($\vec{n}$ is the outward normal to S at x; $T(x,t_0-\tau)$ is the estimate for $T(x_0,t_0)$. Two cases can occur: in one case, the sample x is on a common part of S and Σ, then $T(x,t_0-\tau)$ is either known (if Equation (3a) applies) or obtainable (if Equation (3b) applies) as described in Section II.5 below, and the random walk terminates. In the other case, x is on S but internal to Σ. The point x is then surrounded by another volume $V\varepsilon\Omega$, and the procedure to estimate $T(x,t_0-\tau)$ is identical to that just described to estimate $T(x_0,t_0)$. In all cases the random walk terminates when a known temperature is encountered (either at t=0 or on the boundary).

4. Inhomogeneous External Boundary Conditions

The case of inhomogeneous boundary conditions can be treated in the same fashion as the case of known temperature condition, provided some adjustments are made.

If, at any step of the random walk, the surface S surrounding V has a part S_2 ($S=S_1 + S_2$) in common with a portion of Σ where inhomogeneous boundary conditions apply, the boundary conditions imposed on G (Equation 7) either have to, or can be modified to:

$$G(x_0-x,t_0-t) = 0 \text{ for } x\varepsilon S_1,\ 0\le t\le t_0 \qquad (15a)$$

$$\vec{n}\cdot K\nabla \frac{1}{\rho c} G(x_0-x,t_0-t) = - h \frac{1}{\rho c} G(x_0-x,t_0-t)$$

$$\text{for } x\varepsilon S_2,\ 0\le t\le t_0 \qquad (15b)$$

Equations (9) and (10) are still valid. The first term of Equation (10) is rewritten in shorthand notation as:

$$F = \int_0^{t_0} dt \int_S \left\{ -T \; K\nabla \frac{1}{\rho c} G + G \frac{1}{\rho c} K\nabla T \right\} \cdot \vec{ds}_n$$

$$= \int_0^{t_0} dt \int_{S_1} \left\{ -T \; K\nabla \frac{1}{\rho c} \; G + G \frac{1}{\rho c} K\nabla T \right\} \cdot \vec{ds}_n$$

$$+ \int_0^{t_0} dt \int_{S_2} \left\{ -T \; K\nabla \frac{1}{\rho c} G + G \frac{1}{\rho c} K\nabla T \right\} \cdot \vec{ds}_n$$

The second term of the S_1 - integral vanishes because of the boundary condition (15a). In the S_2 -integral, we replace T by its expression obtained from (3b), and G by its expression obtained from (15b). We obtain

$$F = \int_0^{t_0} dt \int_{S_1} -T \; K\nabla \frac{1}{\rho c} G \cdot \vec{ds} + \int_0^{t_0} dt \int_{S_2} \left\{ -(T_G - \frac{1}{h} \vec{n} \cdot K\nabla T) K \; \frac{1}{\rho c} G \right.$$

$$\left. - (\frac{\rho c}{h} K\nabla \frac{1}{\rho c} G) \frac{1}{\rho c} K\nabla T \right\} \cdot \vec{ds}$$

Combining terms, one is left with

$$F = \int_0^{t_0} dt \left\{ \int_{S_1} -T \; K\nabla \frac{1}{\rho c} G \cdot \vec{ds} + \int_{S_2} -T_G \; K\nabla \frac{1}{\rho c} G \cdot \vec{ds} \right\}$$

The equivalent of Equation (11) is then

$$T(x_0,t_0) = \int_0^{t_0} d\tau \int_V G \frac{1}{\rho c} Q \; dV + \int_V T(x,0) \; G \; dV$$
$$+ \int_0^{t_0} d\tau \left\{ \int_{S_1} -T \; K\nabla \frac{1}{\rho c} G \cdot \vec{ds} + \int_{S_2} -T_G \; K\nabla \frac{1}{\rho c} G \cdot \vec{ds} \right\} \qquad (16)$$

Equation (11) collapses if x_0 is on S. Equation (16) collapses if x_0 is on S_1, but is well-behaved if x_0 is on S_2 (provided $h<\infty$). This shows that one may use Equation (11) (with simpler Green's functions) until the necessity occurs to estimate the temperature <u>at</u> an external boundary where inhomogeneous boundary conditions are imposed. At that point, one has to switch to the more complex boundary condition (15b), which leads to Equation (16).

The Monte Carlo procedure corresponding to Equation (16) is to estimate the internal heat source term (if any) first. Then select τ from Equation (14). If $\tau>t_0$, sample the second term and terminate the history. If $\tau<t_0$, sample the surface terms. If $x\varepsilon S_2$, estimate T_G and terminate history. If $x\varepsilon S_1$, estimate $T(x,t_0-\tau)$ either as a known temperature, or by estimating it using Equation (11).

5. Linearization of a Non-Linear Conduction Equation

The heat conduction equation (1) is non-linear if the thermal properties (K,ρ,c) depend on the temperature T. Consider a(T) which depends explicitly on temperature only.

Let us introduce the new variable

$$\Theta = \int_0^T a(T)dT \tag{17}$$

It follows from (17) that

$$\frac{\partial\Theta}{\partial t} = a\frac{\partial T}{\partial t} \quad , \nabla\Theta = a\,\nabla T \tag{18}$$

Substituting (18) into (1) we obtain

$$-\frac{1}{\rho c}\,\nabla\,\frac{K}{a}\nabla\Theta + \frac{1}{a}\frac{\partial\Theta}{\partial t} = \frac{1}{\rho c}\,Q$$

or:

$$-\frac{a}{\rho c}\nabla\,\frac{K}{a}\nabla\Theta + \frac{\partial\Theta}{\partial t} = \frac{a}{\rho c}\,Q \tag{19}$$

If the problem at hand is such that one can choose an "a" so that K/a is time independent and position independent in any region (thus exhibiting only discontinuous variation across specified boundaries), Equation (19) can be simplified to

$$- \frac{K}{\rho c} \nabla^2 \Theta + \frac{\partial \Theta}{\partial t} = \frac{a}{\rho c} Q \tag{20}$$

In practice this situation occurs in problems involving a single homogeneous material. If the configuration is heterogeneous, we propose to choose a value of a such that K/a is constant for the most important material. Regions involving other materials can be subdivided into small enough subregions so that K/a does not vary appreciably. In that case, Equation (20) holds as a reasonable approximation.

In any case, the diffusivity $K/\rho c$ is still temperature (and *therefore* Θ)-dependent. We propose to subdivide the entire configuration into small enough regions so that the diffusivity $K/\rho c$ can be reasonably approximated as being constant within each region.

The time-dependence of the diffusivity can also be handled approximately by splitting the time span from $t=0$ to $t=t_0$ into time steps short enough so that the thermal parameters do not vary appreciably over each time step. Knowing the temperature at the beginning of the time step, the thermal parameters can be calculated and considered as constant throughout the time step. The temperature has to be calculated everywhere at the end of the time step - and the new temperature distribution can serve as initial conditions for the next time step.

If the time span has been subdivided into small enough time bins, and the homogeneous regions have been subdivided into small enough subregions as discussed above, Equation (20) applies with:

$$K/a = \text{const}$$
$$\rho c/a = \text{const}$$

in each subregion and in each time bin.

Equation (20) is identical in form to Equation (1) provided one makes the following substitutions:

$$\begin{aligned} &\Theta \text{ by } T \\ &K/a \text{ by } K \qquad\qquad (21) \\ &\rho c/a \text{ by } \rho c \end{aligned}$$

The only remaining problem is that of treating the inhomogeneous boundary condition (3b). It is known[3] that the transformation (17) cannot be performed exactly in the presence of such inhomogeneous boundary conditions. One can, however, write the boundary condition in the form

$$\frac{K}{a}\frac{\partial\Theta(x,t)}{\partial n} = \frac{h}{a}\left[\tilde{\Theta}_G(x,t) - \Theta(x,t)\right] \qquad (22)$$

where:

$$\tilde{\Theta}_G(x,t) = \Theta(x,t) + a(T_G(x,t) - T(x,t)) \qquad (23)$$

The problem is that $\tilde{\Theta}_G(x,t)$ is not a known function, as it involves the unknowns $\Theta(x,t)$ and $T(x,t)$. We propose, however, the following approximation, which is in the spirit of all those made in this section:

$$\tilde{\Theta}_G(x,t) = \Theta(x,0) + a(T_G(x,t) - T(x,0)) \qquad (24)$$

Equation (22) is then identical to Equation (3b) when substitutions (21) together with the replacement of

$$\begin{aligned} &h/a \text{ by } h \\ & \qquad\qquad\qquad (25) \\ &\tilde{\Theta} \text{ by } T_G \end{aligned}$$

are made.

III. GREEN'S FUNCTIONS IN HOMOGENEOUS RECTANGULAR PARALLELEPIPEDS

The Monte Carlo method described in Section II is valid for any volume V surrounding x_0, provided V is entirely within the configuration under examination. In practice, the choice of volumes V is limited to such shapes for which the Green's functions are known or easily computable. Muller[1] considered a variety of shapes for the solution of the Dirichlet problem. Haji-Sheikh[2] considered only spheres for the solution of heat conduction problems. Carslaw and Jaeger[3] give Green's functions for a variety of shapes.

In this section, we give all the relevant properties of Green's functions defined over rectangular parallelepipeds. The restriction of volumes V to be rectangular parallelepipeds permits an exact solution in the case of configurations with piece-wise planar boundaries, or solutions to an arbitrary degree of accuracy if curved boundaries are involved. Throughout this section, we assume that K and the combination ρc are constant, and introduce the coefficient of diffusivity

$$D = K/\rho c. \qquad (1)$$

1. Separation of Variables

We propose to solve for $G(x_1,x_2,x_3,t)$ which satisfies

$$D\left[\frac{\partial^2 G}{\partial x_1^2} + \frac{\partial^2 G}{\partial x_2^2} + \frac{\partial^2 G}{\partial x_3^2}\right] - \frac{\partial G}{\partial t} = 0 \qquad (2)$$

for $x \varepsilon V$, where V is defined by

$$\alpha_i^- \leq x_i \leq \alpha_i^+ \quad , \; i = 1,3$$

and $t>0$,

with the boundary and initial conditions

$$G(x_1,x_2,x_3,0) = \delta(x_1)\delta(x_2)\delta(x_3) \tag{3}$$

$$G(x_1,x_2,x_3,t) = -\beta_i^{\pm}\frac{\partial G(x_1,x_2,x_3,t)}{\partial x_i}, \quad x_i = \alpha_i^{\pm}, \; i=1,3 \tag{4}$$

where $\alpha_i^{\pm}$, $\beta_i^{\pm}$ are given constants:

$$\pm\alpha_i^{\pm} \geq 0 \qquad i = 1,3 \tag{5}$$

$$\pm\beta_i^{\pm} \geq 0 \qquad i = 1,3 \tag{6}$$

The inequalities (5) insure that the point $x_i=0$ is inside V. If all $\beta_i^{\pm}$ are zero, the boundary condition (4) is tailored to that of Equation (II.7). Positive values of $\beta_i^{\pm}$ can be chosen to reproduce the boundary condition (II.15b).

Let us assume that the solution G can be written in the form

$$G(x_1,x_2,x_3,t) = X_1(x_1,t)X_2(x_2,t)X_3(x_3,t) \tag{7}$$

Substituting (7) into (2) we obtain

$$X_1X_2X_3 \sum_{i=1}^{3} \frac{1}{X_i}\left[D\frac{\partial^2 X_i}{\partial x_i^2} - \frac{\partial X_i}{\partial t}\right] = 0 \tag{8}$$

A solution of (2) with the conditions (3), (4) is therefore (7) with

$$D\frac{\partial^2 X_i}{\partial x_i^2} - \frac{\partial X_i}{\partial t} = 0 \qquad i = 1,3 \tag{9}$$

with the initial and boundary conditions

$$X_i(x_i,0) = \delta(x_i) \qquad i = 1,3 \tag{10}$$

$$X_i(x_i,t) = -\beta_i^{\pm}\frac{\partial X_i(x_i,t)}{\partial x_i}, \quad x_i = \alpha_i^{\pm}, \; i = 1,3 \tag{11}$$

The four-dimensional problems (2-4) have been reduced to three independent two-dimensional problems (9-11).

2. Sampling the RPP Green's Function

As discussed in Section II.3, one first needs to sample a time t from the distribution (Equation I.13-14)

$$\int_t^\infty p(t)dt = \int_V G(X_1,X_2,X_3,t)\ dx_1 dx_2 dx_3 \qquad (12)$$

Using the expression (7) for G, we obtain

$$\int_t^\infty p(t)dt = \int_{\alpha_1^-}^{\alpha_1^+} X_1(x_1,t)\ dx_1 \int_{\alpha_2^-}^{\alpha_2^+} X_2(x_2,t)\ dx_2 \int_{\alpha_3^-}^{\alpha_3^+} X_3(x_3,t)\ dx_3$$

$$= F_1(t)\ F_2(t)\ F_3(t) \qquad (13)$$

where

$$F_i(t) = \int_{\alpha_i^-}^{\alpha_i^+} X_i(x_i,t)\ dx_i\ , \quad i = 1,3 \qquad (14)$$

To sample the cumulative distribution (13) we can sample each of the three cumulative distributions (14) and retain the smallest sample.

Indeed, the probability density function (pdf) of the retained sample is:

$$p(t)dt = \frac{dF_1}{dt} F_2F_3 + F_1 \frac{dF_2}{dt} F_3 + F_1\ F_2 \frac{dF_3}{dt} \qquad (15)$$

$$= \frac{d}{dt} \left[F_1F_2F_3\right] \qquad \text{QED.}$$

As further discussed in Section II.3, the sample t has to be compared to a given cutoff time t_0. If $t>t_0$, a position x_1,x_2,x_3 has to be sampled from a pdf proportional to $G(x_1,x_2,x_3,t_0)$. As G is separable, the problem is reduced to sampling x_i from $X_i(x_i,t_0)$, for i=1,2,3.

If the sample t is less than t_0, a point x_1,x_2,x_3 has to be sampled on S from a pdf proportional to $-D\frac{\partial}{\partial n} G(x_1,x_2,x_3,t)$. The probability of such a point being on either of the two sides of the parallelepiped which are parallel to the x_2,x_3 plane is:

$$P_1 = D\left[-\frac{\partial X_1}{\partial x_1}(-\alpha_1^-,t) + \frac{\partial X_1}{\partial x_1}(\alpha_1^+,t)\right] \int_{\alpha_2^-}^{\alpha_2^+} X_2(x_2,t)dx_2 \int_{\alpha_3^-}^{\alpha_3^+} X_3(x_3,t)dx_3$$

$$= \left[D\int_{\alpha_1^-}^{\alpha_1^+} \frac{\partial^2 X_1(x_1,t)}{\partial x_1{}^2}dx_1\right] F_2(t)F_3(t)$$

Using Equation (9) we obtain

$$P_1 = \left[\int_{\alpha_1^-}^{\alpha_1^+} \frac{\partial}{\partial t} X_1(x_1,t)dx_1\right] F_2(t)F_3(t) = \frac{dF_1(t)}{dt} F_2(t)F_3(t)$$

The probability P_1 of selecting a point on either one of the pair of sides 1 is therefore proportional to having selected the smallest time t from the cumulative distribution $F_1(t)$ (see Equation 15). This can easily be generalized to the pairs of sides 2 and 3.

Having selected i, (say i=1), x_1 is set to α_1^- or α_1^+ with probability $-\frac{\partial X_1}{\partial x_1}(\alpha_1^-,t)$ or $\frac{\partial X_1}{\partial x_1}(\alpha_1^+,t)$, respectively. The remaining two coordinates (i=2,3) have to be sampled from $X_i(x_i,t)$.

3. The One-Dimensional Green's Function Vanishing at Boundaries

We now consider the case of Green's functions with G=0 boundary conditions at both boundaries.

The Green's function satisfies

$$D \frac{\partial^2 X(x,t)}{\partial x^2} - \frac{\partial X(x,t)}{\partial t} = 0 \qquad (16)$$

with the boundary and initial conditions

$$X(x,0) = \delta(x) \qquad (17)$$

$$X(\pm a/2,t) = 0 \qquad (18)$$

therefore treating the symmetric case $\alpha^+ = -\alpha^- = a/2$.

Let us introduce the reduced variables

$$\xi = x/a$$

$$\tau = Dt/a^2 \qquad (19)$$

$$G(\xi,\tau) = X(x,t)$$

Equation (16-18) become

$$\frac{\partial^2 G(\xi,\tau)}{\partial \xi^2} - \frac{\partial G(\xi,\tau)}{\partial \tau} = 0 \qquad (20)$$

$$G(\xi,0) = \delta(\xi) \qquad (21)$$

$$G(\pm 1/2,t) = 0 \qquad (22)$$

The solution G can be expressed either in the eigenfunction expansion:

$$G(\xi,\tau) = 2 \sum_{n=0}^{\infty} \cos((2n+1)\pi\xi) \exp\left[-(2n+1)^2\pi^2\tau\right], \qquad (23)$$

or in the image expansion:

$$G(\xi,\tau) = \frac{1}{2\sqrt{\pi\tau}} \sum_{n=-\infty}^{+\infty} (-1)^n \exp\left[-(\xi+n)^2/4\tau\right]. \qquad (24)$$

The expansion (23) converges rapidly for large τ, whereas expansion (24) converges rapidly for small τ.

4. Monte Carlo Algorithms for Sampling One-Dimensional Green's Functions Vanishing at Boundaries

The cumulative time distribution corresponding to Equation (14) is given by

$$F(\tau) = \int_{-1/2}^{1/2} G(\xi,\tau)d\xi \tag{25}$$

Using the eigenfunction expansion (23) we obtain:

$$F(\tau) = \frac{4}{\pi} \sum_{n=0}^{\infty} \frac{(-1)^n}{2n+1} \exp\left[-(2n+1)^2\pi^2\tau\right] \tag{26}$$

corresponding to a probability density function

$$-\frac{dF}{d\tau} = 4\pi \sum_{n=0}^{\infty} (-1)^n(2n+1) \exp\left[-(2n+1)^2\pi^2\tau\right] \tag{27}$$

The expansion (27) is absolutely convergent for $\tau>0$, and the absolute values of the terms are monotonically decreasing for $\tau>\tau^{(1)}$, where

$$\tau^{(1)} = \ell n(3)/8\pi^2 = 0.0139$$

At early times, an approximation is suggested: keep only the terms $n=0$ and $n=\pm 1$ in the expansion (24). Substituting this expression into (25) one obtains

$$F(\tau) = 1 - \frac{4}{\sqrt{\pi}} \int_{1/4\sqrt{\tau}}^{\infty} e^{-u^2} du \tag{28}$$

Compared to Equation (26), Equation (28) gives at least five place accuracy for $\tau<0.075$.

To sample a time, one can set a time breakpoint τ_1 $(0.014<\tau_1<0.075)$ and use the early time approximation for $\tau<\tau_1$ and the eigenfunction expansion for $\tau>\tau_1$. Optimum computer times are achieved by setting $\tau_1=0.05$. At that time, only the first two terms of the eigenfunction expansion are non-negligible.

The Monte Carlo algorithm consists in precalculating $F(\tau_1)$. A random number ξ is sampled. If $\xi > F(\tau_1)$, the early time case applies. If $\xi < F(\tau_1)$ the late time case applies.

4.1 Early Time Case

One first has to select $\tau < \tau_1$ from Equation (28). Let $u_1 = 1/4\sqrt{\tau_1}$. Select $u > u_1$ from $e^{-u^2} du$ and set $\tau = 1/(4u)^2$.

As u_1 is of the order of unity, an efficient technique to select u is to select δ from $\exp(-2u_1\delta)$ and accept the sample with probability $\exp(-\delta^2)$. When δ is accepted, u is set equal to $u_1+\delta$.

Once τ has been selected, the coordinate x has to be sampled.

If diffusion occurs to a boundary, it occurs with equal probability to either boundary as

$$\frac{\partial G}{\partial x}(1/2,\tau) = -\frac{\partial G}{\partial x}(-1/2,\tau) \tag{29}$$

If an internal point ξ has to be sampled, its density is proportional to $G(\xi,\tau)$ as obtained from Equation (24), which can be rewritten in the form

$$G(\xi,\tau) = (4\pi^2\tau)^{-1/2} \sum_{n=0}^{\infty} \left\{ \exp\left[-(\xi+2n)^2/4\tau\right] - \exp\left[-(\xi-2n-1)^2/4\tau\right] - \exp\left[-(\xi+2n+1)^2/4\tau\right] + \exp\left[-(\xi+2n+2)/4\tau\right] \right\} \tag{30}$$

The selection is performed as follows:

First $\eta > 0$ is sampled from $\exp(-\eta^2/4\tau)$. Then the largest integer n is found such that

$$\eta = 2n + \xi \ , \quad \xi \geq 0$$

If $\xi > 1/2$, the sample η is rejected.

When η is accepted, the samples n, ξ are tentatively accepted as samples of the first of the four terms constituting the n-th term of Equation (30). The remaining three terms are taken into account by a rejection technique:

Let
$$r_1 = \exp\left\{- \left[(\xi-2n-1)^2 - (\xi-2n)^2\right] /4\tau\right\}$$

$$r_2 = \exp\left\{- \left[(\xi+2n+1)^2 - (\xi-2n)^2\right] /4\tau\right\}$$

$$r_3 = \exp\left\{- \left[(\xi+2n+2)^2 - (\xi-2n)^2\right] /4\tau\right\}$$

The sample ξ is accepted with probability $1-r_1-r_2+r_3$. In practice, this is broken up into three steps. Only r_1 is calculated, and the sample is accepted with probability $1-r_1$. If not accepted, r_2 and r_3 are calculated, and the remaining tests are performed.

4.2 Late Time Case

One first has to select $\tau<\tau_1$ from Equation (27). Keeping only the first two terms of the expansion, one obtains:

$$\frac{dF}{d\tau}\, d\tau = 4\pi \left[\exp(-\pi^2\tau) - 3\exp(-9\pi^2\tau)\right] d\tau \qquad (31)$$

Let $x = \exp(-\pi^2(\tau-\tau_1))$

$\varepsilon = \exp(-8\pi^2\tau_1)$

$p(x)dx \propto (1-3\varepsilon x^8)dx$

or

$$p(x)dx = \frac{1-3\varepsilon}{1-\varepsilon/3}\, dx + \frac{(8/3)\varepsilon}{1-\varepsilon/3}\, \frac{9}{8}\, (1-x^8)dx$$

To sample x, with probability $(1-3\varepsilon)/(1-\varepsilon/3)$ one samples the first term by setting x equal to a random number. With remaining probability, $(8/3)\varepsilon/(1-\varepsilon/3)$, the second term needs to be sampled. x can be set equal to any of nine random numbers except for the largest of these nine. This technique turns out to be time consuming. A more efficient technique consists in setting x to a random number, and accept x with probability $1-x^8$. If x is not accepted, it is mulitplied by another random number. The product is a valid sample x.

Finally, the time variable is set to

$$\tau = \tau_1 - \ln(x)/\pi^2.$$

Once τ has been selected, the coordinate x has to be sampled.

If diffusion occurs to a boundary, it occurs with equal probability to either boundary, as Equation (29) still applies.

If an internal point ξ has to be sampled, its density is proportional to $G(\xi,\tau)$ as obtained from Equation (23), keeping only the first two terms

$$\begin{aligned} G(\xi,\tau)d\xi &= 2\left[\cos(\pi\xi)\exp(-\pi^2\tau) + \cos(3\pi\xi)\exp(-9\pi^2\tau)\right]d\xi \\ &\propto \cos(\pi\xi) + \varepsilon\cos(3\pi\xi) \qquad (32) \\ &\propto \cos(\pi\xi) + \varepsilon\left[4\cos^3(\pi\xi) - 3\cos(\pi\xi)\right] \end{aligned}$$

where $\varepsilon = \exp(-8\pi^2\tau)$.

Substituting $\sin(\pi\xi) = 2x-1$ into equation (32):

$$p(x)dx \propto (1-3\varepsilon)dx + 16\varepsilon x(1-x)dx$$

$$p(x)dx = \frac{1-3\varepsilon}{1-\varepsilon/3}dx + \frac{(8/3)\varepsilon}{1-\varepsilon/3}6x(1-x)dx \qquad (33)$$

To sample equation (33), with probability $(1-3\varepsilon)/(1-\varepsilon/3)$ set x to a random number. With remaining probability, $(8/3)\varepsilon/(1-\varepsilon/3)$, sample three random numbers and set x equal to the one in the middle in the order of magnitude.

5. The One-Dimensional Green's Function with Homogeneous Boundary Conditions

We seek the solution of the equation

$$D\frac{\partial^2 X(x,t)}{\partial x^2} - \frac{\partial X(x,t)}{\partial t} = 0 \qquad (34)$$

with the boundary and initial conditions

$$X(x,0) = \delta(x) \qquad (35)$$

$$X(0,t) = \beta\frac{\partial X(0,t)}{\partial x} \qquad (36)$$

$$X(1,t) = 0 \qquad (37)$$

therefore treating the case of the source point x=0 on the boundary x=0: $\alpha^-=0$, $\alpha^+=1$, with homogeneous boundary condition at x=0 ($\beta^->0$) and zero boundary condition at x=a ($\beta^+=0$).

Let us again introduce the reduced variables (19):

$$\begin{aligned} \xi &= x/a \\ \tau &= Dt/a^2 \\ G(\xi,\tau) &= X(x,t) \\ \gamma &= \beta/a \end{aligned} \qquad (38)$$

Equations (34-37) become:

$$\frac{\partial^2 G(\xi,\tau)}{\partial \xi^2} - \frac{\partial G(\xi,\tau)}{\partial \tau} = 0 \qquad (39)$$

$$G(\xi,0) = \delta(\xi) \qquad (40)$$

$$G(0,\tau) = \gamma \frac{\partial G(0,\tau)}{\partial \xi} \qquad (41a)$$

$$G(1,\tau) = 0 \qquad (41b)$$

The eigenfunction expansion:

$$G(\xi,\tau) = \sum_{n=1}^{\infty} C_n \sin(\alpha_n(\xi-1)) \exp(-\alpha_n^{\,2}\tau) \qquad (42)$$

satisfies the differential equation (39) and boundary condition (41a), (41b). Substituting (42) into (41a) we obtain the eigenvalue equation:

$$\tan \alpha_n = - \gamma\alpha_n \text{ , } n\geq 1 \qquad (43)$$

Taking (43) into account, one can show that

$$\begin{aligned} \int_0^1 \sin(\alpha_n(\xi-1)) \sin(\alpha_m(\xi-1))\, d\xi \\ &= (1+\gamma\cos^2\alpha_n)/2 \text{ for } m=n \\ &= 0 \text{ for } m\neq n \end{aligned} \qquad (44)$$

Equation (44) shows that the expansion (42) is in terms of an orthogonal set. A necessary condition for (40) to be satisfied is

$$\int_0^1 \sin(\alpha_n(\xi-1))\delta(\xi)d\xi = \int_0^1 \sin(\alpha_n(\xi-1))\ G(\xi,0)d\xi \qquad (45)$$

Substituting (42) into (45) and taking (44) into account, one obtains

$$C_n = \frac{-2 \sin\alpha_n}{1+\gamma\cos^2\alpha_n} \qquad (46)$$

Expression (42) is therefore the solution of (39)-(41) provided (44) and (46) are satisfied.

5.1 Solution of the Eigenvalue Problem

The solution of Equation (44) is graphically represented in Figure 1. The solutions can be written as:

$$\alpha_n = n\pi - \eta_n \tag{47}$$

where

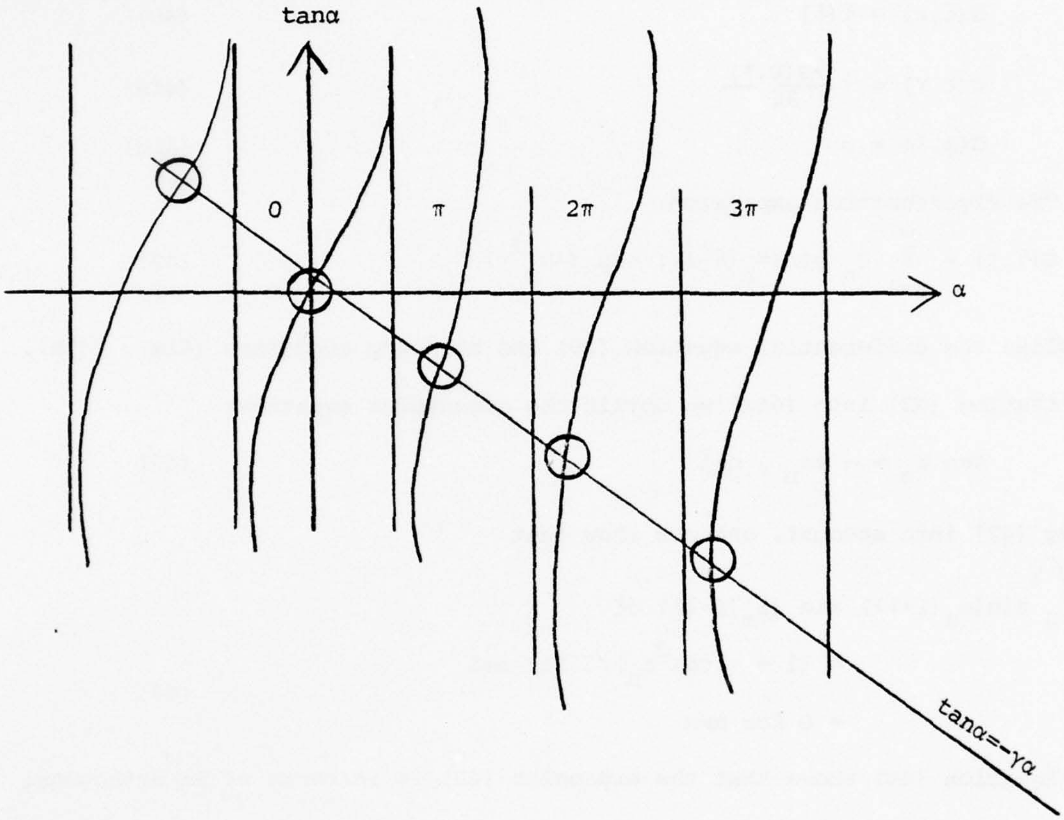


Figure 1 - Solution of tanα=-γα

η_n is a solution of

$$\tan(-\eta_n) = -\gamma(n\pi-\eta_n) \tag{48}$$

for n = 1, 2, ...

Equation (48) can be rewritten as

$$\tan \eta_n = \gamma n\pi - \gamma\eta_n \tag{49}$$

or

$$\eta_n = \text{Arctan } (\gamma n\pi - \gamma\eta_n) \tag{50}$$

$$n = 1,2, \ldots$$

Equation (50) can be used for an iterative method of solution, where the mth iteration can be obtained from the (m-1)th using

$$\eta_n^{m} = \text{Arctan } (\gamma n\pi - \gamma\eta_n^{m-1}) \tag{51}$$

The iteration can be started by using an approximate expression for $\tan \eta_n^0 \approx \eta_n^0/(\pi/2 - \eta_n^0)$ and substituting in Equation (49):

$$\eta_n^0/(\pi/2 - \eta_n^0) = \gamma n\pi - \gamma\eta_n^0 \tag{52}$$

$$\eta_n^0 = \left(\gamma n\pi + \gamma\pi/2+1 - \sqrt{(\gamma n\pi + \gamma\pi/2+1)^2 - 2n\gamma^2\pi^2} \right)/2\gamma \tag{53}$$

Recapitulating the results of the present section, the eigenvalues α_n can be obtained by using the starting value (53), iterating to convergence using Equation (51), and substituting the solution η_n into Equation (47).

For $\gamma>0$ and n large enough $\alpha_n \rightarrow (n-1/2)\pi$.

5.2 Early Time Approximation

The expansion (43) is rapidly converging for large values of τ. It does not converge for $\tau=0$. In order to derive an approximation to $G(\xi,\tau)$ for small values of τ, let us first solve a problem satisfying Equations (39)-(41a), and replacing Equation (41b) by

$$G(\xi,\tau) \rightarrow 0 \quad \text{as } \xi\rightarrow\infty \tag{54}$$

and denoting its solution $G(\xi,t)$ as $G_0(\xi,\tau)$.

Let us define the Laplace transformation of G_0.

$$F(\xi,\omega) = \int_0^\infty e^{-\tau} G_0(\xi,\tau)d\tau \qquad (55)$$

It satisfies the differential equation

$$\frac{\partial^2 F}{\partial \xi^2} - \omega F = 0 \qquad (56)$$

The general solution of (56) satisfying the boundary conditions (40) and (54) is

$$F(\omega) = \frac{e^{-\xi\sqrt{\omega}}}{\sqrt{\omega}} + \int_0^\infty d\eta\ f(\eta)\ \frac{e^{-(\xi+\eta)\sqrt{\omega}}}{\sqrt{\omega}} \qquad (57)$$

the function $f(\eta)$ is to be determined from the boundary condition (41a):

$$\frac{e^{-\xi\sqrt{\omega}}}{\sqrt{\omega}} + \int_0^\infty d\eta\ f(\eta)\ \frac{e^{-(\xi+\eta)\sqrt{\omega}}}{\sqrt{\omega}} = \gamma\left[e^{-\xi\sqrt{\omega}} + \int_0^\infty d\eta\ f(\eta) e^{-(\xi+\eta)\sqrt{\omega}}\right]$$

for $\xi = 0$, or:

$$\int_0^\infty d\eta\ f(\eta) e^{-\eta\sqrt{\omega}} = \frac{\gamma - 1/\sqrt{\omega}}{\gamma + 1/\sqrt{\omega}} = 1 - \frac{2}{1+\gamma\sqrt{\omega}} \qquad (58)$$

The left hand side of Equation (58) defines a Laplace transform of $f(\eta)$, with the transform variable $s=\sqrt{\omega}$. The inverse transform of the right hand side is $\delta(\eta) - \frac{2}{\gamma} e^{-\eta/\gamma}$.

Therefore

$$f(\eta) = \delta(\eta) - \frac{2}{\gamma} e^{-\eta/\gamma} \qquad (59)$$

Substituting this into Equation (57) we obtain

$$F(\xi,\omega) = \frac{2}{\sqrt{\omega}} e^{-\xi\sqrt{\omega}} - \frac{2}{\gamma}\int_0^\infty e^{-\eta/\gamma} \frac{e^{-\xi\sqrt{\omega}}}{\sqrt{\omega}} d\eta$$

The inverse transform of F with respect to ω is

$$G_0(\xi,\tau) = \frac{1}{\sqrt{\pi\tau}}\left[e^{-\xi^2/4\tau} - \frac{1}{\gamma}\int_0^\infty d\eta\ e^{-\eta/\gamma}\ e^{-(\xi+\eta)^2/4\tau}\right] \tag{60}$$

The function

$$G(\xi,\tau) = G_0(\xi,\tau) - G_0(2-\xi,\tau) \tag{61}$$

satisfies the differential equation (39), as well as the boundary conditions (40) and (42). For small enough τ, $G_0(2,\tau) << G_0(0,\tau)$, and therefore from (61): $G(0,\tau) \approx G_0(0,\tau)$; for small enough τ, expression (61) approximately satisfies boundary condition (41).

The function $G(\xi,\tau)$ defined by Equation (61) can therefore be considered as an approximation to the solution of Equation (39), (41b).

6. Monte Carlo Algorithms for Sampling One-Dimensional Green's Functions with Homogeneous Boundary Conditions

Starting from the exact eigenfunction expansion (43), we obtain the following expressions for distributions of conduction time.

The p.d.f. of time passage through the boundary at which $G = -\gamma \frac{\partial G}{\partial n}$ is:

$$p_0(\tau) = -\frac{\partial G_0}{\partial n} = +\frac{\partial G(0,\tau)}{\partial \xi} = \sum_n \frac{2\alpha_n \sin\alpha_n}{1+ \gamma\cos^2\alpha_n} e^{-\alpha_n^2\tau} \tag{62}$$

For the boundary at which G=0, it is:

$$p_1(\tau) = -\frac{\partial G_1}{\partial n} = -\frac{\partial G(1,\tau)}{\partial x} = \sum_n \frac{2\sin\alpha_n(-\cos\alpha_n)_n}{1 + \gamma\cos^2\alpha_n} e^{-\alpha_n^2\tau} \tag{63}$$

$$F_0(\tau) = \int_\tau^\infty p_0(\tau)d\tau = \sum_n \frac{2\sin\alpha_n}{\alpha_n(1+\gamma\cos^2\alpha_n)} e^{-\alpha_n^2\tau} \tag{64}$$

and

$$F_1(\tau) = \int_\tau^\infty p_1(\tau)d\tau = \sum_n \frac{2\sin\alpha_n(-\cos\alpha_n)}{\alpha_n(1+\gamma\cos^2\alpha_n)} e^{-\alpha_n^2\tau} \tag{65}$$

For $\tau=0$, the following equalities can be derived:

$$F_0(\tau) = \frac{1}{(1+\gamma)} \tag{66}$$

$$F_1(\tau) = 1-F_0(\tau) \tag{67}$$

On the other hand, starting from the approximate early time expression of $G(\xi,\tau)$, we obtain the following:

$$F_0(\tau) = \frac{\gamma}{1+\gamma} - 2\left[\int_1^\infty \frac{e^{-\xi^2/4\tau}}{\sqrt{\pi\tau}} d\xi - \frac{1}{\gamma}\int_0^\infty d\eta\, e^{-\eta/\gamma}\int_0^\infty d\xi \frac{e^{-(\xi+\eta)^2/4\tau}}{\sqrt{\pi\tau}}\right] \tag{68}$$

or

$$F_0(\tau) = \frac{\gamma}{1+\gamma} - 2e^{1/\gamma+\tau/\gamma^2}\frac{2}{\sqrt{\pi}}\int_{1/2\sqrt{\tau}+\sqrt{\tau}/\gamma}^\infty e^{-u^2}du \tag{69}$$

and

$$F_1(\tau) = \frac{-1}{1+\gamma} + e^{-\tau/\gamma^2}\frac{2}{\sqrt{\pi}}\int_{\sqrt{\tau}/\gamma}^\infty e^{-u^2}du \tag{70}$$

Expressions (68) and (69) have been calculated numerically for a range of γ, and compared to the exact eigenvalue expressions (64) and (65), keeping up to 900 terms. At least five place agreement occurs for $\tau \leq 0.1$. For that value of τ, convergence of the eigenfunction series occurs with only 8 terms (all 900 terms are required for $t=0.01$). We can therefore define a time breakpoint τ_1 (=0.1) below which the early time approximation applies, and above which the eigenfunction expansion has to be used.

We now turn to describe specific algorithms for time and position selection for $\tau<\tau_1$ and $\tau>\tau_1$. In both cases three possibilities have to be covered: conduction to the $\xi=0$ boundary at time τ, conduction to the $\xi=1$ boundary at time τ, and diffusion to an internal point $0<\xi<1$ at time τ.

6.1 Case of Early Times

6.1.1 Conduction to the $\xi=0$ boundary

A time τ has to be sampled from $F_0(\tau)$ as given by Equation (68). Introducing the new variables

$$\begin{aligned} u &= \xi/2\sqrt{t} \\ v &= (\xi+\eta)/2\sqrt{t} \\ \omega &= 1/2\sqrt{t}, \end{aligned} \tag{71}$$

Equation (68) becomes

$$F(\omega) = \frac{\gamma}{1+\gamma} - \frac{4}{\sqrt{\pi}}\left[\int_{\omega}^{\infty} e^{-u^2}\,du - \frac{1}{\gamma}\int_0^{\infty} d\eta\;\; e^{-\eta/\gamma}\int_{(1+\eta)\omega}^{\infty} e^{-v^2}\,dv\right] \tag{72}$$

Multiplying the u-integral by $\frac{e^{1/\gamma}}{\gamma}\int_1^{\infty} e^{-\zeta/\gamma}\,d\zeta$, which is equal to unity, and changing the variable η to $\zeta-1$, Equation (72) becomes

$$F(\omega) = \frac{\gamma}{1+\gamma} - \frac{4}{\sqrt{\pi}}\frac{e^{1/\gamma}}{\gamma}\int_1^{\infty} e^{-\zeta/\gamma}d\zeta\;\left[\int_{\omega}^{\infty} e^{-u^2}du - \int_{\omega\zeta}^{\infty} e^{-v^2}dv\right] \tag{73}$$

The probability density function of is proportional to $f(\omega) = \frac{dF(\omega)}{d\omega}$:

$$f(\omega) = \frac{4}{\sqrt{\pi}}\frac{e^{1/\gamma}}{\gamma}\int_1^{\infty} e^{-\zeta/\gamma}\,d\zeta\;\; e^{-\omega^2}\left[1 - \zeta e^{-(\zeta^2-1)\omega^2}\right] \tag{74}$$

Expression (74) is positive for all ζ provided $\omega>1$. As $\omega>\omega_1$ and $\omega_1 = 1/2\sqrt{t_1} = 1/2\sqrt{.1} = 1.58$, the condition is satisfied.

The problem of selecting $\tau<\tau_1$ from Equation (72) is therefore reduced to selecting $\omega>\omega_1$ from Equation (74), where

$$\omega_1 = 1/2\sqrt{\tau_3} \qquad (75)$$

This can be done as follows:

1) select $\zeta>1$ from $e^{-\zeta/\gamma}d\zeta$

2) select $\hat{\omega}>\omega_1$ from $e^{-\omega^2}d\omega$

3) accept the sample with probability $1-\zeta e^{-(\zeta^2-1)\omega^2}$, and set $\tau=1/(2\omega)^2$. If rejection occurs, repeat steps 1,2,3.

6.1.2 Conduction to the $\xi= 1$ Boundary

A time τ has to be sampled from $F_1(\tau)$ as given by Equation (70). The selection for $0<\tau<\infty$ is rather simple:

$$\tau = \gamma^2\lambda^2/4v^2$$

where $p(\lambda) = e^{-\lambda}$ and $p(v) = \frac{2}{\sqrt{\pi}} e^{-v^2}$

However, what is needed is selection in a finite range $0<\tau<\tau_1$, where τ_1 can be rather small.

To develop a selection technique, let

$$x = \sqrt{\tau}/\gamma \quad , \quad x_1 = \sqrt{\tau_1}/\gamma$$

Equation (70) becomes

$$F(x) \propto e^{x^2} \frac{2}{\sqrt{\pi}} \int_x^\infty e^{-u^2}\, du$$

$$\propto \frac{2}{\sqrt{\pi}} \int_x^\infty e^{-(u-x)(u+x)} du$$

Let $u-x = v$

$$F(x) \propto \frac{2}{\sqrt{\pi}} \int_0^\infty e^{-v(v+2x)} dv$$

The probability density function of x is

$$p_x(x) = -\frac{dF}{dx} \propto \frac{2}{\sqrt{\pi}} \int_0^\infty 2v\ e^{-v(v+2x)}dv \qquad (76)$$

Let us define the pdf

$$p(x,v) \propto \frac{2}{\sqrt{\pi}}\ 2v\ e^{-v(v+2x)} \qquad (77)$$

$p_x(x)$ is the marginal distribution of (77).

The marginal distribution of v is:

$$p_v(v) = \int_0^{x_1} p(x,v)dx \propto \frac{2}{\sqrt{\pi}}\ e^{-v^2}\left[1 - e^{-2vx_1}\right] \qquad (78)$$

If v is selected from (78), the conditional distribution of x is ratio of (77) to (78):

$$p_x(x,v) \propto 2v\ e^{-2vx} \quad 0 \leq x \leq x_1 \qquad (79)$$

To select v from (78), the following techniques are efficient:

If x_1 is not small with respect to unity, select v from e^{-v^2}, and accept with probability $1-e^{-2v\ x_1}$.

If x_1 is small, select v from $2v\ e^{-v^2}$ ($v=\sqrt{\lambda}$, $p(\lambda)=e^{-\lambda}$), and accept with probability* $(1-e^{-2v\ x_1})/2vx_1$. The optimum breakpoint value of x_1 is $\sqrt{\pi}/2$. The overall efficiency of the rejection technique is 100% as $x_1 \to 0$ and as $x_1 \to \infty$; it reaches a minimum value of 54% at $x_1 = \sqrt{\pi}/2$.

Once v is selected, the selection of x from (79) is trivial: select λ from $e^{-\lambda}$, $0<\lambda<2vx_1$ and set $x = -\lambda/2v$.

Finally the time is given by

$$\tau = \gamma^2 x^2 = \gamma^2\lambda^2/4v^2.$$

*The probability is $>(1-vx_1)$. This fact can be used to avoid, with high probability, the necessity to compute an exponential.

6.1.3 Conduction to an Internal Point

The problem consists in selecting a position coordinate x from the pdf (61), at a given τ.

This equation can be written as:

$$G(\xi,\tau) = \int_0^\infty \frac{d\eta}{\gamma} e^{-\eta/\gamma} \left\{ e^{-\xi^2/4\tau} - e^{-(\xi+\eta)^2/4\tau} - e^{-(2-\xi)^2/4\tau} + e^{-(2-\xi+\eta)^2/4\tau} \right\} \tag{80}$$

To select ξ from (80) one can first select η from $\frac{1}{\gamma} e^{-\eta/\gamma}$, ξ from $e^{-\xi^2/4\tau}$ and accept the sample with probability

$$P = \left[1 - e^{-\left[(\xi+\eta)^2-\xi^2\right]/4\tau} - e^{-\left[(2-\xi)^2-\xi^2\right]/4\tau} + e^{-\left[(2-\xi+\eta)^2-\xi^2\right]/4\tau}\right]$$

The efficiency of the rejection technique deteriorates for small values of $g = \gamma/2\sqrt{\tau}$. In that case, another algorithm becomes efficient.

With probability $1/(1+g\sqrt{x})$, sample x from xe^{-x} and u from $2ue^{-u^2}$. With remaining probability sample x from $x^2/2\ e^{-x^2}$ and u from $2/\sqrt{\pi}\ e^{-x^2}$. In both cases set $\eta = 2\sqrt{\tau}\ gx$ and $\xi = 2\sqrt{\tau}\ u$, and accept the sample with probability

$p' = p \cdot 4\tau/(\ (2\xi+\eta))$.

For optimum operation the first sampling technique should be used for $g>g_0$, the second one for $g<g_0$. The optimum value of g_0 can be calculated for $\tau<<1$:

$$g_0 = \sqrt{\frac{1}{4\pi} + \frac{1}{2}} - \sqrt{\frac{1}{4\pi}} = 0.479$$

The efficiency of the mixed algorithm is 100% as $g\to 0$ and $g\to\infty$. It assumes a minimum value of 47% at $g = g_0$.

6.2 Case of Late Times

6.2.1 Conduction to Either Boundary

The time is sampled by sampling a random number r and solving the equation

$$rF_i(0) = F_i(\tau) \qquad i = 0 \text{ or } 1 \qquad (81)$$

i = 0 or 1 corresponds to boundary ξ = or $\xi = 1$, respectively.

For each sample (each value of γ), the eigenvalues of coefficients are calculated. The solution of (81) is obtained by performing a binary search: if $\tau_1<\tau<\tau_2$, where $F_i(\tau_j)$ is known for j = 1,2, the next approximation of τ is τ_3 obtained by linear interpolation between τ_1 and τ_2. The value $F_i(\tau_3)$ is calculated numerically, and τ_3 replaces either τ_1 or τ_2, thus reducing the interval $\tau_1-\tau_2$. The search is terminated if $F_i(\tau_1)-F_i(\tau_2)$ becomes small enough (5% of $F_i(0)$). If $\tau_2 = \infty$ (e.g., at the beginning of the binary search), the "linear interpolations" for τ_3 is replaced by selection from the first term of expansion (3) or (4):

$$\tau_3 = \tau_1 - \frac{1}{\alpha_1^2} \log (r \cdot F_i(\tau_1))$$

6.2.2 Conduction to an Internal Point

At late time τ, the pdf of ξ $0<\xi<1$ is proportional to:

$$G(\xi,\tau) = \sum_n \frac{2\sin\alpha_n}{1+\gamma\cos\alpha_n} \sin(\alpha_n \xi) e^{-\alpha_n^2 \tau}$$

The cumulative pdf of ξ is therefore

$$P(\xi) = \int_0^\xi G(\xi,\tau)d\xi = \sum_n \frac{2\sin\alpha_n}{\alpha_n(1+\gamma\cos\alpha_n)} (1-\cos\alpha_n \xi)\, e^{-\alpha_n^2 \tau} \qquad (82)$$

As in the case of time distribution, the selection is performed by solving

$$r \cdot P(1) = P(\xi) \qquad (83)$$

using a binary search technique.

7. Importance Sampling of RPP Green's Functions with Homogeneous Boundary Conditions

Biased sampling of the Green's functions becomes necessary if boundary conditions of the type

$$T - T_g = - \beta \frac{\partial T}{\partial n}$$

occurs with large values of both T_g and β.

The natural probability of scoring T_g is small, but T_g is much larger than any local temperature T. This leads to a high variance of the estimates. The situation can be corrected by proper biasing if reasonable estimates $\tilde{T}_g$ of T_g and $\tilde{T}$ of the local temperature T can be made.

Let us assume that the RPP is oriented along the $x_1x_2x_3$ axis, that zero boundary conditions are satisfied in the x_1-x_3 and x_2-x_3 planes as well as on one of the x_1-x_2 planes. The condition $G = - \gamma\frac{\partial G}{\partial n}$ is satisfied on the other x_1-x_2 plane. The procedure outlined in Section 3 is altered: the time t_1 of diffusion to the x_1-x_3 planes and to the x_2-x_3 planes are sampled first. The smallest of t_0, t_1, t_2 is determined and set as t_m, where t_0 is the time cutoff. t_m is scaled to the frame of x_3: $\tau_m = Dt_m/a_3^2$. If $F_0(\tau)$ is the function defined by Equation (64) or (69) of Section 6, then, with probability

$$p_g = F_0(0) - F_0(\tau_m)$$

the score will be T_g. With remaining probability (1-p) it will be a local temperature: with probability

$$p_a = F_1(0) - F_1(\tau_m)$$

the score will be a temperature T_a at a point on the opposite side of the x_3 interval, and, with probability

$$p_i = 1 - F_0(\tau_m) - F_1(\tau_m) \qquad (p_g + p_a + p_i = 1)$$

the score will be a temperature T_i at a point internal to the interval of x_3.

Assuming that T_g is reasonably close to $\tilde{T}_g$, and that both T_a and T_i are reasonably close to $\tilde{T}$, reasonable biased probabilities and associated weight factors can be defined as:

$$\tilde{p}_g = p_g\tilde{T}_g/N \qquad W_g = N/\tilde{T}_g$$

$$\tilde{p}_a = p_a\tilde{T}/N \qquad W_a = N/\tilde{T}$$

$$\tilde{p}_i = p_i\tilde{T}/N \qquad W_i = N/\tilde{T}$$

where $N = p_g\tilde{T}_g + (1-p_g)\tilde{T}$.

The associated scores will be W_gT_g, W_aT_a, W_iT_i, which are all approximately equal to N.

IV. GREEN'S FUNCTIONS IN INHOMOGENEOUS RECTANGULAR PARALLELEPIPEDS

The Green's functions defined for homogeneous RPP in Section III can be used provided the RPP is fully contained in a homogeneous region. Another type of Green's functions has to be introduced when a point x_0 is placed on the interface between two dissimilar materials. As in the previous section, we restrict the shape of the volume V to a rectangular parallelepiped. The inhomogeneity is restricted to two media with a planar interface, the interface being parallel to a side of the RPP.

To be more specific, the volume V is defined by:

$$\alpha_i^- \leq x_i \leq \alpha_i^+ \qquad i = 1,3 \tag{1}$$

The diffusion parameters are:

$$\left.\begin{aligned} K &= K_1 \\ \rho_c &= \rho_{c_1} \end{aligned}\right\} \text{ for } x_3>0$$

$$\left.\begin{aligned} K &= K_2 \\ \rho_c &= \rho_{c_2} \end{aligned}\right\} \text{ for } x_3<0 \tag{2}$$

Equations (5)-(8) of Section I.2 can be rewritten as:

$$\frac{K}{\rho c}\left(\frac{\partial^2 G}{\partial x_1^2} + \frac{\partial^2 G}{\partial x_2^2} + \frac{\partial^2 G}{\partial x_3^2}\right) - \frac{\partial G}{\partial t} = 0 \tag{3}$$

with the boundary and initial conditions

$$G(x_1,x_2,x_3,0) = \delta(x_1)\delta(x_2)\delta(x_3) \tag{4}$$

$$G(x_1,x_2,x_3,t) = 0 \ , \ x_i = \alpha_i^{\pm} \ , \quad i = 1,3 \tag{5}$$

$$\frac{1}{\rho c_1} G(x_1,x_2,0^+,t) = \frac{1}{\rho c_2} G(x_1,x_2,0^-,t) \tag{6}$$

$$\frac{K_1}{\rho c_1}\frac{\partial}{\partial x_3} G(x_1,x_2,0^+,t) = \frac{K_2}{\rho c_2} G(x_1,x_2,0^-,t) \tag{7}$$

Unfortunately, separation of variables does not apply to this problem. Partial separation can be achieved in cylindrical symmetry (if the RPP is replaced by a cylinder around the x_3-axis). The remaining problem is three-dimensional (r,z,t-dependent), and an eigenfunction expansion can be written down. Attempts to implement an approach based on this eigenfunction expansion have been given up. The solution of the eigenvalue problem is too time consuming to be carried out during the course of a Monte Carlo calculation. Precalculation of sampling tables for a representative set of parameters $(K_1, K_2, \rho c_1, \rho c_2, \alpha_i^{\pm})$ appear to be impractical as the tables would be too bulky to keep in computer core.

An approximation (which can be reduced to any degree) has been introduced. It is well known that the diffusion equation can be considered as an approximation of the transport equation in the limit of small mean free paths. Conversely, the diffusion equation can be approximated by a transport equation with small mean free paths.

Consider the transport of radiation in a homogeneous medium with the following properties: The root mean square free path of a particle is ε, the velocity is v, scattering is isotropic. For small enough ε, the diffusion approximation applies, with a diffusivity*

$$D = K/\rho c = v\varepsilon/6. \qquad (8)$$

The distribution of free paths is irrelevant in the limit of small ε. We are free to take constant free paths of length ε.

At the interface between two different materials, the scattering is anisotropic: with probability p_1, it is isotropic into medium 1, and with probability $1-p_1$, it is isotropic into medium 2. In order to satisfy the boundary conditions (6),(7)

$$p_1 = \sqrt{K_1 \cdot \rho c_1} \;/\; (\sqrt{K_1 \cdot \rho c_1} + \sqrt{K_2 \cdot \rho c_2}) \qquad (9)$$

* In neutron transport, the root mean square free path $\varepsilon = 2\lambda$, where λ is the mean free path, giving the familiar $D = v\lambda/3$.

The accuracy of the method is determined by the smallness of ε. Once ε_i is chosen for medium i, i=1,2, the velocity v_i is determined from Equation (8). An adequate accuracy is achieved by setting ε_i to the smallest of ε_i^t, ε_i^ℓ, where

$$\varepsilon_i^{\,t} = \sqrt{6\ D_i t_0/25}$$

$$\varepsilon_i^{\,\ell} = L_i/6 \text{ where } L_i \text{ is the smallest of}$$

$$\pm\alpha_1^{\pm},\ \pm\alpha_2^{\pm},\ \alpha_3^{+} \text{ if } i=1,\ -\alpha_3^{-} \text{ if } 1=2.$$

Setting $\varepsilon_i = \varepsilon_i^{\,t}$ will insure 25 collisions before reaching a given time cutoff t_0. Setting $\varepsilon_i = \varepsilon_i^{\,\ell}$ will insure a number of collisions of the order of 25 before leaking out of the RPP.

Once the particle diffuses away from the interface, one can switch to the floating homogeneous RPP method. Optimum computing times are achieved if this switch occurs when the particle diffuses a distance of $\sim 6\varepsilon_i^{\,\ell}$ from the interface.

V. HEAT TRANSFER GEOMETRY PACKAGE

1. Geometrical Description

The geometrical description is of the combinatorial type[4]. Regions are described in terms of intersection of bodies. Bodies are described in terms of intersection of quadratic surfaces. The only surfaces currently being implemented consist of planes, spheres, cylinders, and circular cones (corresponding to combinatorial bodies RPP, BOX, SPH, RCC, TRC, WED, ARB, but excluding REC and ELL).

If a configuration is axially repetitive, only a single repetitive element needs to be described, as explained in Section VII.9. The geometrical package is completely general. However, if the configuration exhibits at least partial cyclindrical symmetry, more efficient heat transfer calculations will be performed if the axis is in the z-direction (only subroutine SOUSET is affected. See Section VII.5).

2. Geometrical Input

The input consists of

1) Title Card

2) Surface Data

3) Body definition in terms of above surfaces

4) Region definition in terms of above bodies

It is envisioned to develop a preprocessor which will accept as input combinatorial bodies.

2.1 Title Card (I4,19A4)

IPR   If zero - input and processed data will be printed back.

HOLL   Any hollerith information serving as a title.

2.2 Surface Input (2X, A3, 8E9.3)

The input number ε required for each surface can be regarded as a surface thickness. Its meaning is described in Section V.4.1.

Plane: PLN ε V_1 V_2 V_3 H_1 H_2 H_3

where V is a point on the plane, and H is orthogonal to the plane, pointing "inside" the plane (arbitrary normalization)

Sphere: SPH ε V_1 V_2 V_3 R

where V is the center and R the radius

Cylinder: CYL ε V_1 V_2 V_3 H_1 H_2 H_3 R

where V is a point on the axis, H points along the axis (arbitrary normalization) and R is the radius

Cone: CON ε V_1 V_2 V_3 H_1 H_2 H_3 T

where V is the vertex, H points along the axis (arbitrary normalization), and T is the tangent of the angle of aperture

End of surface input marker: END

2.3 <u>Body Input</u> (2X, A3, 15I5)

GEN N_1 N_2 N_3 ...

where $|N_i|$ is surface number $N_i>0$ implies that the body is inside the surface, $N_i<0$ - outside. The first $N_i = 0$ implies the end of list at i-1. If none, end occurs at i = 16.

End of body input marker: END

2.4 <u>Region Input</u> (2X, A3, 15I5)

ANY N_1 N_2 N_3 ...

where ANY is any three characters ($\neq$ END) and $|N_i|$ = body number. $N_i>0$ implies the region is inside body, $N_i<0$ - outside. $N_i = 0$ end of list marker.

End of region input marker: END

3. The Input Processing Routines

GENI, GENIP, GENICK

These subroutines read card input and store processed data into a single dimensional array called AST in floating point mode and MAS in integer mode.

The layout of the information is shown in Table 1.

The surface data is processed first, and stored sequentially in the first available location IS of the array. One word is left blank for future use. The TYPE is changed to a numerical code as shown in Table 1. The vector H is normalized to unity. An exhaustive search of surface duplicates is performed. Each time a surface is read in, the previous list is examined for an identical surface. If a duplicate is found, the new surface is ignored, and the parameter ε of the old surface is changed to the smallest of the old and new ε. The address JS constitute the only identifier of the surface. A temporary dictionary ISLOC(IS)=JS is generated, which defines the surface identifier JS corresponding to the ordinal surface number IS. In the case of a new plane identical to an old plane except for the vector H pointing in the opposite direction, the old H is left unchanged, but the dictionary entry of the new plane is tagged with a negative sign: ISLOC(IS)=-JS.

The body definition is read in next. The information is stored in terms of surface identifiers as determined from the surface dictionary ISLOC. A temporary body dictionary IBLOC is generated, which defines the body identifier in terms of body input number.

The last block of information read in is the region definition. The information is stored in terms of body identifiers as determined from the dictionary IBLOC. A permanent region dictionary ISLOC is generated, which defines the region body list identifers in terms of the ordinal region numbers. (The original surface dictionary ISLOC is destroyed.)

TABLE 1

Layout of the MAS - AST Array

LOCATION	CONTENT					
1	BLANK					surface "1"
2	ITYPE					
3	V_1					
---	---					
---	---					
---	---					
JS	BLANK*					
JS+1	ITYPE	1 = SPH	2 = CYL	3 = CON	4 = PLN	SURFACE "JS"
JS+2	V_1	R	H_1	H_1	H_1	
JS+3	V_2		H_2	H_2	H_2	
JS+4	V_3		H_3	H_3	H_3	
---				$\sin\alpha$		
---				$-\cos\alpha$		

---	---					
IB	BLANK					Body "IB"
IB+1	$\pm JS_1$					
IB+2	$\pm JS_2$					
---	---					
IB+ℓ-1	$\pm JS_\ell$					
IB+ℓ	0					

TABLE 1 (continued)

LOCATION	CONTENT	
IRB = ISLOC(IR)	BLANK	
IRB+1	$\pm IB_1$	
IRB+2	$\pm IB_2$	region IR
---	---	body list
IRB+m-1	$\pm IB_m$	
IRB+m	0	

IRS = IBLOC(IR)	JS_1	
IRS+1	BLANK	
IRS+2	BLANK	
IRS+3	BLANK	
IRS+4	JS_2	
IRS+5	BLANK	
IRS+6	BLANK	
IRS+7	BLANK	region IR
	---	surface list

ISM = IRS+4(n-1)	JS_n	
ISM+1	BLANK	
ISM+2	BLANK	
ISM+3	BLANK	
IRS+4n	0	

*All blank locations are zero'd in at input time

The final task of the preprocessing routines is to build, for each region, the list of surfaces involved in defining the region boundary. The list is in terms of surface identifiers, and is stored in the MAS array. A permanent region dictionary IBLOC is generated, which defines the region surface list identifiers in terms of the ordinal region numbers. (The original body dictionary IBLOC is destroyed.)

Throughout the input processing, a rather thorough search of errors and inconsistencies is performed. Error or warning messages are issued. If possible, fatal errors are temporarily fixed up to allow continuation of scanning. If any fatal errors occurred, a stop is executed at the end of input processing.

4. The Geometry Subroutines

4.1 The Box Fitting Subroutine GBOX

The principal purpose of this subroutine is, given a point in a region, to construct a box centered around the point, and fully contained within the region. The secondary purpose is, given a point on the boundary of a region, to construct a box fully contained within the region, with one face centered around the point. In both cases the box has to be "as large as conveniently possible". The ill-defined statement in quotation marks can be qualified as follows. The efficiency of the diffusion code requires on one hand that the smallest dimension of the box be as large as possible, and, on the other hand, that the largest fraction of the surface area of the box coincides with the region boundary. The efficiency of the box construction routine, however, requires avoiding lengthy calculations and tests. The subroutine GBOX implements a reasonable compromise between these conflicting requirements.

The requirement that a fraction of the surface area of the box coincides, at least occasionally, with the region boundary is a necessary one for termination of a Monte Carlo history. This can be achieved exactly if and only if the region boundary consists of planar facets. To avoid this restriction, we consider the curved surfaces as slightly diffuse. A point within a distance ε from the surface is considered to be on the surface, where ε is the input number defined in Section 2.2, which should depend on the diffusion properties of the materials on either sides of the surfaces.

The construction of a box abutting a single surface is shown in Figure 1 and 2 for convex and concave surfaces. The distance D_1 is the distance of closest approach, plus or minus ε depending on the convexity of the surface. D_2 is the length of the diagonal of the largest box, one face of which can be considered as being entirely "on" the surface. R is the length of the diagonal of the abutting rectangle. If R_s is the small radius of curvature of the surface, $R^2 = 4R_s\varepsilon - \varepsilon^2$, or $R \approx 2\sqrt{R_s\varepsilon}$. ($R_s = R$ for a sphere or a cylinder, $R_s = \infty$ for a plane, and R_s is determined as a function of position for the cone).

The subroutine examines each surface of the list IRS. It calculates the distance of closest approach, and, if this turns out to be the currently shortest one, calculates also R, D_2, as well as $\vec{Q}$ where $\vec{Q}$ is the vector from the point of closest approach to the given point X. The distance of closest approach to the next nearest boundary is stored as D_3.

When the list of surfaces has been exhausted, a box is tentatively constructed. The diagonal of the box is D_3', the shortest of D_2 and D_3. The dimension of the box along $\vec{Q}$ is $2D_1$. The other two dimensions are equal to $2D_2'$, with $D_2' = \sqrt{(D_3'^2 - D,^2)/2}$. The address ISM (see Table 1) for the closest surface is saved.

The box just generated is perfectly valid. For efficiency purpose, two tests are performed to investigate the possibility of a "better" box:

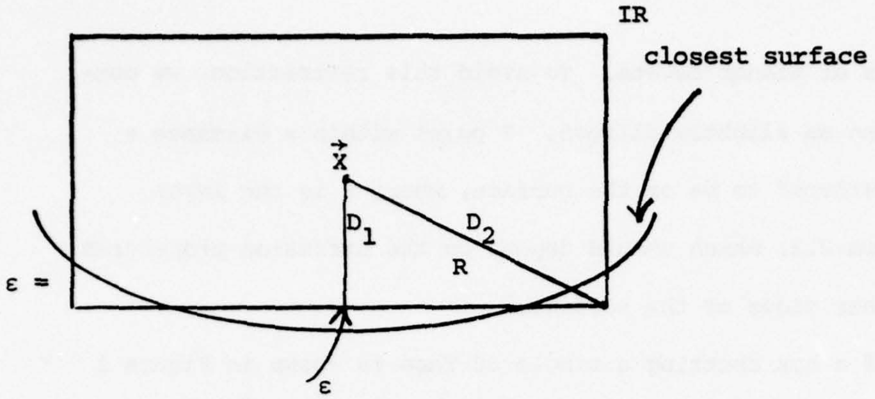


Figure 1 - concave surface

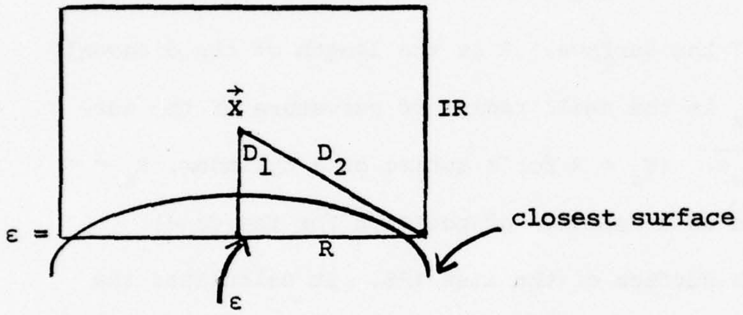


Figure 2 - convex surface

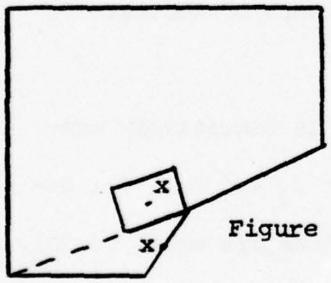


Figure 3 - tentative box abuts boundary extension, not to real boundary.

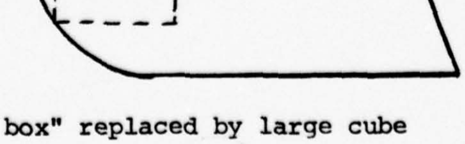


Figure 4 - "long box" replaced by large cube

1) The box does abut a surface which is part of the definition of the region boundary. But does it abut that part of the surface which really defines the boundary? The situation is described in Figure 3. The point of closest approach X_0 is tested using subroutine REGCK described in Section 4.2.2. If the point X_0 turns out not to be on the region boundary, the box is changed to a (larger) cube with diagonal D_3: $D_1=D_2'=D_3/\sqrt{3}$. The vector Q is changed (arbitrarily) to 0,0,1. ISM is set to 0. However, if X_0 is on the real boundary, the following test is performed:

2) Is the abutting side of the box much further than the non-abutting sides of the box, which we implement as $D_1>2D_2$? If yes, diffusion to the abutting side is rather unlikely, and a cube of diagonal D_3' is more efficient. ($D_1=D_2'=D_3'/\sqrt{3}$,Q=0,0,1,ISM=0). The situation is sketched in Figure 4.

The above discussion describes the construction of a box centered around an internal point X. To obtain that case, the variable JSM in COMMON/GEOM/ has to be ≤ 0.

For the case of X "on" (in the diffuse sense) a boundary, the variable JSM has to be set to the identifier of the particular surface. The box construction proceeds in much the same way as previously described, except that abutting is forced to surface JSM. The tests to replace the box by a cube are bypassed. The vector Q is ill-defined. Another vector, QIN is generated, which is the inside normal of surface JSM at X.

4.2 The Point Checking Routines

The package consists of subroutine REGCK, SURFCK.

The variables of interest are found in COMMON/GEOM/ and consist of

X(3)	point being checked
IR	region number
IRP	neighbor region number
JSM	identifier of surface "on" which the point is assumed to be
LOOP	the Guber number

The Guber number should be familiar to all combinatorial geometricians. It is initialized to zero by subroutine GENI. It should be incremented by unity each time X changes between calls to any point checking routines. Its function will become apparent in the following subsections.

4.2.1 The Surface Checking Routine SURFCK(JS,ILOOP)

JS is the surface identifier. ILOOP should be numerically equal to LOOP. (Do not use LOOP itself as an argument!) The subroutine performs simple algebraic tests and determines whether X is "inside" or "outside" the surface.

The output ILOOP=LOOP if X is inside JS, ILOOP=-LOOP if X is outside JS. The output ILOOP is also stored in MAS(JS) (labeled as 'blank' in Table 1).

4.2.2 The Region Checking Routine REGCK(JRB,JLOOP)

The purpose of this routine is to check if a point X is in a region IR (JRB=ISLOC(IR)). JLOOP should be numerically equal to LOOP. The subroutine handles differently the case of a point inside a region or "on" (in the diffuse sense) a boundary.

4.2.2.1 Testing a Point Inside a Region

The "inside" case will be discussed first. If the point is not intentionally "on" a boundary, one should set JSM≤0.

The output of REGCK is JLOOP=LOOP if X is inside IRB, JLOOP=-LOOP if X is outside IRB. The output JLOOP is also stored in MAS(IRB).

The subroutine functions as follows:

It accesses the region definition in terms of bodies, and examines each body IB in turn.

It first examines MAS(IB) and compares its absolute value with LOOP. If they match, the body has been tested before for the same point X and the following set of tests can be bypassed.

If |MAS(IB)| does not match LOOP, the body definition in terms of surfaces is accessed, and each surface JS is examined in turn.

If |MAS(JS)| = LOOP the surface has been examined before. If not, a call to SURFCK is performed. In both cases the sign of MAS(JS) indicates whether X is inside or outside JS. The examination continues as long as the result of the tests match the body definition. The first mismatch indicates that X is outside the body; KLOOP is set equal to -LOOP. No mismatch until the end of information (JS=0) marker is reached indicates that the point is inside the body; KLOOP is set equal to +LOOP. In both cases KLOOP is stored into MAS(IB).

Whether determined previously or by the tests described above, the sign of MAS(IB) indicates whether X is inside or outside IB. The examination continues as long as the results of the tests match those of the region definition. The first mismatch indicates that X is outside the region; JLOOP is set equal to -LOOP. No mismatch until the end of information (IB=0) marker is reached indicates that the point is inside the region; JLOOP is set equal to +LOOP. In both cases JLOOP is stored in MAS(IRB).

4.2.3 Testing a Point "On" a Surface

In addition to X and LOOP, the following variables have to be set:

JSM	identifier of surface "on" which X lies
IR	known region number on one side of the surface
IRP	neighbor candidate being tested on the other side of the surface
IRB	(argument) IRB=MAS(IRP)

The body checking proceeds as usual unless surface JS=JSM is encountered. If it is, the testing of that particular surface is bypassed, and the remaining surfaces are considered. If the final result of the body test indicates that the point is outside the body, either assumption of X inside or outside JS=JSM will not change that result and the logic of the region checking is unaffected.

If, however, the result indicates that the point is inside the body, the point X is actually on the boundary of the body. If the region IRP being checked is equal to IR, the result can only be false, and is therefore set as such. If, however, IRP ≠ IR, the body test is tentatively assumed to match the region definition of IRP, the code proceeds to check the next body, and the logic proceeds along the lines of Section 4.2.2.1. The tentative assumption involves a definite assumption on the position of IRP with respect to surface JSM. That assumption is recorded as a signed integer K. In the event that, for another body, the opposite assumption has to be made the neighbor IRP is declared as invalid.

For a successful IRP, the subroutine also generates a vector QB which points from IR to IRP. This is done by setting QB=±QIN, where QIN is a vector (generated by GBOX, see Section 4.1) pointing into surface JSM, and the sign is determined by the sign of K.

4.3 The Neighbor Finding Subroutine NEBFND

The purpose of this subroutine is to find the neighbor IRP of a region IR at a point X "on" a particular surface. The surface is defined by the variable ISM (set by subroutine GBOX). Referring to Table 1, the n-th surface of the IRS list is defined by ISM=IRS+4(n-1).

For efficiency purposes, the subroutine builds up a list of neighbors previously determined, for each region across each surface. The list is stored in the MAS array. The first entry (if any) is stored in MAS(ISM+1). The second one (if any) is stored in MAS(ISM+2). If a third entry becomes necessary, the first available location LS in the MAS array is determined, and the entry is stored there; the address of LS of that third entry is stored in MAS(ISM+3). LS is set to LS+3 to reserve a location for a possible fourth entry, and for the location of a possible fifth entry, etc. The buildup of tables stops if the MAS array has been filled.

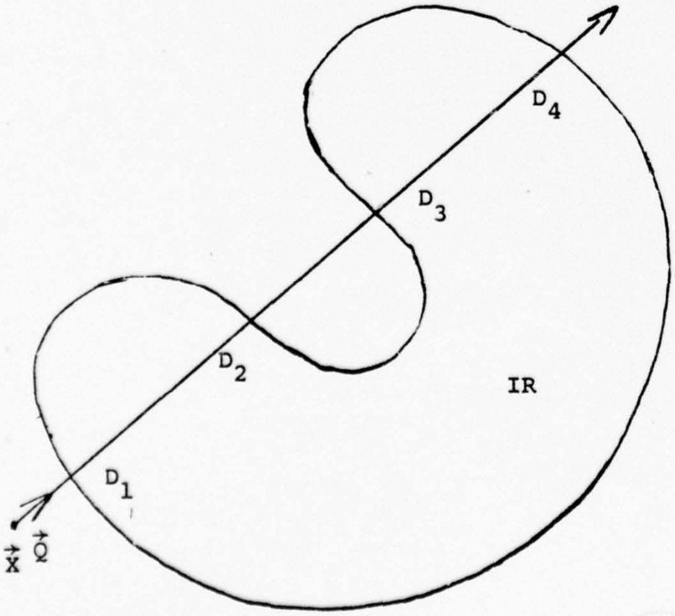
The detailed operation of the subroutine is as follows: it examines the first candidate (IRP=MAS(ISM+1)). If IRP=0, no candidate exists and the following step is bypassed. If IRP>0, REGCK is called for checking IRP. If IRP is found valid, the subroutine returns to the calling program. If not, the next candidate is examined, until either a validity check or IRP=0 is encountered. In the latter case, a loop over all IRPs is initiated. The first check performed is whether the region has been tested before (IRB=ISLOC(JRP), |MAS(IRB)| = LOOP?). If yes, the next IRP is considered. If IRP has not been tested before, the next test consists in determining whether the boundary of IRP includes the particular surface. If not, the next IRP is considered. Otherwise, IRP is checked further by calling REGCK. If the region is found invalid, the next IRP is considered. The first valid IRP terminates the loop. It is stored as a next candidate. Before returning to the calling program, the fact that IR is a neighbor of IRP across the same surface is also recorded.

4.4 The Region Finding Subroutine REGFND

The subroutine determines the region number IR for a point X. Error messages are printed if the region is undefined or multiply defined. Subroutine REGCK is called for all regions.

4.5 The Tracking Routines G1, G1P

Given a point $\vec{X}$, a unit vector $\vec{Q}$, and a region IR, the subroutine calculates the number NI of intersections, and the intersections DI(I), I=1, NI of region IR and of the line passing through $\vec{X}$ in the direction $\vec{Q}$.



The operation of the subroutine is as follows:

First, all the intersections with all the surfaces which are involved in the definition of IR are calculated. These include all intersections with IR, and may include some intersections which are not on the surface of IR. Subroutine G1P first orders all the intersections, then weeds out the extraneous ones. The point X is moved to below the first intersection (thus to a point definitely outside IR). Subroutine REGCK is called, with a special switch JSM=-33333 which forces REGCK to check all surfaces without returning at the first mismatch encountered. A loop is initiated, which runs through all the tentative intersections. For each intersection, the logical information of whether X is in or out of the corresponding surface is negated. Subroutine REGCK, which queries that information, and its logical output (in or out of IR) is compared to its previous logical output. If the two match, the corresponding intersection is not on the surface of IR, and is weeded out. If the two do not match, a true intersection has been encountered and is therefore retained.

VI. THE MAJOR MONTE CARLO HEAT TRANSFER SUBROUTINES

The main program of the heat transfer code is a rather complex driver calling input subroutines, source generating routines, and, under their controls, the main heat transfer routine GETEMP. Finally, edit and output routines are called. The description of that operation will be given in Section VII.

The present section is devoted to the description of the main Monte Carlo routine GETEMP, which delivers a single estimate of the temperature at a given point $\vec{X}$ in region IR, implementing the algorithm described in Section II.3, and to the descriptions of the main subroutines called by GETEMP, which perform the calculations described in Section III and IV.

1. Subroutine GETEMP

The subroutine GETEMP is the main Monte Carlo routine which follows the histories along the lines described in Section II.3. It delivers a single estimate TEMP of the temperature at a point X at time TB in region IR.

We first describe its operation under normal circumstances, which is invoked when all DFAST(IR) are set equal to zero. The weight W is set to 1. Subroutine GBOX (described in Section IV.4.1) is called. A box is therefore constructed which is wholly in region IR, and which is centered at $\vec{X}$. A call to subroutine DIFFUS (described below) samples the RPF Green's function with zero boundary conditions along the lines of Section III.2. A subsequent call to subroutine MOVE (described below) moves the point X to the sampled location and updates the time variable. If the time variable is zero, the temperature at t=0 in IR is scored and a return is executed. If t>0, a test is made whether the new point is on an abutting boundary. If it is not, a new box is constructed around the new point. If this construction indicates that the point was within a distance ϵ (defined in Section IV) of the abutting surface, the point is treated as being on the abutting surface. If not on the abutting surface, the sequence of calls to DIFFUS and MOVE continues until either the time variable reaches zero, or the point has diffused to a region boundary.

If the point is on a boundary, the neighbor region number IRP is determined. The diffusion coefficient DP of the neighbor region is looked up. The convention is that if DP=0, the neighboring region is one with known temperature. If DP<0 then inhomogeneous boundary conditions apply with $\beta=-DP/K$ (β is defined in Section II.2). If DP>0, the neighboring region is a conductive one.

If DP=0, a score is made, and a return is executed. If DP≠0, a "half box" is constructed in region IR: this is a box one side of which is on the surface of interest, centered around X. The box is fully contained in IR. A "half box" is also constructed in IRP.

If DP>0, the subroutine MENDEL is called. Subroutine MENDEL is described below; it samples the two region RPP Green's function along the lines discussed in Section V. A call to subroutine MOVE moves the point X and updates the time t. If t=0, a score is made and a return is executed. If not, a new diffusion loop is started as described above.

If DP<0, the subroutine NORMAN (described below) is called. It samples the RPP Green's function with inhomogenous boundary conditions along the lines of Section III.5 and updates the weight value.

Prior to the call to NORMAN, γ is set, as well as are the estimates $\tilde{T}_g$ and $\tilde{T}$ needed for importance sampling. These are taken to be the value of the gas temperature in IRP and the value of the temperature in IR at t=0. If the particle diffuses to the abutting boundary, a score is made. The same is done if t becomes zero. In both cases, a return is executed. If neither occur, a new diffusion loop is initiated.

The number of diffusion steps within a region is limited to be less or equal to NRSTEP. The number of region crossings is limited by NRCR. If any of these is reached, a return with TEMP=-1. is executed. The same error return occurs if any of the three dimensions of any box becomes smaller than DTINY, presently set to 10^{-10}. Experience in running the code should teach how to avoid such collapsing boxes - a rarely occurring event anyway. NRSTEP, NRCR are currently set to 200 and 5000, respectively.

The above description is applicable if the input parameters DFAST(IR) is set to zero for all regions IR. It turns out that, in solving heat transfer problems with expected high temperature gradients, homogeneous regions have to be subdivided into rather small subregions, to satisfy the requirement that the diffusion properties be constant in each subregion. With such a detailed geometrical subdivision, the calculations involved become excessively slow, because each boundary crossing involves two calls to subroutine GBOX and a call to the boundary crossing subroutine MENDEL. The accuracy of the calculations involved in such a detailed treatment for crossing imaginary boundaries cannot be justified, and an approximation to speed up the calculation is in order. Selected regions can be declared on input as not requiring completely accurate boundary treatment. This is specified by setting the input quantity DFAST(IR) for selected regions IR. The approximation consists in assuming that the diffusion parameters are constant within a (small) distance $|DFAST(IR)|$ from any point in region IR, including points on the boundary of IR.

Before the call to GBOX, subroutine GETEMP examines the value of DFAST(IR). If zero, the calculations proceed as described above. If the value of DFAST(IR) is positive, the call to GBOX is performed and the smallest dimension of the RPP is compared to the value of DFAST(IR). If smaller, the calculations are performed as described above. If larger, a cube of dimension DFAST(IR) is set up and the calls to DIFFUS, MOVE, are replaced by a call to subroutine CUBE. If the value of DFAST(IR) is negative, the call to GBOX is bypassed altogether, a cube of dimension |DFAST(IR)| is set up, and the calls to DIFFUS, MOVE, are replaced by a call to subroutine CUBE. Subroutine CUBE (described below) moves the point $\vec{X}$ to its new position, updates t, and, if necessary, changes the region number IR.

2. Subroutine DIFFUS

Subroutine DIFFUS samples homogeneous RPP Green's functions with zero boundary conditions, as discussed in Section III.2.

The input consists of a time cutoff value T, and of $FOX(I)=a_i^2/D$, where a_i, i=1,3 are the dimensions of the RPP, and D is the diffusion coefficient. It is assumed that $a_1 = a_2 = 2*D1$ and $a_3 = 2*D2$.

The output consists of a time variable $TMIN \leq T$ and of a local vector $XX(I) \leq a_i/2$.

The time selection is achieved by three calls to subroutine PICKT. The position selection is achieved by calls to subroutine PICKX.

3. Subroutine MOVE

This subroutine displaces the absolute position $\vec{X}$ by the displacement $\vec{XX}$ generated by subroutine DIFFUS. The local vector $\vec{XX}$ is given in a coordinate system in which XX(3) is along the absolute vector $\vec{Q}$; the remaining two coordinates are orthonormal but otherwise arbitrary.

The time T is set to T-TMIN.

4. Subroutine CUBE

Subroutine CUBE is a speeded up version of the combination of DIFFUS and MOVE, under the simplifying conditions that

FOXX=FOX(1)=FOX(2)=FOX(3)

and that the vector $\vec{Q}$ points along the absolute z-axis, i.e., that the local coordinate system XX is parallel to the absolute system X.

5. Subroutine NORMAN

Subroutine NORMAN is the equivalent of DIFFUS, except that homogeneous boundary conditions are imposed at XX(3)=0. Importance sampling is implemented and the calculations follow the lines described in Section III.7. The selection of times t_1 and t_2 is achieved by calls to subroutine PICKT. The selection of t_3 is via a call to a subroutine PICKTG. Similarly, selection of XX(1) and XX(2) is via calls to PICKX, whereas the selection of XX(3) is via a call to a subroutine PICKXG.

6. Subroutine MENDEL

This is the boundary crossing subroutine. First the diffusion coefficients and the heat conductivities in the two regions are compared. If both comparisons show agreement within 5%, an average diffusion coefficient is calculated, and a call to DIFFUS is followed by a return.

If the homogeneity test fails, the subroutine switches to the method described in Section IV. The part of the calculation involving the floating RPP method is performed in subroutine DIMOR.

7. Subroutine DIMOR

Constructs an RPP within an RPP, and samples time and position via calls to PICKT and PICKX.

8. Auxiliary Subroutines

Special subroutines perform sampling of one dimensional Green's functions, as described in Section III. These include:

PICKT, PICKX - See Section III.4

PICKTG, PICKXG - See Section III.6.2.

When appropriate, the early time approximation is invoked by calls to subroutines EARLTG, EARLXG. (See Section III.6.1.) When appropriate, the eigenvalues are calculated by a call to subroutine TRANS. The function X=GAUSS(I) returns X>0 sampled from $2/\sqrt{\Pi} \exp(-X^2)dx$. The function y=ECHERF(X) returns $y=2/\sqrt{\pi} \exp(X^2) \int_0^X \exp(-y^2)dy$.

VII. THE THREE DIMENSIONAL TIME DEPENDENT ADJOINT MONTE CARLO HEAT TRANSFER CODE

The simplest application of the code is to a problem where the diffusion properties of the medium are independent of the local temperature. In that case, the operations of the code are reduced to reading the geometrical descriptions, the initial conditions, the boundary conditions, the coordinates of detector points, the value of the time variable at which the temperature needs to be calculated, and the number of Monte Carlo histories to be followed. The code then performs that many calls to subroutine GETEMP, calculates the effected temperature and its variance, and prints the results.

A more complex problem arises in the case where the diffusion properties do depend on the local temperature. The time span between initial and final time must then be subdivided into a number of time bins. The time bins must be small enough so that the local diffusion properties can be considered as constant throughout the time bins. Homogeneous regions must be subdivided into subregions. Each subregion must be small enough so that the local temperature at any point in the subregion does not differ appreciably from the average temperature in the region.

Given such a subdivision of space and time, the operation of the code is as follows. The initial conditions are read for the very first time bin only. The boundary conditions are also read in. The diffusion properties are calculated for each region at the value of the initial temperatures. These are assumed to remain constant throughout the time bin. The code then estimates temperatures at the end of the time bin for all regions. These calculated temperatures serve as initial conditions for the next time bin.

The calculation of region temperatures is performed in the following order. The code searches for the region with the highest known temperature at the end of the time bin. (This is restricted to external regions only at the beginning of the calculation.) Its neighbors are examined. The first neighbor found for which the temperature has not been calculated yet is then picked up, and the temperature is estimated by adjoint Monte Carlo. When all the neighbors have been completed, the region is excluded from the list, and the next region with the highest known temperature is searched. The calculations continue as long as such highest temperature is appreciably higher than T_m, where T_m is set slightly higher than the lowest temperature T_z of the configuration at the beginning of the time bin. All remaining regions are surrounded by regions with temperature less than T_m; their temperature is set equal to T_z.

The calculation of temperatures averaged over regions is performed by calculating the temperature at points distributed uniformly over the region; the sampling of such points is described in Section VII.5 below. The efficiency of that sampling may be improved by providing additional input as described there.

1. The Main Program HEATON

The main program is a driver which reads input, and, under its control, calls additional input and processing routines, and calculational routines.

A complete description of the input is given in Section IX. The first card read provides the main specifications: NHIST is the number of histories to be run for each region, for each time bin (or for each point if point detectors are specified). NSTAT ($\leq$200) is the size of a group of histories for the purpose of source generation. ITCUT is a CPU time cutoff: the calculations will be properly terminated at the end of a time bin, and a restart tape will be written if CPU time is nearing that limit. IT1 is the first time bin to be considered (IT1=1 at first, if no restart tape is available). IT2 is the last time bin to be considered. If IPD=0, the last time bin IT2 is treated like any other ones. If IPD>0, the temperatures are calculated only at the given IPD point detectors for time bin IT2. IREZ should normally be equal to zero. If set equal to 1, geometry input will not be expected, but geometry information will be picked up from a restart tape even if IT1=1. IPLO is a flag controlling the quantity of printed output and calls to subroutine PLOT. If IPLO is even (including zero or blank), all input quantities are printed back. If odd, the bulk of the output is suppressed, except for estimated temperatures. Subroutine PLOT will be called if IPLO is equal to 2 or 3. LRN controls the initialization of the random number routine. Leaving that as blank will leave the random number as 1 if IT1=1, or as recorded on the restart tape. If non-zero, it must be an odd integer.

The geometry input (if called for) is read in by subroutine GENI (see Section V). The coordinates X, Y, Z, of a point within the configuration must also be provided.

The next item of input deals with reflection regions and translationally repetitive arrays. These are described in Section VII.9.

Optional input to improve the efficiency of source generation is read in by the main program. The meaning of that input will become apparent in Section 5. For those regions for which no input has been specified, or for which none is available from the restart tape, the information is generated by subroutine SOUSET. If either input or generated, the information is checked by subroutine SOUCK.

The physical compositions are defined next. The temperature dependence of the diffusion parameters is defined as follows:

Product of density and specific heat:

$$\rho C = r_0 + r_2 T^2$$

Thermal conductivity

$$K = k_0 + k_2 T^2, \text{ where } k_2 \leq 0.$$

The input consists of the parameters r_0, r_2, k_0, k_2 for the NCOMP materials present in the problem. The first material entered is assumed to be the main one. The transformations discussed in Section II.5 are performed on its basis.

The meaning of the quantities DFAST is described in Section VI.1.

If IT1=1, the initial conditions are read in. The reading of the boundary conditions is done by subroutine REATIM.

If IT2=0, the coordinates x,y,z of point detector are read in and the subroutine REGCAL is called to calculate the temperature at that point.

If IT2>IT1, the calculation of region-averaged temperatures is done by subroutine TIMSTP.

Two restart tapes are generated: TAPE 8 involves processed geometry information, including information learned about neighbors, and source generation information. TAPE 9 involves the temperatures at the end of the time step.

2. Subroutine REATIM

The subroutine reads the parameters of boundary conditions at the end of the current time step. Linear interpolation is assumed between the beginning and the end of the time step.

3. Subroutine TIMSTP

The subroutine searches for the region with the highest known temperature, examines its neighbors, and, for each region for which a calculation is needed, calls the subroutine REGCAL.

4. Subroutine REGCAL

The subroutine either calls the source generating routine SOUSET, or picks up the coordinates of a detector point, and performs NHIST calls to subroutine GETEMP (see Section VI.1). It calculates the mean and root mean square temperature and prints one line:

JR = region number

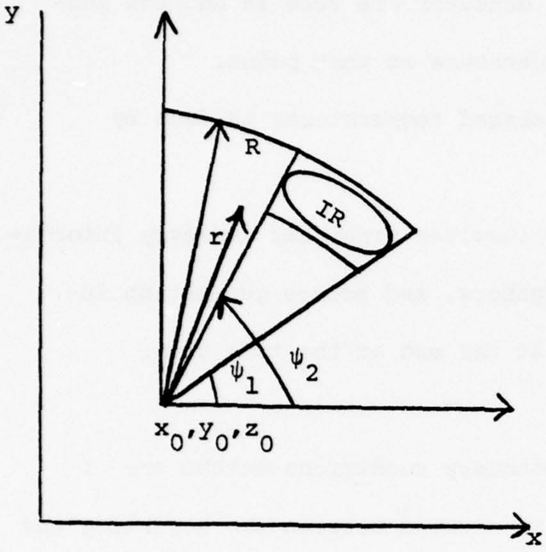
TQ = mean temperature T

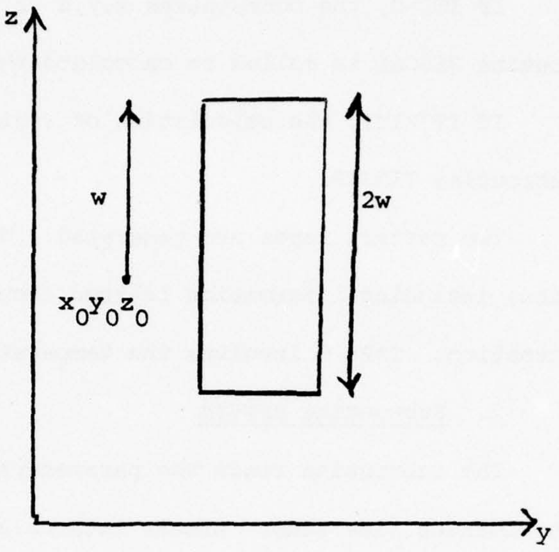
TR = root mean square temperature

$$\left.\begin{array}{l}\text{TR} = \text{mean } \Theta \\ \text{TS} = \text{root mean square } \Theta\end{array}\right\}\text{ see Section II.5}$$

CPU = CPU time in seconds for that calculation.

5. <u>Subroutine SOUSET</u>





The subroutine produces NSTAT points uniformly distributed in region IR, under the asusmption that the region is completely enclosed by the sector of a circular cylinder defined by the two figures above. The relevant parameters are:

XSTART(1,IR) = x_0 }
XSTART(2,IR) = y_0 } central point on axis
XSTART(3,IR) = z_0 }

XSTART(4,IR) = R Outer radius

XSTART(5,IR) = r Inner radius

XSTART(6,IR) = W Half Height

AZIM(1,IR) = ψ_1 = Minimum azimuth

AZIM(2,IR) = ψ_2 = Maximum azimuth

The code samples points uniformly distributed in the sector, and rejects the sample if the point is outside of IR. In the process, the subroutine checks for point in IR with a distance to the axis which is either greater than R or smaller than r. If such points are found, R and r are properly updated, and points previously generated within the current aggregate are properly corrected if necessary. No such check is performed on W, ψ_1, ψ_2.

The efficiency of the routine depends on the fit of the sector to region IR.

6. Subroutine REGSET

Subroutine REGSET generates the XSTART parameters for those regions for which that input has not been provided. The minimum azimuth is set equal to 0, and the maximum azimuth to 2π; thus only full cylindrical annuli are generated.

The first region considered is one for which any internal point P=(x,y,z) is known. If none other, it may be the point read in by the main program after the subroutine GENI has been called. A ray is fired isotropically from P and points of intersection of this ray with region IR are generated. The point P is moved to the midpoint between the pair of points which are most distant from each other. A new ray is fired from the new P position, and intersections are added to the list.... The process terminates when 200 points have been generated on the surface of IR. This population of points is then fitted by the "best" cylindrical annulus in the z-direction.

The fitting is actually to the smallest finite cylinder in the z-direction. The cylinder extends from the smallest z-coordinate to the largest z-coordinate. The smallest circle enclosing the population of x, y points is determined. This is done as follows. A first pass through all pairs of points determines the pair of points x_1,x_2 with the largest separation distance. A tentative circle of radius equal to one half of that distance is centered at the midpoint of that pair. If all points are within that circle, the tentative circle is the smallest possible circle. To find out, a next pass through all the points determines the point x_3 with the largest distance to the tentative center. If that distance is larger than the tentative radius, the circle is drawn as passing through the points x_1, x_2, x_3. Although this last circle is not necessarily the smallest possible one, it is kept as is.

During the process of ray firing, the neighbor IRP of region IR is determined at each intersection. If no starting point has yet been determined for IRP, the intersection point (slightly displaced) is stored as one.

7. Subroutine SOUCK

This subroutine performs the same operations as subroutine REGSET, but with many more checks. It also keeps track of the efficiency. The content of the XSTART and AZIM arrays is printed. If the efficiency becomes intolerable, the hollerith information BAD is also printed, thus suggesting that source information be specified on input.

8. Subroutine PLOT

This subroutine provides a rather limited capability of obtaining printer plots of x-y cuts through the geometry.

9. Reflection Regions and Translationally Repetitive Arrays

Only regions described as single planes can be declared as reflection regions. If the corresponding composition number ICOMP is set to any positive number, reflection boundary conditions will be imposed on the plane defining that region.

The geometrical description of a configuration consisting of groups of regions called cells which are translationally repetitive can be simplified to the description of only one cell with the use of translationally repetitive arrays. The conditions for repetitive geometry are given by the following.

Given a vector $\vec{D}$, a repetition number m and a total number of regions $n\cdot m$, if a point $\vec{X}$ is in region i then point $(\vec{X}+\vec{D})$ is in region $(i+n)$, provided $i+n<n\cdot m$, and point $(\vec{X}-\vec{D})$ is in region $(i-n)$, provided $i-n>0$ (see Figure 1).

For the configuration shown in Figure 2, m repetitive cells can be defined. To describe the entire configuration, only the primary cell needs to be defined. Thus, in Figure 3, consider the first cell which can be described by n regions. The boundary surfaces are described by regions i_1 and i_2, such that $0<i_1\leq n, 0<i_2\leq n$. The rest of the cell can be arbitrarily subdivided into $(n-2)$ regions numbered 1, 2, ...n excluding i_1 and i_2. The configuration is composed of m of these cells so that the external configuration surfaces are defined by regions i_1 and $i_2 + (m-1)\cdot n$. The boundary conditions at these surfaces are determined by the signs of JCOMP(i_1) and ICOMP$(i_2+(m-1)\cdot n)$. If ICOMP(J)≤ 0, the usual conventions hold, i.e., 0-known temperature, <0-radiation types; if ICOMP(I)>0, reflection boundary conditions are imposed.

The input to describe repeating arrays and reflection regions consists of a list of up to 10 region numbers. The first two regions correspond to regions i_1 and i_2 defined above, i.e., regions on either side of the first cell. The remaining regions in the list are reflection regions in the first cell. The repetition number is also read in. If there is only one region number in the list or if m=1, then every region in the list is a reflection region.

With the use of repeating arrays, the region and helps (ITEM 6) input is reduced in that only one cell of n regions need be described. Computer memory requirements are also reduced. Temperature, composition and DFAST array input still consists of data for all n.m regions.

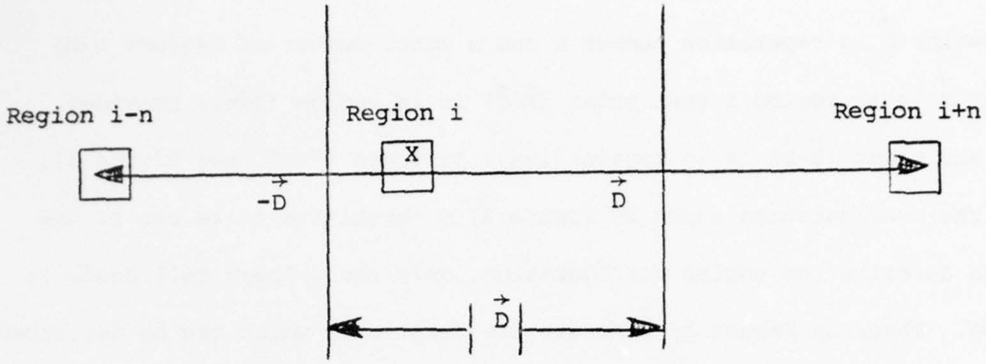


FIGURE 1 - Translationally Repetitive Geometry

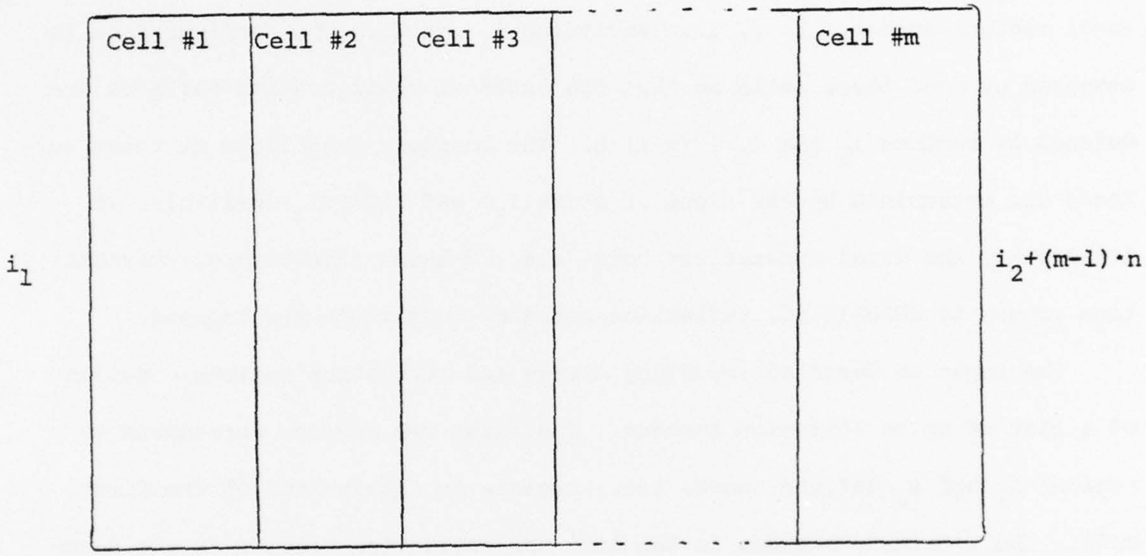


FIGURE 2 - Configuration with Repetitive Cells

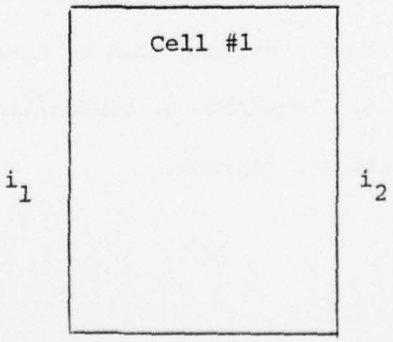


FIGURE 3 - Primary Cell

VIII. ADDITIONAL PROGRAMMING INFORMATION

Table 2 provides information on large arrays. Table 3 gives information on all the subroutines provided.

TABLE 2

DIMENSION OF LARGE ARRAYS

COMMON BLOCK	ARRAY NAME	DIMENSION*
BLANK	IBLOC	NDQ1
	ISLOC	NDQ1
	MAS	NDQ
SOU	XS	$3 \cdot n_1$
	XSTART	$6 \cdot$NDQ2
	AZIM	$2 \cdot$NDQ2
INP	DIFF	NDQ3
	AAK	NDQ3
	TO	NDQ3
	T1	NDQ3
RES	ICOMP	NDQ3
FAST	DFAST	NDQ3
DI	ALP	n_2
	ALS	n_2
	GG	n_2
	CC	n_2
	GQ	$3 \cdot n_2$

*NDQ = large enough to store geometry information and neighbor information

NDQ1 $\geq$ NS = number of surfaces

NB = number of bodies

NR = number of geometrical regions

NDQ2 $\geq$ NR = number of geometrical regions

NDQ3 $\geq$ NPHYS = NBI*NR = number of physical regions

n_1 = 200 = maximum size of statistical aggregates

n_2 = 200 = maximum number of eigenfunctions

TABLE 3. INFORMATION ON SUBROUTINES

SUB \ COMMON	BLANK	BEER	DD	DI	DIFF	FAST	G	GEOM	INP	KGAUSS	REFL	RES	SOU	STAT	THTEM	Reads Input	DESCRIBED IN SECTION
HEATON	X		X		X	X		X	X	X	X	X	X	X	X	X	VII.1
CUBE	X		X		X	X		X			X	X					VI.4
DIFFUS	X		X		X			X									VI.2
DIMOR		X	X		X			X									VI.7
EARLTG				X	X												VI.8
EARLXG					X												VI.8
ECHERF																	VI.8
GAUSS										X							VI.8
GBOX	X							X									V.4
GENI	X						X	X								X	V.3
GENICK	X						X	X									V.3
GENIP	X						X	X									V.3
GETEMP	X	X	X		X	X		X	X		X			X	X		VI.1
G1	X							X					X				V.4.5
G1P	X							X					X				V.4.5
MENDEL		X	X					X				X					VI.6
MOVE			X					X									VI.3
NEBFND	X							X									V.4.3
NORMAN	X		X					X									VI.5
PLOT	X		X					X					X			X	VII.8
PICKT					X												VI.8
PICKTG			X	X	X												VI.8
PICKX					X												VI.8
PICKXG				X	X												VI.8
REATIM			X					X	X		X	X	X		X	X	VII.2
REGCAL	X							X	X		X	X	X	X	X		VII.4
REGCK	X					X		X									V.4.2.2
REGFND	X							X									V.4.4
REGSET	X							X					X				VII.6
SOUCK	X							X					X				VII.7
SOUSET	X							X					X				VII.5
SURFCK	X							X									V.4.2.1
TIMSTP	X							X	X		X	X	X				VII.3
TRANS				X	X												VI.8

IX. INPUT DESCRIPTION

The card input consists of the following items:

Item 1 FORMAT (7I5,I20)

NHIST Number of histories

NSTAT Group of histories ($\leq$200)

ITCUT Real time cutoff

IT1 First time bin to be treated ($\geq$1)

IT2 Last time bin to be treated ($\geq$IT1)

IPD Number of point detectors

IREZ Restart option switch (1 on, 0 off)

IPLO Output control: input printed back if IPLO=0 or 2
plot routine called if IPLO=2 or 3

LRN Random number initialization (0-use default - must be
odd positive otherwise)

Omit Items 2-4 if IT1>1 or IREZ≠0

Item 2 GEOMETRY INFORMATION

Item 2.1 Header card FORMAT (I4,19A4)

IPR If zero, input and processed data printed

HOLL Any hollerith information

Item 2.2 Surface input FORMAT (2X,A3,8E9.3)

TYP Surface type identifier

EPS "Surface thickness"

X
Y coordinates of vertex
Z

A_1-A_4 - as specified in the following table:

TABLE 4

COLUMN	3-5	6-14	15-23	24-32	33-41	42-50	51-59	60-68	69-77
SURFACE TYPE	TYP	ε	VERTEX			A_1	A_2	A_3	A_4
PLANE	PLN	ε	X	Y	Z	H_1	H_2 (inside normal)	H_3	
SPHERE	SPH	ε	X	Y	Z	R (radius)			
CYLINDER	CYL	ε	X	Y	Z	H_1	H_2 (axis)	H_3	R (radius)
CONE	CON	ε	X	Y	Z	H_1	H_2 (axis)	H_3	T (tan)
END OF LIST	END								

Item 2.3 Body input FORMAT (2X,A3,15I5)

TYP Body type: only GEN allowed. END is end of list.

$$\left.\begin{array}{l} \pm IS_1 \\ \pm IS_2 \\ --- \end{array}\right\} \text{List of signed surface numbers}$$

Item 2.4 Region input FORMAT (2X,A3,15I5)

ANY Any three characters. END is end of list.

$$\left.\begin{array}{l} \pm IB_1 \\ \pm IB_2 \\ --- \end{array}\right\} \text{List of signed body numbers}$$

<u>Item 3</u> FORMAT (3E15.5)

$$\left.\begin{array}{l} X \\ Y \\ Z \end{array}\right\} \text{Coordinates of any point within the configuration}$$

<u>Item 4</u> Repeating Array - Reflector regions

Item 4.1 FORMAT (10I5)

NREF No. of regions given (NREF$\leq$10)

$$\left.\begin{array}{l} IR_1 \\ IR_2 \\ --- \\ IR_{NREF} \end{array}\right\} \text{Region numbers}$$

NBI Repetition number

<u>Note</u> - Omit Item 5 if IPLO=0 or 1

Item 5   x-y plot input   FORMAT (5E12.4)

For each plot enter

x_1
y_1 } coordinate of left-upper corner
z

x_2 } coordinates of right lower corner ($x_2 > x_1$, $y_2 < y_1$)
y_2

A blank card terminates plot input

Item 6   Optional input for source generation

Item 6.1 FORMAT (I5)

NHE   Number of regions for which input is provided

Note - omit Item 6.2 if NHE=0

Item 6.2 FORMAT (I5,10E7.0)

For NHE regions, enter a card consisting of the description:

IR   Region number

The following defines a sector of a cylindrical annulus parallel to the z-axis, which encloses region IR:

x_0
y_0 } Coordinates of midpoint on axis
z_0

R   Outer radius of cylinder

r   Inner radius of cylinder

w   Half height of cylinder

$A \cos\psi_1$ } A arbitrary - ψ_1 = minimum angle
$A \cos\psi_1$ } If A=0, ψ_1 is set to 0

$B \cos\psi_2$ } B arbitrary - ψ_2 = maximum angle
$B \sin\psi_2$ } If B=0, ψ_2 is set to 2π

(Notation refers to Figure in Section VII.5)

* Information for selected regions can be entered in any order. The information currently entered for region IR will be recorded on a restart tape. That information will overide information recorded on a previous restart tape (if any).

Item 7 COMPOSITION INFORMATION

Item 7.1 FORMAT (I5)

NCOMP: Number of compositions

Item 7.2 FORMAT (4E15.5)

For each of the NCOMP compositions, enter:

$$\left.\begin{array}{l} RC_0 \\ RC_2 \end{array}\right\} \quad \begin{array}{l} \text{parameters defining the product of density} \\ \text{and heat capacity } \rho C = RC_0 + RC_2 \cdot T^2 \end{array}$$

$$\left.\begin{array}{l} K_0 \\ K_2 \end{array}\right\} \quad \begin{array}{l} \text{parameters defining the heat conductivity} \\ K = K_0 + K_2 \cdot T^2 \end{array}$$

(The first composition entered is the main composition of the problem.)

Item 7.3 FORMAT (10I5)

(ICOMP(IR),IR=1,NR) For regions internal to the configuration, enter composition number. For external regions, enter 0 if temperature is known, and -1 if boundary condition of the radiation type applies. Enter any positive number for reflection regions.

Item 8 INFORMATION FOR "SMALL BOX APPROXIMATION" FORMAT (8E10.2)

(DFAST(IR),IR=1,NR) Enter zeroes for no approximation. See text, otherwise: if DFAST<0, diffuse with box of fixed size |DFAST| . If DFAST>0 diffuse with either actual box, or box of size DFAST, whichever is larger.

Note - Omit Item 9 if IT1>1

Item 9 INITIAL CONDITIONS

Item 9.1 FORMAT (E15.5)

TB1: Initial value of the time variable

Item 9.2 FORMAT (5E15.5)

(T1(IR),IR=1,NR): Initial temperatures for all regions.

Note - Repeat items 10-12 for each time bin.

Item 10 FORMAT (E15.5)

TB: value of the time variable at end of time bin

Item 11 Boundary conditions FORMAT (5E15.5)

T2(IR1)	Final temperature of first region with ICOMP(IR1)$\leq$0
T2(IR2)	Final temperature of second region with ICOMP(IR2)$\leq$0
⋮	
T2(IRL)	Final temperature of last region with ICOMP(IRL)$\leq$0

Item 12 Coefficients of surface heat transfer. FORMAT (5E15.5)

H(IR1)
H(IR2)

H(IRL)

} Enter value of coefficient if ICOMP(IRN)<0.
Leave blank if ICOMP(IRN)=0

Note - Omit Item 13 if IPD=0

Enter Item 13 IPD times

Item 13 Detector input FORMAT (3E15.5)

x
y
z

} Coordinates of the point detector

Item 14 END OF FILE

Input Tapes

Tape 5 - Card input

Tape 8 - Geometry restart tape - needed if IT1>1, or if IT1=1 and IREZ=1.

Tape 9 - Initial conditions - needed if IT1>1.

Output Tapes

Tape 6 - Printer

Tape 8 - Geometry restart tape

Tape 9 - Final conditions - generated if IT2>1 or if IT2=1 and IPD=0.

X. REFERENCES

1. A. Haji-Sheikh and E. M. Sparrow, "The Solution of Heat Conduction Problems by Probability Methods", J. Asme, 89 (1967).

2. M. E. Muller, "Some Continuous Monte Carlo Methods for the Dirichlet Problem:, Ann. Math. Stat., 27 (1956).

3. H. S. Carslaw and J. C. Jaeger, "Conduction of Heat in Solids", Second Edition, Oxford University Press, London (1959).

4. M. O. Cohen *et al.*, "SAM-CE: A Three Dimensional Monte Carlo Code for the Solution of the Forward Neutron and Forward and Adjoint Gamma Ray Transport Equation".

DISTRIBUTION LIST

No. of Copies	Organization
12	Commander Defense Documentation Center ATTN: DDC-TCA Cameron Station Alexandria, VA 22314
1	Commander US Army Materiel Development and Readiness Command ATTN: DRCDMA-ST 5001 Eisenhower Avenue Alexandria, VA 22333
1	Commander US Army Aviation Systems Command ATTN: DRSAV-E 12th and Spruce Streets St. Louis, MO 63166
1	Director US Army Air Mobility Research and Development Laboratory Ames Research Center Moffett Field, CA 94035
1	Commander US Army Electronics Command ATTN: DRSEL-RD Fort Monmouth, NJ 07703
1	Commander US Army Missile Command ATTN: DRSMI-R Redstone Arsenal, AL 35809
1	Commander US Army Tank Automotive Development Command ATTN: DRDTA-RWL Warren, MI 48090
2	Commander US Army Mobility Equipment Research & Development Command ATTN: Tech Docu Cen, Bldg. 315 DRSME-RZT Fort Belvoir, VA 22060
1	Commander US Army Armament Command Rock Island, IL 61202
1	Commander US Army Frankford Arsenal ATTN: SARFA-FCV, Stan Goodman Philadelphia, PA 19137
1	Commander US Army Harry Diamond Labs ATTN: DRXDO-TI 2800 Powder Mill Road Adelphi, MD 20783
1	Director US Army TRADOC Systems Analysis Activity ATTN: ATAA-SA White Sands Missile Range NM 88002
1	Commander US Army Research Office ATTN: CRD-AA-IP P. O. Box 12211 Research Triangle Park NC 27709
1	Director Lawrence Livermore Laboratory ATTN: Tech Info Div P. O. Box 808 Livermore, CA 94550

DISTRIBUTION LIST

No. of Copies	Organization
1	Director Los Alamos Scientific Lab ATTN: Rept Lib P. O. Box 1663 Los Alamos, NM 87544
1	Mathematical Applications Group, Inc. 3 Westchester Plaza Elmsford, NY 10523
1	Science Applications, Inc. 1250 Prospect Street ATTN: Dr. E. Straker La Jolla, CA 92037

Aberdeen Proving Ground

Marine Corps Ln Ofc
Dir, USAMSAA