HYBRID COMPUTER SOLUTIONS OF PARTIAL DIFFERENTIAL EQUATIONS BY MONTE CARLO METHODS

Warren D. Little

Process Computer Engineering, Canadian General Electric Company
Peterborough, Ontario

INTRODUCTION

In addition to finite-difference methods, Monte Carlo methods also are known for solving certain partial differential equations. When implemented on a digital computer, however, the Monte Carlo methods have generally proven to be very inefficient. In 1960, a study carried out at the University of Michigan described analog computer techniques for mechanizing Monte Carlo methods. From the Michigan study it became evident that a fast analog computer together with a small digital computer and a modest interface could obtain Monte Carlo solutions at rates competitive with standard finite-difference methods.

With rare exception, the Monte Carlo methods that have been programmed on either an analog or digital computer have been for elliptic partial differential equations (e.g., Laplace’s equation). In this paper the classical Monte Carlo methods are generalized to yield methods for solving parabolic equations (e.g., the diffusion equation) as well as homogeneous and nonhomogeneous elliptic equations of a very general form. Techniques for implementing the Monte Carlo methods on a hybrid system consisting of a general-purpose analog computer and a general-purpose digital computer as well as some typical results are also discussed.

PARTIAL DIFFERENTIAL EQUATIONS FOR CONTINUOUS MARKOV PROCESSES

The Monte Carlo methods to be discussed are based upon partial differential equations that can be written for continuous Markov processes. A continuous Markov process is a stochastic process having the property that future values of a stochastic variable \( \bar{r} \) depend only upon present and not past values. To solve a boundary value problem that is defined on an open bounded region \( R \) and its boundary \( C \), a continuous Markov process defined on \( R \) and \( C \) must be simulated.

To facilitate description, consider a three-dimensional Markov process with stochastic vector \( \bar{r} = (x,y,z) \) where components \( x, y \) and \( z \) are given by the following stochastic differential equations:

\[
\frac{dx}{dt} + A_1(x,y,z,t) = B_1(x,y,z,t)N_1(t) \quad (1)
\]

\[
\frac{dy}{dt} + A_2(x,y,z,t) = B_2(x,y,z,t)N_2(t) \quad (2)
\]

\[
\frac{dz}{dt} + A_3(x,y,z,t) = B_3(x,y,z,t)N_3(t) \quad (3)
\]

The coefficients \( A_i \) and \( B_i \) are, in general, slowly varying continuous functions of \( x,y,z \) and \( t \). The driving terms \( N_i(t) \) are uncorrelated with each other.
and each term is ideally Gaussian white noise with power spectral density $2D_i$.

For a Markov process as defined by Eqs. (1) to (3), two relevant conditional probability density functions $f$ and $g$ are defined.

(A) $f(\mathbf{r}_0, t_2 | \mathbf{r}_0, t_0) \, dr_2$ is the probability that the stochastic vector $\mathbf{r}$ is in $dr_2$ at time $t_2$ if at time $t_0$, $\mathbf{r} = \mathbf{r}_0$.

(B) $g(\mathbf{r}_0, t_b | \mathbf{r}_0, t_0) \, dr_b dt_b$ is the probability that the stochastic vector $\mathbf{r}$ will reach boundary $C$ for the first time within $dr_b$ between times $t_0$ and $t_0 + dt_b$ if at time $t_0$, $\mathbf{r} = \mathbf{r}_0$.

The notation used in definitions (A) and (B) is illustrated in Fig. 1.

The conditional probability density functions defined above satisfy the following initial and boundary conditions

(A) $\lim_{t_0 \to t_2} f(\mathbf{r}_2, t_2 | \mathbf{r}_0, t_0) = \delta(\mathbf{r}_2 - \mathbf{r}_0)$

where $\delta(\mathbf{r}_2 - \mathbf{r}_0) = 0$ for $\mathbf{r}_2 \neq \mathbf{r}_0$

and $\int \delta(\mathbf{r}_2 - \mathbf{r}_0) \, dr_2 = 1$.

(B) $\lim_{t_0 \to t_2} g(\mathbf{r}_2, t_2 | \mathbf{r}_0, t_0) = 0$

(C) $\lim_{\mathbf{r}_0 \to \mathbf{r}_0} f(\mathbf{r}_2, t_2 | \mathbf{r}_0, t_0) = 0$

(D) $\lim_{\mathbf{r}_0 \to \mathbf{r}_0} g(\mathbf{r}_2, t_2 | \mathbf{r}_0, t_0) = \delta(\mathbf{r}_2 - \mathbf{r}_0) \delta(t_2 - t_0)$

where $\delta(\mathbf{r}_2 - \mathbf{r}_0) \delta(t_2 - t_0) = 0$ unless $\mathbf{r}_2 = \mathbf{r}_0$ and $t_2 = t_0$.

and $\int_{t_0}^{t_2} \int \delta(\mathbf{r}_2 - \mathbf{r}_0) \delta(t_2 - t_0) \, dr_2 \, dt_2 = 1$

For the above Markov process it can be shown that the density functions $f$ and $g$ also satisfy the following backward Kolmogorov partial differential equations: 4,5

\[ \frac{\partial f}{\partial t} (\mathbf{r}_2, t_2 | \mathbf{r}_0, t_0) = L \cdot f(\mathbf{r}_2, t_2 | \mathbf{r}_0, t_0) \]

\[ \frac{\partial g}{\partial t} (\mathbf{r}_2, t_2 | \mathbf{r}_0, t_0) = L \cdot g(\mathbf{r}_2, t_2 | \mathbf{r}_0, t_0) \]

where

\[ L \cdot f(\mathbf{r}_2, t_2 | \mathbf{r}_0, t_0) = a_i(\mathbf{r}_0, t_0) \frac{\partial}{\partial x_i} f(\mathbf{r}_0, t_0) + b_i(\mathbf{r}_0, t_0) \frac{\partial^2}{\partial x_i^2} f(\mathbf{r}_0, t_0) \]

\[ L \cdot g(\mathbf{r}_2, t_2 | \mathbf{r}_0, t_0) = a_i(\mathbf{r}_0, t_0) \frac{\partial}{\partial y_i} g(\mathbf{r}_0, t_0) + b_i(\mathbf{r}_0, t_0) \frac{\partial^2}{\partial y_i^2} g(\mathbf{r}_0, t_0) \]

\[ + \frac{\partial}{\partial z_i} b_i(\mathbf{r}_0, t_0) \frac{\partial^2}{\partial z_i^2} g(\mathbf{r}_0, t_0) + \frac{\partial^2}{\partial z_i^2} g(\mathbf{r}_0, t_0) \]

The coefficients $a_i(\mathbf{r}_0, t_0)$ and $b_i(\mathbf{r}_0, t_0)$ are moments of the Markov process and are related to the coefficients $A_i$ and $B_i$ of the stochastic differential equations by Eqs. (7) and (8).

\[ a_i(\mathbf{r}_0, t_0) = -A_i(\mathbf{r}_0, t_0) \]

\[ b_i(\mathbf{r}_0, t_0) = \left[ B_i(\mathbf{r}_0, t_0) \right]^2 \]

By simulating the Markov process given by Eqs. (1) to (3), boundary value problems with partial differential equations of the same form as Eqs. (4) and (5) can be solved. More general partial differential equations in which the dependent variable and not only its derivative exit can also be solved by considering auxiliary probability density functions $u$ and $v$ as defined below.

\[ u(\mathbf{r}_2, t_2 | \mathbf{r}_0, t_0) = \exp \left[ -m(t_2, t_0) \right] f(\mathbf{r}_2, t_2 | \mathbf{r}_0, t_0) \]

\[ v(\mathbf{r}_0, t_0) = \exp \left[ -m(t_2, t_0) \right] g(\mathbf{r}_0, t_0) \]

Figure 1. Space region illustrating terms used in definitions.
HYBRID COMPUTER SOLUTIONS OF PARTIAL DIFFERENTIAL EQUATIONS

In these definitions the brackets

\[ r(t_0) = r_0 \]

denote a conditional expectation; that is, the expected value of the function within the brackets subject to the condition that \( r(t_0) = r_0 \) and \( r(t_2) \) is in a small region \( dr_2 \) about \( r_2 \). The term \( m(t_2, t_0) \) is the integral of a slowly varying continuous positive function \( d(r,t) \) of the stochastic vector \( r(t) \) and \( t \). That is,

\[ m(t_2, t_0) = \int_{t_0}^{t_2} d(r(t), t) dt \] (11)

The auxiliary probability density functions \( u \) and \( v \) satisfy the following partial differential equations.

\[ \frac{\partial v}{\partial t_0} (r_0, t_0) = L r_0,0 (r_0, t_0) \]

\[ - \frac{\partial v}{\partial t_0} (r_0, t_0) - d(r_0, t_0) v(r_0, t_0) \] (12)

\[ \frac{\partial v}{\partial t_0} (r_0, t_0) = L r_0,0 (r_0, t_0) \]

\[ - \frac{\partial v}{\partial t_0} (r_0, t_0) - d(r_0, t_0) v(r_0, t_0) \] (13)

Equations (4), (5), (12) and (13), together with initial and boundary conditions (A) to (D), will now be used to develop Monte Carlo methods for solving a large class of partial differential equations.

MONTE CARLO METHODS FOR SOLVING PARTIAL DIFFERENTIAL EQUATIONS

In this section, relationships between probability density functions \( f, g, u \) and \( v \) and solutions \( \phi \) of boundary value problems will be developed. The methods apply to problems in which \( \phi \) itself is given initially and on all boundaries. For all problems the solution at a point within a region \( R \) is obtained as the expected value of initial and boundary values at the terminal points of random walks that originate at the point for which the solution is desired. The expected value is written in terms of the functions \( f, g, u \) and \( v \). An approximation to the expected value is determined experimentally from a large number of random walks simulated on an analog computer.

Consider the following boundary value problem involving a parabolic partial differential equation.

**Problem A**

Determine \( \Phi(r_0, t_0) \) such that:

\[ \Phi(r_0, t_0) = L r_0,0 \Phi(r_0, t_0) \] (14)

is satisfied within a bounded region \( R \);

(2) a piecewise continuous initial condition \( \phi_0(r_0) \) is satisfied within \( R \); i.e.,

\[ \Phi(r_0, 0) = \phi_0(r_0) \] (15)

(3) a piecewise continuous boundary condition \( \phi_c(r_0, t_0) \) is satisfied on the boundary \( C \) of \( R \); i.e.,

\[ \Phi(r_0, t_0) = \phi_c(r_0, t_0) \] (16)

The boundaries and initial and boundary conditions of a typical problem with one space variable are shown in Fig. 2. Note that \( \Phi(r_0, t_0) \) is a function of the initial position \( r_0 \) and starting time \( t_0 \). The time \( t_0 \) is defined to be negative so that random walks take place in the time interval \( t_0 \leq t \leq 0 \).

To obtain the solution \( \Phi \) of Problem A at a point \( (r_0, t_0) \), random walks are started at \( (r_0, t_0) \) and each walk is terminated as soon as a boundary is reached or at \( t = 0 \). If a walk terminates on a boundary at some \( (r_0, t_0) \) the boundary value \( \phi_c(r_0, t_0) \) is recorded, whereas if a walk is terminated at \( t = 0 \) with position \( r_2 \) the initial value \( \phi_0(r_2) \) is recorded. The expected value of the initial and boundary values obtained in this manner is a solution of Problem A.

The expected value defined above can be written in terms of density functions \( f \) and \( g \) as follows:

\[ \Phi(r_0, t_0) = \int_R \phi_0(r_0) (r_0, 0) \Phi(r_0, t_0) dr_0 \]

\[ + \int_C \int_0^t \phi_c(r_0, t_0) g(r_0, t_0) \Phi(r_0, t_0) dr_0 dt_0 \] (17)

The fact that \( \Phi(r_0, t_0) \) as given above is indeed a solution of the problem can be shown by operating on both the right and left side of Eq. (17) with the operator \( \frac{\partial}{\partial t_0} + L r_0,0 \). This gives
\[
\left( \frac{\partial}{\partial t_0} + L_{T_{r_0,t_0}} \right) \phi(T_{r_0,t_0}) = \\
\int_R \phi_e(T_{r_0}) \left( \frac{\partial}{\partial t_0} + L_{T_{r_0,t_0}} \right) f(T_{r_0},0 \mid T_{r_0,t_0}) \, dr_0 \\
+ \int_{t_0}^0 \int_C \phi_e(T_{r_0,t_0}) \left( \frac{\partial}{\partial t_0} + L_{T_{r_0,t_0}} \right) g(T_{r_0,t_0} \mid T_{r_0,t_0}) \, dr_0 dt_0 \\
- \int_C \phi_e(T_{r_0,t_0}) g(T_{r_0,t_0} \mid T_{r_0,t_0}) \, dr_0 
\] (18)

The right side of Eq. (18) is zero by Kolmogorov Eqs. (4) and (5) and initial condition (B). Thus \( \phi(T_{r_0,t_0}) \) satisfies the partial differential equation of Problem A.

By direct application of initial and boundary conditions (A) to (D) to Eq. (17), it also follows that \( \phi(T_{r_0,t_0}) \) satisfies the initial and boundary conditions of Problem A.

The Monte Carlo solution of Problem A is obtained by approximating the expected value \( \phi(T_{r_0,t_0}) \) given by Eq. (17) with the average \( \phi_N(T_{r_0,t_0}) \) of initial and boundary values \( \phi_i \) that are recorded from a set of \( N \) random walks originating at \( T_{r_0,t_0} \). This average is

\[
\phi_N(T_{r_0,t_0}) = \frac{1}{N} \sum_{i=1}^N \phi_i 
\] (19)

Steady-state solutions of equations of the type considered in Problem A are the solutions of an important class of elliptic partial differential equations. Consider the following problem for this type of equation.

**Problem B**

Determine \( \phi(r_0) \) such that:

1. \( L_T \phi(T_{r_0}) = 0 \)
2. a piecewise continuous boundary condition \( \phi_e(T_0) \) is satisfied on the boundary \( C \) of \( R \), i.e.,

\( \phi(T_{r_0}) = \phi_e(T_{r_0}) \) (21)

The subscript \( t_0 \) is deleted from operator \( L_{T_{r_0,t_0}} \) to indicate that \( L_{T_{r_0}} \) is independent of time.

The solution of Problem B is the solution of a corresponding problem of type A for \( t_0 = -\infty \). Since Problem B is independent of time, the required expected value can be obtained by starting random walks at \( t_0 = 0 \) rather than \( t_0 = -\infty \) and allowing each walk to continue until a boundary point is reached. The expected value of the boundary values at such terminal points is

Figure 2. Random walks and initial and boundary conditions of a typical problem with one space variable.
As was shown for Problem A, it follows the expected value of Eq. (22) is a solution of Problem B. An average of the same form as that of Eq. (19), in which the \( \theta_i \) are boundary values selected by random walks starting at \( \bar{r}_o \), approximates the expected value and is the Monte Carlo solution.

Solutions of Problems A and B can be combined to yield solutions of a class of nonhomogeneous partial differential equations as defined by Problem C.

**Problem C**

Determine \( \Phi(\bar{r}_o) \) such that:

\[
\begin{align*}
1) & \quad L_{\bar{r}_o}\Phi(\bar{r}_o) = -H(\bar{r}_o) \\
2) & \quad \Phi(\bar{r}_o) = \Phi_b(\bar{r}_o) \text{ on the boundary } C \text{ of } R; \text{ i.e.,} \\
3) & \quad \Phi(\bar{r}_o,0) = H(\bar{r}_o)
\end{align*}
\]

To obtain the solution of Problem C, consider the time-independent boundary value problem of type B:

\[
\begin{align*}
1) & \quad L_{\bar{r}_o}\Phi(\bar{r}_o) = 0 \text{ within } R, \\
2) & \quad \Phi(\bar{r}_o) = \Phi_b(\bar{r}_o) \text{ on the boundary } C \text{ of } R; \text{ and}\\n3) & \quad \Phi(\bar{r}_o,0) = H(\bar{r}_o)
\end{align*}
\]

The solution of Problem C is given in terms of the solutions of the two subproblems by

\[
\Phi(\bar{r}_o) = \Phi(r_o) + \int_{-\infty}^{0} \Phi(\bar{r}_o,t) dt
\]

This solution is verified by substituting Eq. (25) into the conditions of Problem C and using the information provided by the subproblems.

The Monte Carlo solution of Problem C is obtained by determining \( \Phi(\bar{r}_o) \) and \( \Phi(\bar{r}_o,t) \) by the methods of Problems A and B and then integrating \( \Phi(\bar{r}_o,t) \) with respect to \( t \) by some numerical technique.

More general partial differential equations than those for which Monte Carlo methods have been outlined can be solved by using Eq. (12) and (13) for density functions \( u \) and \( v \). From the definitions of \( u \) and \( v \) (Eqs. (9) and (10)) it follows that \( u \) and \( v \) satisfy the same initial and boundary conditions as density functions \( f \) and \( g \) respectively. It therefore follows that problems similar to Problems A, B, and C but with \( L_{\bar{r}_o,t'_o} \) and \( L_{\bar{r}_o} \) replaced by \( L_{\bar{r}_o,t'_o} - d(\bar{r}_o,t'_o) \) and \( L_{\bar{r}_o} - d(\bar{r}_o) \) respectively have solutions that can be written as expected values with respect to functions \( u \) and \( v \). For example,

\[
\frac{\partial \Phi}{\partial t_o}(\bar{r}_o,t_o) = L_{\bar{r}_o,t'_o}\Phi(\bar{r}_o,t_o) - d(\bar{r}_o,t_o)\Phi(\bar{r}_o,t_o)
\]

with initial and boundary conditions as given by Problem A has solution

\[
\Phi(\bar{r}_o,t_o) = \int_{R} \Phi_o(\bar{r}_o) u(\bar{r}_o,0 | \bar{r}_o,t_o) d\bar{r}_o
\]

\[
+ \int_{t_o}^{0} \int_{C} \Phi_o(\bar{r}_o,t) v(\bar{r}_o,t | \bar{r}_o,t_o) d\bar{r}_o dt_o
\]

From definitions (9) and (10), Eq. (27) becomes

\[
\begin{align*}
\Phi(\bar{r}_o,t_o) &= \int_{R} \Phi_o(\bar{r}_o) u(\bar{r}_o,0 | \bar{r}_o,t_o) d\bar{r}_o \\
& \quad + \int_{t_o}^{0} \int_{C} \Phi_o(\bar{r}_o,t) v(\bar{r}_o,t | \bar{r}_o,t_o) d\bar{r}_o dt_o
\end{align*}
\]

From the collection of the Computer History Museum (www.computerhistory.org)
The expected value \( \varphi(\bar{r}_o,t_o) \) given by Eq. (28) can be approximated by the average \( \varphi(\bar{r}_o,t_o) \) of the product \( \gamma_i \varphi_i \) for walks originating at \( (\bar{r}_o,t_o) \) where \( \varphi_i \) is the initial or boundary value at the terminal point of the \( i \)th walk and \( \gamma_i \) is the value

\[
\gamma = \exp \left[ - \int_{t_o}^{T} d(\bar{r}(t),t) \, dt \right]
\]

for the corresponding walk. The upper limit of integration, \( T \), is 0 for walks terminating at \( t = 0 \), and \( t_b \) for walks terminating at a boundary. The Monte Carlo solution is therefore

\[
\varphi(\bar{r}_o,t_o) = \frac{1}{N} \sum_{i=1}^{N} \gamma_i \varphi_i
\]

In a manner similar to above it follows that if \( L_r \) in Problem B is replaced by \( L_r - d(\bar{r}) \) the Monte Carlo solution is as given by Eq. (30). In this case each \( \varphi_i \) is a boundary value selected by a random walk that starts at \( \bar{r}_b \) and each \( \gamma_i \) is a corresponding value of

\[
\gamma = \exp \left[ - \int_{t_b}^{t_o} d(\bar{r}(t)) \, dt \right]
\]

The superposition of solutions of Problems A and B to obtain solutions of Problem C also applies to equations containing the \( d(\bar{r}) \) term.

**HYBRID COMPUTER MECHANIZATION OF MONTE CARLO METHODS**

A computing system for mechanizing the Monte Carlo methods that have been developed must carry out the following operations.

(A) Simulate stochastic differential Eqs. (1) to (3).
(B) Evaluate \( \gamma \) as given by Eq. (29) or (31).

![Figure 3. Hybrid computer system for Monte Carlo methods.](image)

![Figure 4. Block diagram for simulation of a stochastic differential equation.](image)

(C) Terminate solution of the stochastic differential equations at either time \( t = 0 \) or whenever a boundary is reached.
(D) Generate the initial or boundary values corresponding to the terminal points of the random walks.
(E) Form the averages given by Eq. (19) or (30).

Automatic readout of the solutions \( \varphi(\bar{r}_o,t_o) \) and adjustment of \( (\bar{r}_o,t_o) \) after each set of \( N \) random walks is also desirable.

A hybrid system in which synchronism of the two computers is realized by a mode-control flip-flop is shown in Fig. 3. Operations (A) to (D) listed above are performed by the analog computer whereas operation (E) as well as adjustment of the analog computer and readout of solutions are handled by the digital computer.

**SIMULATION OF STOCHASTIC PROCESSES**

An analog computer block diagram for simulating stochastic differential Eq. (1) in its most general form is shown in Fig. 4. The function generation indicated in this figure can be realized simply with diode function generators and multipliers whenever closed-form mathematical expressions are known for the functions or whenever the functions are of only a single variable. Special techniques are required for generation of functions of a more general form.

Integrator A2 shown in Fig. 4 is used to "track-and-hold" the random variable \( x \). This track-and-hold feature is used to hold the terminal value of the random vector \( \bar{r} \), and consequently the initial or boundary value \( \varphi_i \) generated from \( \bar{r} \) at a constant value while \( \varphi_i \) or \( (\gamma_i \varphi_i) \) is read by the digital computer.
The mode-control signal \( c \) and its logical inverse \( \bar{c} \) synchronize the track-and-hold modes of integrator \( A2 \) with the compute and initial-condition modes respectively of integrator \( A1 \).

The uncorrelated Gaussian white noise terms \( N_i \) that drive the stochastic differential Eqs. (1) to (3) must be simulated on the analog computer by noise sources that are physically realizable. Noise sources with characteristics that approximate the ideal characteristics can be derived from gas tubes, pseudorandom noise generators \(^8\) or discrete-interval binary noise generators.\(^9\) The example solutions given in this paper were obtained with a discrete-interval binary noise generator that was multiplexed to give three essentially uncorrelated noise channels.

For problems in which \( \gamma \) must be generated, the following implicit method is used. From Eq. (29)

\[
\frac{d\gamma}{dt} = -d(F(t),t)\gamma \\
\gamma(t_0) = 1
\] (32)

The response \( \gamma \) of these equations is obtained easily on the analog computer. The mode of the integrator that is used to simulate Eq. (32) is controlled by the signal \( c \) of the mode-control flip-flop.

DETECTION OF BOUNDARIES

The mode-control flip-flop is set when a random walk reaches a boundary or at the terminal time \( t = 0 \) by using the outputs of suitably driven analog voltage comparators. The flip-flop is reset by the digital computer after the boundary or initial value at the terminal point of the random walk has been read into the digital computer for averaging. For example, for the problem illustrated in Fig. 2 the scheme shown in Fig. 5 would be used to trigger the mode-control flip-flop.

The boundaries of problems with two or more spatial variables can be detected by using function generators together with voltage comparators. Consider the two-dimensional region \( R \) shown in Fig. 6, in which the curves \( C_i(X) \) that form \( C \) are both single-valued functions of \( X \). It is clear from the figure that a random walk with instantaneous components \((x,y)\) reaches \( C \) whenever \( y = C_i(x) \) or \( y = C_j(x) \). These boundary detection criteria are implemented on the analog computer by the method shown in Fig. 7.

The boundaries of a region as shown in Fig. 8 in which \( C_i(U) \) and \( C_j(U) \) are single valued functions of position \( U \) along a dividing line \( D \) are detected by using a coordinate transformation. That is, variables \( x \) and \( y \) are transformed to \( u \) and \( v \) by the transformation

\[
u = x \cos \beta - y \sin \beta - a \\
v = x \sin \beta + y \cos \beta - b
\] (33)
Figure 8. Coordinate transformation used for boundary detection of two-dimensional regions.

In \( u, v \) coordinates the criteria that a random walk is at the boundary is \( v = C_1(u) \) or \( v = C_2(u) \).

The boundaries of simply connected regions of arbitrary shape are detected by dividing the region into simple regions \( R_1, R_2, \ldots, R_n \) that have dividing lines \( D_i \). The exit of a random walk from each simple region is detected by the previous methods. The resultant signals are combined with an AND gate to give a signal when the walk leaves all \( R_i \) and hence the total region \( R \).

The boundaries of circles and ellipses of the form\[ (X - a)^2 + (Y - c)^2 = 1 \]are detected by comparing the function \( f(x,y) = \left( \frac{x + a}{b} \right)^2 + \left( \frac{y + c}{d} \right)^2 \) with 1. Hence only two multipliers and one comparator are required for these common regions.

The preceding methods can be generalized to three-dimensional regions by using a dividing plane separating the boundary into surfaces that are single-valued functions of position on the plane. For boundary surfaces which are simple functions of the two plane variables, the method is easy to apply. If this is not the case, special purpose function generator techniques \( \text{7} \) are necessary.

The boundaries of three-dimensional regions with some type of symmetry can often be detected by combining the methods described for one- and two-dimensional regions. For example, cubic regions are detected by using three pairs of comparators in the same manner that a single pair is used for one-dimensional problems. In addition, the boundaries of spheres and ellipsoids can be detected with three multipliers and a single comparator in the same manner that two multipliers and a comparator are used for circles and ellipses.

**GENERATION AND AVERAGING OF INITIAL AND BOUNDARY VALUES**

At the instant a boundary \( C \) is reached, the mode-control flip-flop is triggered from a comparator by the methods that have been discussed. The triggering of this flip-flop places track-and-hold amplifiers in the hold mode so that the terminal values of the components \( x, y, \) and \( z \) of \( r \) are available as constant voltages on the analog computer. The initial and boundary values \( \theta_i \) are generated with function generators from these components of \( r \).

The function generation prescribed above can be carried out with function generators and multipliers whenever the boundary values are known as simple functions of \( x, y \) and \( z \) or whenever they can be expressed as a function of a single variable. For two-dimensional problems in which a dividing line \( D \) is used for detecting the boundaries, the boundary values are conveniently generated as a function of the variable \( u \) defined along the dividing line.

When the values \( \theta_i \) cannot be generated conveniently by analog computer techniques, they can always be generated within the digital computer. When the digital computer is used for function generation, the components \( x, y \) and \( z \) of the terminal position vector are read; then a table stored within the computer is scanned, or some other method is used, to determine the corresponding value of \( \theta_i \). Since more than one value must be read by the digital computer and since additional digital operations are required, this procedure with slow digital equipment is more time-consuming than analog function generation. However, with fast digital equipment it is possible to store the terminal components \( x, y \) and \( z \) of the \( i \)th walk with a track-and-hold arrangement and read them during the \((i + 1)\)st walk. Thus, if the conversion equipment and digital computer are sufficiently fast, the values \( x, y \) and \( z \) can be read and the digital function generation for the \( i \)th walk can be carried out while the analog computer is simulating the \((i + 1)\)st random walk. This procedure is very efficient in that essentially no time is wasted between walks.
The average \( \phi \) for each point \((r_0, t_0)\) at which a solution is desired is formed by adding, in sequence, each value \( \phi_i \) (or \( y_i \)) to a partial sum stored within the digital computer. A tally of the number of random walks that have been completed is kept by the digital computer and after \( N \) walks the average is typed out. A digital computer, or at least a digital method, is necessary for the averaging operation because for \( N \) large (1000-40,000) a large dynamic range is required to obtain the sum precisely. After the solution has been obtained at a point the digital computer adjusts the starting point \((r_0, t_0)\) on the analog computer so that the solution at another point can be obtained. In this manner the solutions at all points of interest are obtained.

**EXAMPLES**

The examples given in this paper were selected from a large number of problems that were solved using an EAI 231 R-V analog computer and a Logistics Research Alvac III-E digital computer. With these slow computers, it was possible to simulate about 10 random walks per second. To obtain solutions with a small variance, 1000 random walks were used for each point solution.

Solutions of a one-dimensional time-independent problem for three parameter values are shown in Fig. 9. When \( K/D_1 = 0 \), the average duration of a random walk for this problem is

\[
T(r_0) = \frac{1 - x_0^2}{2D_1} \tag{34}
\]

Since \( T(r_0) \) is very easily measured, Eq. (34) provides a simple method for determining the power spectral density \( 2D_1 \) of the noise source.

The solutions of another one-dimensional problem for three different boundary values are shown in Fig. 10. This problem is of the type that requires generation of the functional \( y \). According to Eq. (31) the required functional is

\[
y = \exp \left[ -D_1 \int_{t_0}^{t_b} (1 - x^2) \, dt \right] \tag{35}
\]

As a final example, solutions of the diffusion equation

\[
\frac{\partial \phi}{\partial t_0}(r_0, t_0) = \nabla^2 \phi(r_0, t_0) \tag{36}
\]

with initial condition \( \phi_0(r_0) = -1 \) and boundary condition \( \phi_r(r) = 1 \) are shown in Fig. 11. The solutions shown are for the center of a line, square and cubic region. From this example, it is significant to note that the average time \( T(0) \) for a random walk to reach a boundary decreases with the dimension of the problem.

Figure 9. Solutions of

\[
D_1 \frac{d^2 \phi}{dx_0^2} + \frac{\partial \phi}{\partial t_0} = 0, \text{ for } \phi(-1) = -1 \text{ and } \phi(1) = 1.
\]

Figure 10. Solutions of

\[
\frac{d\phi}{dx_0} - (1 - x_0^2) \phi = 0, \text{ for } \phi(-1) = -1 \text{ and } \phi(1) = A.
\]

From the collection of the Computer History Museum (www.computerhistory.org)
CONCLUSIONS

Monte Carlo methods have been developed for obtaining approximate solutions to partial differential equations of a very general form. The methods are easily mechanized with a small analog computer coupled by means of a modest interface to a small digital computer.

The Monte Carlo solutions are obtained sequentially in a point by point manner. Therefore, if solutions at only a few points are required the methods may be more efficient, even using slow computers, than conventional finite-difference methods. On fast computers, and especially special-purpose fast computers, the methods should provide a powerful means for solving many types of partial differential equations.

REFERENCES


