


Technical Report No. 151

04967-4-T

AN ALGORITHM FOR CONCURRENT
RANDOM WALKS ON HIGHLY PARALLEL COMPUTERS

by

R. F. Rosin

Approved by: 
B. F. Barton

COOLEY ELECTRONICS LABORATORY
Department of Electrical Engineering
The University of Michigan
Ann Arbor

Submitted in partial fulfillment of the requirements for
the degree of Doctor of Philosophy in
The University of Michigan

July 1964

PREFACE

The author wishes to thank the members of his Doctoral Committee for their advice and counsel in this work. Special appreciation is given to Professor Bernard A. Galler for his encouragement and guidance, to Professor Leonard J. Savage for sharing first interest in this work and to Professor William A. Ericson for continuing this interest upon Professor Savage's later absence from the Committee. Thanks are also due to Mr. James Dickey who participated in early discussions of this research and who contributed the paper included as Appendix II.

The experimental work in this thesis would have been impossible without the facilities and, equally important, the cooperation of the senior and operating staffs of The University of Michigan Computing Center.

Finally, the author is very grateful for the support of this research by the Cooley Electronics Laboratory and the Information Systems Laboratory of the Department of Electrical Engineering. In particular, thanks go to the Cooley Clerical Staff, for preparing the early drafts, Mrs. Marthalee Shellman, for manuscript preparation and Mr. Anthony Adaschik, for preparation of the figures.

TABLE OF CONTENTS

PREFACE	iii
LIST OF ILLUSTRATIONS	vii
LIST OF SYMBOLS	viii
LIST OF TABLES	x
LIST OF APPENDICES	xi
ABSTRACT	xiii
CHAPTER I: INTRODUCTION	1
1.1 Survey of Parallel Computation	2
1.2 Survey of the Random Walk	4
CHAPTER II: THE CONCURRENT RANDOM WALK ALGORITHM A_1	5
2.1 Definition of the Algorithm and Comparison with Alternatives	5
2.2 Discussion of the Concurrent Algorithm	8
CHAPTER III: PARAMETERS WHICH DESCRIBE SIMULATION ACCORDING TO THE ALGORITHM A_1	9
3.1 Definition of G	9
3.2 Other Parameters	11
3.2.1 t as a Function of N and k	12
3.2.2 E as a Function of P and k	12
3.2.3 k as a Function of e	13
3.2.4 Obtaining k , E and P for Symmetric Random Walks in Rectangular Two-Dimensional Spaces from the Results of McCrea and Whipple	13
CHAPTER IV: THEORETICAL ANALYSIS	15
4.1 Analysis of the One-Dimensional Random Walk with Central Source Location	15
4.2 A Lower Bound on G for Spaces of Dimension Two and Higher	17
4.2.1 A Lower Bound on G	17
4.2.2 Comments on the Theorem of 4.2.1	21
4.3 An Estimate of G Based on the Geometry of Two-Dimensional Spaces	23
4.4 Summary of Theoretical Results	25
CHAPTER V: EXPERIMENTAL ANALYSIS OF TWO-DIMENSIONAL SPACES	25
5.1 Experiments on $L \times L$ Squares with Central Source Location	26
5.1.1 Experimental Design	26
5.1.2 Preliminary Experiments	28
5.1.3 Experimental Results	30
5.2 Experiments on Spaces with Irregular Boundaries	35
5.2.1 Experimental Design	35
5.2.2 Specification of the Spaces	37
5.2.3 Experimental Results	42
5.3 Interpretation of Results	42
5.3.1 General Comments on G	42
5.3.2 E as a Function of Geometry	43
5.3.3 D as a Function of Geometry	45
5.3.4 G as a Function of Geometry	46

TABLE OF CONTENTS (Continued)

CHAPTER VI: CONCLUSIONS	49
6.1 The Use of the Algorithm A_1 on Highly Parallel Computers	49
6.2 The Use of the Algorithm A_1 on Contemporary Computers	51
6.3 Suggestions for Further Research	51
REFERENCES	91
DISTRIBUTION LIST	93

LIST OF ILLUSTRATIONS

<u>Figure</u>	<u>Title</u>	<u>Page</u>
1	One-dimensional space with central source location	16
2	G as a function of p for values of K = 1, . . . , 6	18
3	Example of a boundary point for which the probability of particle absorption is $\leq 1/2$	22
4	Partition of a space for the purpose of estimating G	24
5	Flow of information in the experiments on square spaces	27
6	k, P and E as a function of L	31
7	1/1-k as a function of L	32
8	Results for single point target sets in square spaces	33
9	Various target sets used	34
10	G as a function of the number of points in "corners" and "sides" target sets	36
11	Design for experiments on irregular spaces	37
12	Information flow in the experiments on spaces with irregular boundaries	38
13	First stage in definition of an irregular space	39
14	Second stage in definition of an irregular space	39
15	Final stage in definition of irregular space no. 1	40
16	Additional spaces defined	41
17	E as a function of E_e	44
18	E as a function of L for a 5 by L rectangle, Source at point (3, 3)	46
19	E as a function of E_e for squares and 5 by L rectangles	47
20	D as a function of $E^{1/2}$	48
21	G as a function of $E_e^{1/2}$	49
22	General organization of the computer	54
23	Individual processor for the Laplace equation	55
24	Four-bit multiplier for optimum over-relaxation module	56
25	Concurrent random walk processor for drum machine	60
26	Counter module for random walk processor	61

LIST OF SYMBOLS

<u>Symbol</u>	<u>Meaning</u>	<u>Defined by, or first used in</u>
A_1	concurrent random walk algorithm	Section 2.1
A_0	conventional random walk algorithm	Section 2.1
A_2	alternate concurrent algorithm	Section 2.1
A_i^*	algorithm A_i for use in a single module of a highly parallel computer	Section 2.1
G	gain of A_1 over A_0	Section 3.1
E	expected number of active particles	Section 3.1
T	target set	Section 3.1
N	number of particles simulated	Section 3.1
x	number of particles absorbed	Section 3.1
p	probability of absorption	Section 3.1
q	probability of non-absorption	Section 3.1
σ_i^2	variance of x using algorithm i	Eq. (3.1)
D	measure of ineffectiveness of A_1 compared to A_0	Eq. (3.2)
P	expected path length of a particle	Section 3.2
t	expected number of time steps for an N -particle simulation	Section 3.2
e	expected number of times a particle will return to the source	Section 3.2
P^*	expected number of time steps to absorb remaining particles	Section 3.2.1
T^*	total particle activity	Section 3.2.2
f	fraction of its life a particle is at the source	Section 3.2.3
F_1, F_2	functions used to evaluate particle activity	Eq. (3.19)
K	number of points on a line segment	Section 4.1
p_ℓ, p_r	probabilities of particle motion	Section 4.1
X_i	random variable defined by absorption of particle i	Section 4.2.1
S, \bar{S}	sets of particles	Section 4.2.1
ρ	correlation coefficient	Section 4.2.1
$\hat{\rho}$	maximum correlation	Eq. (4.12)
$\bar{\rho}$	average correlation	Eq. (4.28)
s	grid point	Section 4.2.2
I_s	event that particle i is at s	Section 4.2.2
W	grid point adjacent to T	Section 4.2.2
a_i	area of subspace i	Eq. (4.39)
E_e	estimate of E	Eq. (4.39)
L	length of one side of a square	Section 5.1.1
L^*	number of symmetry classes	Section 5.1.1

c_i	symmetry class i	Section 5.1.1
$n(c_i)$	number of elements in c_i	Section 5.1.1
D_i	D for target set consisting of point i	Eq. (5.4)
D_{c_i}	D for target set consisting of c_i	Eq. (5.4)
\bar{D}	average D	Eq. (5.5)
$\hat{\sigma}^2$	estimate of variance	Eq. (5.7)
R	number of replications	Section 5.1.2
$D_{1/2}, D_{1/4}$	D for target set consisting of one-half or one-fourth of total points	Section 5.1.2
$G_{1/2}, G_{1/4}$	G based on corresponding D	Section 5.1.2
r	number of replications	Section 5.2.1
G_Σ	G for a specific subset of points in T	Section 5.2.1
G_i	G for point i	Section 5.2.1
G_{\min}	minimum G_i	Section 5.2.1
p_{\min}	p at point determining G_{\min}	Section 5.2.1
g	lower bound on G	Eq. (5.21)

LIST OF TABLES

<u>Table</u>	<u>Title</u>	<u>Page</u>
Ia	\bar{D} as a function of N and R for a 13 by 13 square space	29
Ib	$\hat{\sigma}^2(\bar{D})$ as a function of N and R for a 13 by 13 square space	29
II	k, P and E as a function of L	30
III	G and G_{\min} as a function of L	30
IV	$G_{1/4}$ and $G_{1/2}$ as a function of L	34
V	G as a function of the number of points in "corners" and "sides" target sets	35
VI	Results of experiments on irregular spaces	43

LIST OF APPENDICES

A-1. A Special Purpose Computer for Partial Differential Equation Solution and Other Iterative Algorithms	53
A-2. Implementation of the Algorithm A_1 on the Computer of Appendix A-1	59
B. Concurrent Random Walks in a One-Dimensional Space with the Source at the Center	62
C. Computer Programs Used in this Research	65
1. RDEP	65
2. MOVE	67
3. ISRR	69
4. NDEP	70
5. MVX	72
6. MVY	73
7. ISRA	75
8. RNCSA	75
9. ECOMP	76
10. PROBS	77
11. COVAN	79
12. NXOUT	83
13. DANAL	84
D. Philosophical and Practical Insights into the Use of a Computer as a Research Tool	86

ABSTRACT

The purpose of this research is to propose and evaluate an algorithm for concurrent simulations of symmetric absorbing boundary random walks on highly parallel computers.

The algorithm to be studied is presented and compared with the conventional nonconcurrent algorithm and with other concurrent algorithms. There follows a definition of G , which is a measure of effectiveness of the new algorithm as compared to the conventional one, and definition of several other parameters which describe simulations according to the new algorithm.

A two-part theoretical discussion is then presented. In the first part, the algorithm is analyzed for one-dimensional spaces with the source in the center. Then a lower bound is established for the general case. Whereas, $G \geq 1$ for the one-dimensional spaces, the proof in the second part of this theoretical study shows that $G \geq \frac{1}{2}$, in general, and that $G \geq 1$ subject to two intuitively reasonable but unproven constraints.

An experimental study of G for simulation of symmetric random walks in two-dimensional spaces is then presented. First considered are random walks in spaces with square absorbing boundaries of various sizes and with centrally located sources. This is followed by a study of random walk simulations in spaces with irregular boundaries. The results of these experiments demonstrate that (a) $G > 1$ for any two-dimensional space, (b) when the objective is to determine the absorption probability for a very small set of boundary points, G increases linearly with the number of nonabsorbing points near the source, (c) based on the preceding results, a prediction formula for G as a function of the geometry of the space is verified.

The main conclusions to be drawn from this work are (a) the proposed algorithm is at least as effective as the conventional algorithm for simulating absorbing boundary random walks on highly parallel computers for one- and two-dimensional spaces, (b) in many cases (e.g., small target sets), the algorithm is much more effective than the conventional one and this effectiveness can be predicted without random walk simulation.

CHAPTER I INTRODUCTION

The study of digital computer organization can be subdivided readily into three areas:

- a) The distribution of structure between fixed (hardware) and variable (program) substructures of a computer system.
- b) The distribution of structure among the hardware substructures.
- c) The distribution of structure among the program substructures.

In addition, there has been a recent increase of interest in the relationship between the problem to which a computing system is applied and the organization of the system itself. Examples of contemporary computers organized with a class of applications in mind are:

Burroughs B-5000	Compiler language translation and execution
Control Data Corp. 6600	Multiple access
Westinghouse Corp. SOLOMON	Iterative computations in two or more dimensions

It is, therefore, appropriate to add the following item to the list of areas defining computer organization:

- d) The relationship between computer structures and the problems to which they are applied.

This dissertation presents an investigation of a particular problem in this fourth aspect of computer organization. The problem is to define and evaluate an algorithm for simulation of absorbing boundary random walks in any two-dimensional space on a computer organized for efficient execution of highly iterative algorithms in parallel. To this end several subtopics are presented:

1. Definition of the algorithm studied and reasons underlying its choice.
2. Selection of a measure suitable for evaluation of the effectiveness of the algorithm.
3. Determination of various other parameters describing the behavior of the algorithm.
4. Determination of a method for estimating this effectiveness measure directly from the geometry of the space without recourse to computer simulation.

The remainder of this chapter is devoted to brief surveys of parallel computer systems and random walk simulation. In Chapter II the algorithm to be studied is defined and compared to some alternative choices. In the third chapter the measure of effectiveness G and several other parameters for describing the algorithm are defined. Chapter IV contains a discussion of theoretical investigations of the effectiveness of the algorithm, and a proposed method for estimating G .

Chapter V consists of a description of the experimental plan used to determine G for symmetric random walks in two-dimensional spaces and includes a summary of the results of the experiments performed. Conclusions as to the overall effectiveness of the algorithm are presented in Chapter VI along with suggestions for further research in this area. This is followed by several appendices and a list of references.

1.1 Survey of Parallel Computation

The evolution of the contemporary computer has been marked repeatedly by the concept of parallel or concurrent computation. In the 19th century, Babbage (Ref. 1), being well aware of the iterative nature of certain processes, provided for the repeated use of a fixed set of operation cards with changing address cards for his analytical engine. He also suggested that counters, similar to today's index registers, should be provided on this machine. These ideas were not realized until the mid-20th century.

The early relay-tube machines of the 1940's allowed for several control units to operate more-or-less in parallel. The IBM Selective Sequence Electronic Calculator (1948) is an example of an implementation of this form of concurrent computation. The IBM 650 and UNIVAC I computers provided for an overlapping of input-output processing and calculation, and more recent developments have led to the use of subsidiary input-output hardware systems. This technique has been refined further into some of today's systems in which input-output processing is accomplished by separate, sophisticated computers. The sophistication of these systems has led to the development of correspondingly complicated sets of programs to take full advantage of the compute/input-output overlap.

In the realm of parallelism in computation, as distinguished from compute/input-output processing overlap, the first major development was the Williams Computer at Manchester University in the late 1940's. Although often singled out primarily for being the first machine to use the cathode ray tube for central storage, its pioneering use of the B-box or B-line was probably its most important contribution to the current technology. This special device was an embodiment of the realization that the nature of the arithmetic performed on computer addresses is much simpler and less powerful than general computation. By building a simple, fast store and processor, address modification could be realized at the same time as initial instruction decoding in this machine.

More recently, computers such as the IBM 7030 (Stretch), CDC 6600 and Bull Gamma 60 have exploited very much the same concept in allowing the simultaneous processing of several forms of arithmetic (e.g., floating point and integer, addition and multiplication, calculation and memory search). Several systems, such as the CDC 3600, Univac LARC and Burroughs B-5000, allow more than one central processor to time-share the same set of storage modules, permitting, in general, completely concurrent processing (subject to conflicting demands for storage access).

Most recent developments (also involving software) have led to multi-access time sharing systems such as the MIT CTSS (Ref. 2). Users of these systems address the computer and its associated software systems through consoles (usually of a teletype nature, but sometimes more elaborate) which can be remote from the computer itself. A dialogue between the user and the system takes place with what appears to be immediate response by the machine to the man at the console. However, the sheer speed of contemporary equipment allows the computer actually to accept, process, and respond sequentially to commands received from several consoles, with the delay involved remaining generally less than that detectable by the human users. Often, batch processing "background" jobs share central processor time. However, since there is at present only one central computer involved in these systems, these are not examples of truly general concurrent processing.

Developments of a yet unrealized nature have tended toward the concurrent processing of algorithms normally executed in a serial manner. von Neumann (Ref. 3) proposed a modular computer capable of carrying out a very large number of simultaneous computations of a very primitive nature as well as being able to reproduce itself. Holland (Ref. 4) proposed the concept of the iterative circuit computer consisting of a large array of identical processors each with one word of storage and each capable of relatively powerful computation. As in von Neumann's machine, each module can communicate with any of its immediate neighbors, and activity is transmitted from module to module rather than staying in one fixed location as in a conventional computer where the central processor retains activity as information is transmitted to and from it. Parallel computation is effected when a module transmits its activity to two or more modules instead of along a single path. Developments in this direction have been offered by Squire and Palais, Gonzalez, and Comfort (Refs. 5-7).

Slotnick (Ref. 8) has proposed a form of iterative circuit computer called SOLOMON lying somewhere between the contemporary general-purpose machine and the iterative circuit organizations described above. In this computer a limited two-dimensional array of identical processing elements, each with a limited amount of storage, is capable of receiving commands from a more classical memory and obeying these commands in parallel. As in the earlier iterative circuit computers, modules may address the stores of adjacent modules; however, activity is controlled by a set of states which are assigned to both modules and instructions. A version of this machine with a 32 by 32 array of processing elements has been under development for manufacture by Westinghouse Corporation.

A large class of applications of an iterative multidimensional nature (one dimension of time and two or more of space) appears to have influenced the parallel organization of other proposed computers as well as SOLOMON (e.g., Ref. 10). The work reported in this dissertation was motivated by the desire to determine the effectiveness of a computer of this class, which was invented independently by the author, for simulating random walks (Appendix A).

For the purposes of this research, a parallel computer is defined to be one which, in one unit of time, can simulate either one time-step of the proposed algorithm (Chapter II), or one time-step of the conventional algorithm, but one in which the individual modules do not contain enough storage for each one to simulate a conventional random walk. The SOLOMON computer fits this definition for all cases such that each grid point in the space in which random walks are to be simulated can be matched by a distinct processing element of the computer. Thus, for some geometries and some numbers of modules, SOLOMON will not fit this definition.

1.2 Survey of the Random Walk

In 1658, Huygens¹ proposed a variation to the classical problem of points; namely, to determine the duration of the game. James Bernoulli solved this problem in 1801 and it was treated further by DeMoire, Lagrange and Laplace. Feller (Ref. 12) demonstrates the equivalence between this problem, the gamblers' ruin problem, and the one-dimensional random walk.

The concept of a random walk per se has been in existence for several years; at least since 1928 when Courant et al. (Ref. 13) stated the problem of determining the expected termination point of a particle wandering in random steps of uniform length in a two-dimensional lattice. McCrea and Whipple (Ref. 14) in 1940 appear to have solved this problem subject to two constraints. The first is the uniform distribution of values of the random variable determining the direction to be taken at the next time step; i.e., $P(\text{North}) = P(\text{South}) = P(\text{East}) = P(\text{West}) = 1/4$. We shall call a simulation subject to this constraint a "symmetric random walk." The second constraint applied was that the absorbing boundaries must describe a rectangle, semi-infinite strip, infinite strip, infinite half-plane or infinite plane.²

Examination of the modern literature reveals a moderate amount of material on general application of random walk techniques which will be briefly examined in later paragraphs. However, published accounts of the computer implementation or analysis of these models are very scarce. Perhaps this is due to the governmental and commercial secrecy surrounding some areas in which random walk models appear to be useful. One example of a report of this sort of investigation is a pair of papers (Refs. 15, 16) which discuss random walk simulation with spontaneous particle emission. As in the majority of papers on the analytic aspects of random walks, the McCrea and Whipple paper is cited, serving to point up the fundamental applicability of these earlier results.

With the advent of atomic and particle physics, the applicability of the random walk simulation was emphasized. Brown (Ref. 17) discusses the relationship between the random walk simulation and classical diffusion processes. He develops the analogy between the simple random

¹This and the following historical citations are found in Ref. 11.

²This paper is discussed more completely in Section 3.2.4.

walk and the solutions to Laplace's equation, and then discusses generalizations of the method, particularly the changing of transition probabilities as a function of grid point which leads to the solution of more general partial differential equations.

The random walk model, hence its simulation, has applicability in many areas. Scientists and engineers who are concerned with diffusion processes find the random walk to be a valuable concept. Notable problems to which the concept of random walk may be applied are heat transfer and propagation, diffusion of petroleum in porous media, and gaseous diffusion, as well as the applications in atomic physics. For example, Berger (Ref. 18) discusses the diffusion and penetration of fast charged particles and shows that the random walk concept has played a part in the development of the associated theory.

The random walk is also found in other areas of application. For example, Wall, Windwer and Gans (Ref. 19) show how a highly refined random walk model is used in the study of the structure of polymers (long complex molecular chains). They are careful to point out that there exists an occupancy problem, i.e., no two particles (molecules) can ever occupy the same node. In the various application areas in which the random walk has been used, there are many problems which cannot be solved today in any other way. This partially accounts for the widespread use of this tool.

CHAPTER II

THE CONCURRENT RANDOM WALK ALGORITHM A_1

In this chapter three algorithms for random walk simulation will be presented and discussed. The conventional algorithm will be called A_0 ; the concurrent algorithm to be studied in this thesis will be called A_1 , and an alternate concurrent algorithm to be evaluated here along with A_1 will be called A_2 . Each of these can be presented in two forms: as a set of commands defining the total simulation, and as a set of rules defining the simulation of the activity at one grid point by a single module or processor of a highly parallel iterative computer. It will suffice here to present A_1 in both forms, A_2 in the latter, and A_0 as modification of A_1 .

2.1 Definition of the Algorithm and Comparison with Alternatives

The algorithms to be studied in this dissertation are defined to simulate concurrent random walks in a geometry or space consisting of a simply connected absorbing boundary and an enclosed particle source. The paths which a particle may follow are defined as being unit-length segments of a uniform rectangular grid overlaying the space.

The algorithm assumes the existence of a random variable which, when evaluated for a two-dimensional space, takes on the values "left," "right," "up," or "down" (or, "North," "South," "East," or "West"). The random variable is said to be symmetric if it assumes its values with equal probabilities.

To simulate the random walks of N particles (i.e., an N trial simulation) by the proposed algorithm A_1 :

1. Void the grid of particles.
2. Inject a particle at the source.
3. Evaluate the random variable once and move all particles in the grid one grid unit in the direction indicated by this value of the random variable.
4. Particles which alight on grid points defining the boundary are noted, along with the coordinates of these boundary points, and are said to have terminated or to have been absorbed. They are then removed from the grid.
5. If N particles have been injected into the space, go to step 7.
6. If the grid point defined as the source is not occupied by a particle, go to step 2; otherwise, go to step 3.
7. If N particles have not yet been absorbed, go to step 3; otherwise, stop.

Algorithm A_0 , the conventional algorithm, differs from A_1 only in step 6 which, in this case, should be changed to:

- 6^0 . If all previously emitted particles have been absorbed, go to step 2; otherwise, go to step 3.

Thus, since in A_1 , step 2 can be reached any time the source point is not occupied, it allows the simulation of several particles in one time step, whereas, A_0 allows no more than one active particle in the grid at one time.

A_1 appears to contain a restriction which might be eliminated to allow the simulation of even more particles in every time step. Let us call the algorithm with this restriction removed A_2 . In this algorithm, step 6 becomes:

- 6^2 . Go to step 2.

Note that this allows the insertion of a particle at every time-step, and also allows more than one particle to occupy a single grid point at one time. Many revisions to A_1 can be suggested to allow processing of these multiple occupancies. In the one chosen for study here, step 3 becomes:

- 3^2 . For each particle in the grid which has not been moved in this time-step, perform a separate evaluation of the random variable and move the particle one grid unit in the direction indicated by the random variable.

To allow more complete evaluation of the two algorithms, A_1 and A_2 , they are presented below, reinterpreted so as to specify the activity of a single module or processor of a highly parallel computer. In these definitions, if a module is simulating the source node, it is said to be in "source state" and is assumed to contain a counter, initially set to N, which is called "the insertion counter." Similarly, a module which is simulating a boundary point is said to be in "absorption state" and to contain a counter, initially set to 0, which is called "the absorption counter." The algorithms are:

A_1^*

1. If this module is not in source state, go to step 3.
2. If this module is not occupied by a particle and the insertion counter is greater than zero, insert a particle and decrease the particle insertion counter by 1.
3. If this module is in absorption state, go to step 6.
4. If this module is occupied, transmit the particle to a neighboring module in the direction determined by the random variable value obtained from master control.
5. Go to step 1.
6. If this module is occupied, remove the particle, add 1 to the absorption counter in this module and transmit the new total to master control.
7. Go to step 1.

A_2^*

1. If this module is not in source state, go to step 3.
2. If the particle insertion counter contents is greater than zero, insert a particle (another particle) and subtract 1 from the counter.
3. If this module is in absorption state, go to step 7.
4. If this module is not occupied, go to step 1. (Newly arrived particles are not considered as occupants for this time step.)
5. Evaluate the random variable and move one particle to the neighbor determined by that value.
6. Go to step 4.
7. If this module is occupied, remove one particle and increase the absorption counter by 1 and go to step 7.
8. Transmit the new total of absorptions to master control.
9. Go to step 1.

Comparison of these two algorithms leads to the following conclusions with respect to the relative complexity of A_1^* versus A_2^* : beyond the requirements of A_1^* , A_2^* requires:

1. The processing of more than one particle per time-step of simulation.
2. More than one evaluation of the random variable per time-step for all grid points.
3. Storage of more than one particle in a module.
4. Provision of a separate store for incoming particles in each module.

Items numbered 1 and 2 might be eliminated by using step 4 of A_1^* , in place of steps 4 and 5 in A_2^* . This alternative choice to the solution of the multiple occupancy problem would result in allowing more than one particle per module but processing them all in the same manner and at the same time. It is not reasonable to expect that the advantage gained from using this method for increasing the number of active particles in one time-step outweighs the facts that additional particle storage is required in each module, and that the two or more particles at the same grid point at the same time will follow identically the same paths for major portions of their active lives. Also, for any version of A_2^* , more time will be required to simulate one time-step (given comparable hardware components) than will be required for A_1^* .

Since A_1^* represents the simplest and potentially least expensive module to construct, it will be the one considered in this dissertation.

2.2 Discussion of the Concurrent Algorithm

An often useful intuitive picture of the behavior of a group of particles under A_1 , the new algorithm, can be given as follows. A cloud of particles, similar to a loosely packed ball of fuzz, drifts around in space. The cloud displays no rotation; only translation. At one point in the space, the source, a new particle of fuzz arises whenever the point is not covered. Meanwhile, the boundaries of the space are coated with an absorbent material which retains any fuzz in contact with it.

Note that with this model, as in the random walks, the fuzz can never cover the entire space. Generally, if enough of the fuzz lies to one side of the source so that it touches a boundary, there will be little or none on the other side. The actual behavior is determined by the geometry of the space, that is, the relative spatial locations of the boundary and the source points.

Consider the history of any single particle following the new (concurrent) algorithm A_1 . Clearly, it follows exactly the same rules (algorithm) as does a particle in the conventional random walk; that is, at each time-step it moves one path segment in a direction determined by a random variable, until it is absorbed. Thus, the expectation of its absorption at any boundary point is the same as that for a particle under the conventional scheme, and, if x is the number of particles absorbed at boundary point i in an N trial simulation, then, as in the conventional random walk, x/N is an estimate of p_i , the probability of absorption of a particle at i . If some expected number, E of particles can be said to exist in the grid at one time, then A_1 can be said to simulate random walks at a rate E times faster than the conventional method, A_0 and to yield the same unbiased estimates of the absorption probabilities at the boundary points.

However, consider two particles initiated one time-step apart; the paths of these two particles will be identical (but shifted one grid unit from each other in same direction) until

one or the other is absorbed, whereupon the second can be viewed as initiating a path dependent on the first only in that the starting point of this new path is a point one unit from the termination point of the absorbed particle. Thus, the path histories of the two particles are more highly correlated than in the case where the particles are independent.

Particles separated by more than one time-step will, in general, be less highly correlated, and the particles initiated P or more time-steps apart, where P is the expected path length of a particle,³ will in some average sense be uncorrelated.

Thus, there is, for any simulation, a correlation in time between the position of any particle and the positions of those which precede and follow it. The effect of the correlation is to cause the experimental estimates of the termination probabilities to be less precise than if they were determined by an equal number of independent particles.

Equivalently, a larger number of particles must be simulated using A_1 than are necessary using A_0 to obtain equally precise results.

If one is interested in using A_1 instead of A_0 for simulating random walks, it is desirable to know whether the concurrency achieved by running an average of E particles in every time-step outweighs the additional simulations necessary to obtain results of a given precision. In the next chapter, a measure of effectiveness of A_1 relative to A_0 will be defined which is based on the two factors discussed here. In the fourth and fifth chapters, theoretical and experimental analyses of this measure are carried out for various spatial geometries.

CHAPTER III

PARAMETERS WHICH DESCRIBE SIMULATION ACCORDING TO THE ALGORITHM A_1

Included in this chapter is a definition of G , a measure of the effectiveness of A_1 compared to A_0 on a parallel computer of the type defined at the end of Section 1.1. If $G \geq 1$ for any geometry, then A_1 is at least as effective as A_0 for simulating random walks in that space. This discussion is followed by the definition of several other parameters which also describe a simulation according to A_1 . Relationships between several of these parameters are then derived, and results are obtained for determining these parameters directly from the geometry of any rectangular two-dimensional space in which random walks are simulated.

3.1 Definition of G

As suggested earlier, A_1 can be compared to A_0 by the following two characteristics which must be understood before this new algorithm can be used:

³ P is a measure of time-steps as well as spatial unit since the algorithm moves a particle one path segment in one time-step.

- a) The expected relative number of particles which will be active (i.e., emitted but as yet unabsorbed) in any one time-step for a given spatial geometry. Clearly this is a measure in a multiplicative sense of the positive attributes of the algorithm. Since for A_0 the number of active particles is 1, this characteristic can be represented by E , the expected number of active particles in A_1 .
- b) The expected relative precision of the results of the simulation. Using A_1 will require a larger number of simulations than would be required using A_0 to obtain results of the same precision, so this is a measure of the negative attributes of A_1 .

Through the use of random walk techniques one expects to obtain an estimate of the probability of particle absorption at any point or set of points, called the target set, on the boundary of the space. In actual random walk simulation, this probability is estimated by dividing x , the number of particles absorbed in the target set T , by N , the total number of particles emitted in the simulation. One measure normally applied to determine the precision of an estimate is the variance, and this will be used as a partial measure of the effectiveness of the random walk results.

Consider a conventional random walk simulation carried out according to A_0 which involves simulation of N independent particles. If x is the number of particles absorbed in the target set T , then x/N is an estimate of p , the absorption probability in that set. Letting $q = 1-p$, then the variance of x is

$$\sigma_0^2 = Npq \tag{3.1}$$

The estimates of the variance of x as determined by using the concurrent algorithm A_1 are not readily available, and more will be said about how to determine them in the next sections. In general, they shall be denoted by σ_1^2 . Clearly, σ_1^2 will depend upon the correlation between simulated particles. A measure of relative effectiveness of the new to the old methods is defined to be

$$D = \frac{\sigma_1^2}{\sigma_0^2} = \frac{\sigma_1^2}{Npq} \tag{3.2}$$

This ratio of variances has the property that, if the new variance is proportional to N , (which it might well be), then D will be independent of N and will reflect the correlation effects of the new algorithm as induced by the particular geometry under question. It will also serve to indicate the factor by which N must be multiplied to obtain the number of particles which will yield a variance (imprecision) using A_1 which is at least as good as that for A_0 ; that is, such that $D \leq 1$.

Having obtained two measures, one of effectiveness, E, and one of ineffectiveness, D, for comparing the total effects of using the new algorithm, let us combine them into a single measure of gain of A_1 as compared to A_0 . This definition has been chosen to be the ratio

$$G = \frac{E}{D} \quad (3.3)$$

It compares the number of particles which can be run at one time using the new algorithm (compared to 1 using the old) to the number of particles which must be run using the new algorithm in order to have as precise results as obtained by running one particle under the old.

For example, if $E = 14$ and $D = 7$, then, although the results per particle run are only $1/7$ as precise using the new algorithm, in any time period 14 times as many particles can be run. In other words, equally precise results can be achieved in half the time resulting in a gain, G, equal to two.

Since E and D can be expected to be dependent only on the geometry of the space in which the random walks are simulated and the target set, G also is a function of the geometry and the target set chosen.

In Section 4.3 a method is proposed for estimating G for any two-dimensional space and a target set of one point. This estimate is based only on the geometry of the space.

3.2 Other Parameters

In the discussion which follows, several parameters other than G are presented which will be useful later for describing the behavior of random walks simulated by algorithm A_1 . Several relationships among these parameters are then derived, and a method is presented for their evaluation as a function of the geometry for some two-dimensional spaces.

The parameters are:

N the number of random walks simulated.

P the expected path length of a particle.

t the expected number of time-steps required to simulate N random walks.

k the expected fraction of time-steps during which the source is occupied by an unabsorbed particle.

E the expected number of unabsorbed particles in the grid in any time-step.

e the expected number of times a particle will return to the source before absorption.

In order to estimate an absorption probability with a high degree of precision using random walk simulation, N must be very large. This, in turn, yields the relationship $N \gg P$ which will be used in the following derivations. The relationships derived in Sections 3.2.1, 3.2.2, and 3.2.3 are independent of the dimensionality of the space in which random walks are simulated.

Although it is not generally valid that

$$\text{Ex} \left(\frac{a}{b} \right) = \frac{\text{Ex}(a)}{\text{Ex}(b)} \quad (3.4)$$

where Ex denotes expected value; this approximation is used in many of the derivations which follow. The results obtained from evaluations using these derivations indicate that this approximation is a reasonable one in these cases.

3.2.1 t as a Function of N and k. Conventional random walks require an average of $N \cdot P$ steps to simulate N random walks each with expected path length P. The advantage of the concurrent method is that fewer than $N \cdot P$ time-steps are required for this simulation. Since a number of particles can be simulated at once, what is required in order to simulate concurrent walks is to initiate N particles in N' separate time-steps and then consider that it will require P^* time-steps to absorb all remaining particles. P^* is similar to P since the time required to absorb a set of emitted particles is closely related to the path length of a single particle. Thus,

$$t = N' + P^* . \quad (3.5)$$

N' is greater than N by a factor of $\frac{1}{1-k}$ where k is the fraction of time that the source cannot initiate a new particle because it is occupied by a previously active particle. Hence,

$$t = \frac{N}{1-k} + P^* . \quad (3.6)$$

Since, in general, $N \gg P^*$, the approximation

$$t \approx \frac{N}{1-k} \quad (3.7)$$

will be used in much of what follows, although derivations using (3.6) will be provided also.

3.2.2 E as a Function of P and k. Let the total particle activity, T^* , for N trials in the grid be expressed as

$$T^* = N \cdot P , \quad (3.8)$$

the product of the number of particles and the expected active life of a single particle. T^* can also be expressed as

$$T^* = E \cdot t . \quad (3.9)$$

Equating (3.8) and (3.9), the two expressions for T^* , and using (3.6), one obtains

$$N \cdot P = E \left(\frac{N}{1-k} + P^* \right) , \quad (3.10)$$

which yields

$$E = \frac{N \cdot P(1 - k)}{N + P^*(1 - k)}. \quad (3.11)$$

As $N \gg P^*$,

$$E \approx P(1 - k). \quad (3.12)$$

3.2.3 k as a Function of e. Let f be the expected fraction of its active life which a particle spends at the source (after birth). This can be expressed as

$$f = \frac{e}{P}. \quad (3.13)$$

The expected fraction of time during which the source is covered is given by the following equation

$$k = f \cdot E, \quad (3.14)$$

that is, the expected amount of time a single particle covers the source multiplied by the expected number of active particles. Substituting for f in (3.14) yields

$$k = \frac{e}{P} E. \quad (3.15)$$

Substituting for E as derived in (3.11) of the previous section yields

$$k = \frac{eN \cdot (1 - k)}{N + P^*(1 - k)}. \quad (3.16)$$

Using the approximation that $N \gg P^*$ yields

$$k \approx e(1 - k). \quad (3.17)$$

Solving for k , one obtains

$$k \approx \frac{e}{1 + e}. \quad (3.18)$$

3.2.4 Obtaining k, E and P for Symmetric Random Walks in Rectangular Two-Dimensional Spaces from the Results of McCrea and Whipple. Long before random walks were ever simulated by computers, McCrea and Whipple (Ref. 14) offered a paper in 1940 discussing two- and three-dimensional random walks and their explicit solutions. Their results have been very fruitful in evaluating the method under discussion here.

Using difference equations as their primary tool, they produced an expression F for "the expectation that (the particle) will visit the point (p, q) before finally emerging at a boundary point," or, in the terminology used in this report, the expected number of time-steps the particle will occupy a point (p, q) before absorption. F is expressed as two functions:

$$F_1(p, q) = \frac{8}{m+1} \sum_{r=1}^m \sin \frac{ar\pi}{m+1} \sin \frac{pr\pi}{m+1} \frac{\sinh(q \cdot Br) \sinh((n+1-b) \cdot Br)}{\sinh(Br) \sinh((n+1)Br)} \quad (3.19a)$$

and

$$F_2(p, q) = \frac{8}{m+1} \sum_{r=1}^m \sin \frac{ar\pi}{m+1} \sin \frac{pr\pi}{m+1} \frac{\sinh((n+1-q) \cdot Br) \sinh(b \cdot Br)}{\sinh(Br) \sinh((n+1)Br)} \quad (3.19b)$$

where p and q are the coordinates of the point at which F is to be evaluated, a and b are the coordinates of the source, and m and n , the dimensions of the rectangular space. F_1 applies whenever $q \leq b$, F_2 when $q \geq b$. An auxiliary relationship is provided for Br :

$$Br = \cosh^{-1} \left(2 - \cos \frac{r\pi}{m+1} \right). \quad (3.20)$$

McCrea and Whipple also state the probability of absorption at any boundary point (x, y) is $1/4$ times the expected number of occupations of the adjacent interior grid point. In (3.19), as n gets large, the various hyperbolic sine computations yield very large numbers resulting in computer overflow. Using the approximation that $e^{-n} \rightarrow 0$ as $n \rightarrow \infty$ yields the following equations for large n :

$$F_1(p, q) = \frac{8}{m+1} \sum_{r=1}^m \sin \frac{ar\pi}{m+1} \sin \frac{pr\pi}{m+1} \left(\frac{e^{(1+q-b)Br} - e^{(1-q-b)Br}}{e^{(2Br-1)}} \right) \quad (3.21a)$$

$$F_2(p, q) = \frac{8}{m+1} \sum_{r=1}^m \sin \frac{ar\pi}{m+1} \sin \frac{pr\pi}{m+1} \left(\frac{e^{(1+b-q)Br} - e^{(1-b-q)Br}}{e^{(2Br-1)}} \right). \quad (3.21b)$$

For the purposes of the experimental work in Chapter V, the equations were modified slightly to account for the differences in labeling systems used to identify the grid points, and the results were used as the basis for two computer programs.

Appendix C contains a listing of the program "PROBS" which computes the theoretically expected absorption probabilities for an $m \times n$ space with source at (a, b) using the relationship given by McCrea and Whipple. This program is based on a simplification of (3.20). The program also takes some advantage of symmetry conditions.

Appendix C also contains a listing of the program "ECOMP" which computes E , P , and k . From (3.18) and the definition of e , it is seen that k is dependent only on e , (i.e., the expected number of occupations of the source (a, b) by particle in the grid). Since the McCrea and Whipple equations include the occupation of (a, b) at birth,

$$k = \frac{F(a, b) - 1}{F(a, b)}. \quad (3.22)$$

The expected path length for any particle is the sum over all grid points of the expected numbers of occupations of those points by that particle. Since the last path segment to an absorbing state is not included in the program "ECOMP," it is not necessary to subtract out the occupation of the source at birth. This yields

$$P = \sum_{r=1}^m \sum_{c=1}^n F(r, c) . \quad (3.23)$$

Given P and k for some rectangular geometry, and using (3.11), then the formula

$$E \approx P(1 - k) \quad (3.24)$$

provides the expected number of particles active in the space. Thus, the use of ECOMP, which employs these results, allows one to evaluate E , k , and P for general rectangular spaces.

CHAPTER IV THEORETICAL ANALYSIS

A measure G having been defined for the relative effectiveness of A_1 compared to A_0 in Section 3.1, this chapter will present theoretical analyses of G for two cases. From the definition of G it is seen that if $G = 1$, then it is equally advisable to use A_1 or A_0 for random walk simulation. If $G > 1$, then using A_1 allows results of any given precision to be obtained in fewer time-steps than would be required by using A_0 .

The first section of this chapter contains an analysis of G for one-dimensional spaces with the source centrally located, and shows that $G \geq 1$ for all cases. The second section contains a proof that, subject to the existence of a particular partition which is intuitively valid though not proven here, $G \geq \frac{1}{2}$ for any space of any dimension, and that $G \geq 1$ for many spaces.

In the third section of this chapter a prediction formula for G , based only on the geometry of the space, is proposed for any single point target set in any two-dimensional space. Since this estimate is based on several approximations, its verification rests on the results of the following chapter. Such a prediction formula allows one to evaluate the performance of A_1 for any problem without requiring simulation.

4.1 Analysis of the One-Dimensional Random Walk with Central Source Location

Consider a random walk along a line containing $2K - 1$ nonabsorbing points with the source located K units from both the left- and right-hand absorbing points, 1 and $2K + 1$, respectively, as in Fig. 1. Let us treat first the symmetric case, i.e., that in which the random variable takes on the values "left" and "right" with equal probabilities, $p_l = p_r = \frac{1}{2}$.

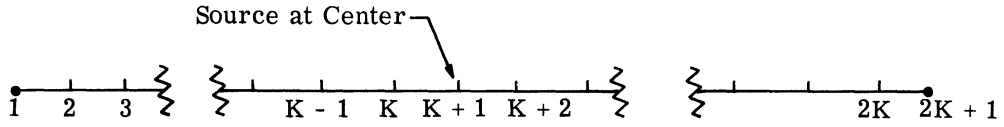


FIGURE 1. ONE-DIMENSIONAL SPACE WITH CENTRAL SOURCE LOCATION

Dickey (Appendix B) treats a one-dimensional simulation according to A_1 as a Markov Chain (Ref. 20) with two states:

- ℓ a particle has been absorbed at 1 and there are K particles extending from 2 to $K + 1$;
and
- r a particle has been absorbed at $2K + 1$ and there are K particles extending from $K + 1$ to $2K$.

This approximation is appropriate for large N since it neglects only the very early and very late time-steps of the simulation. Dickey then shows that the variances of the obtained estimates of the termination probabilities at either absorbing point are each equal to

$$\sigma_1^2 = \frac{1}{4} NK. \quad (4.1)$$

Using the gambler's ruin analogy (Ref. 12), one obtains the corresponding value

$$\sigma_0^2 = \frac{1}{4} N \quad (4.2)$$

for A_0 , yielding

$$D = \frac{\sigma_1^2}{\sigma_0^2} = K. \quad (4.3)$$

Again for large N , there is for any time-step a number of particles as yet unabsorbed, E . From the definitions of the two states ℓ and r there are always K unabsorbed particles in the space. Thus, $E = K$, and for the symmetric one-dimensional case

$$G = \frac{E}{D} = \frac{K}{K} = 1. \quad (4.4)$$

In the asymmetric case, $p \neq q$, Dickey presents a result which reduces to

$$D = \frac{1}{|(p_\ell - p_r)|} \cdot \frac{\left| \binom{K}{p_\ell} - \binom{K}{p_r} \right|}{p_\ell + p_r}. \quad (4.5)$$

As K becomes larger this takes on its maximum value

$$D < \frac{1}{|p_\ell - p_r|}. \quad (4.6)$$

Thus, for any K

$$G = \frac{E}{D} = \frac{K|p_\ell - p_r|(p_\ell^K + p_r^K)}{|p_\ell^K - p_r^K|}, \quad (4.7)$$

and for large K the approximation

$$G \geq K \cdot |p_\ell - p_r| \quad (4.8)$$

can be used.

Thus, for the one-dimensional case, $G = 1$ for $p_\ell = p_r$, and $1 < G \leq K$ for $p_\ell \neq p_r$. G increases with K and grows as $|p_\ell - p_r|$ increases. Figure 2 shows plots of G versus p_ℓ and $1 - p_\ell$ for values of K between 1 and 6.

4.2 A Lower Bound on G for Spaces of Dimension Two and Higher

G having been determined for one-dimensional spaces with the source in the center, this section carries the discussion to spaces of higher dimension. A proof establishing a lower bound on G is followed by discussion of this proof and the conditions assuring its validity.

4.2.1 A Lower Bound on G . An analytic result establishing that $G \geq 1$ for any space is presented here and is subject to the following two constraints:

1. Given a random walk simulation according to A_1 , in an intuitive sense it is true that no more than an expected number $E - 1$ of previously emitted particles can be correlated with any newly emitted particle. For purposes of the following proof, this condition becomes: for all N there exists a partition such that those previously emitted particles correlated with the N^{th} particle can be distinguished from those uncorrelated with the N^{th} , and that there is an expected number $E-1$ of particles in the former set.

2. Given X_i , the random variable which has the value 1 when particle i is absorbed in the target set and 0 otherwise, and that $P(X_i = 1) = p$, then a strong result is obtained if it can be shown that $P(X_i = 1 | X_j = 1) \leq \frac{1}{2}$. Although this may not in general be true, it is shown in discussion following the proof that for at least one case of practical interest this inequality is valid. For cases where this is not true, it is proven that $G \geq \frac{1}{2}$.

Theorem

Given that in a simulation according to A_1 a partition exists such that particles in the set S are uncorrelated with the N^{th} particle for any N and there are $E-1$ particles in \bar{S} , then for any set of target points T and for any geometry, $G \geq \frac{1}{2}$; and when $P(X_i = 1 | X_j = 1) \leq \frac{1}{2}$, $G \geq 1$.

Proof

Consider an N -trial simulation using algorithm A_1 . It is clear that

$$\sigma_1^2(X_1 + \dots + X_N) = \sum_{i=1}^N \sigma_1^2(X_i) + 2 \sum_{i=1}^N \sum_{j=1}^{i-1} \text{cov}(X_i, X_j). \quad (4.9)$$

Since the second term has the value 0 for $i = 1$, and from the definition of the sum of covariances, it follows that

$$\sigma_1^2(X_1 + \dots + X_N) = Npq + 2 \sum_{i=2}^N \text{cov}(X_1 + \dots + X_{i-1}, X_i). \quad (4.10)$$

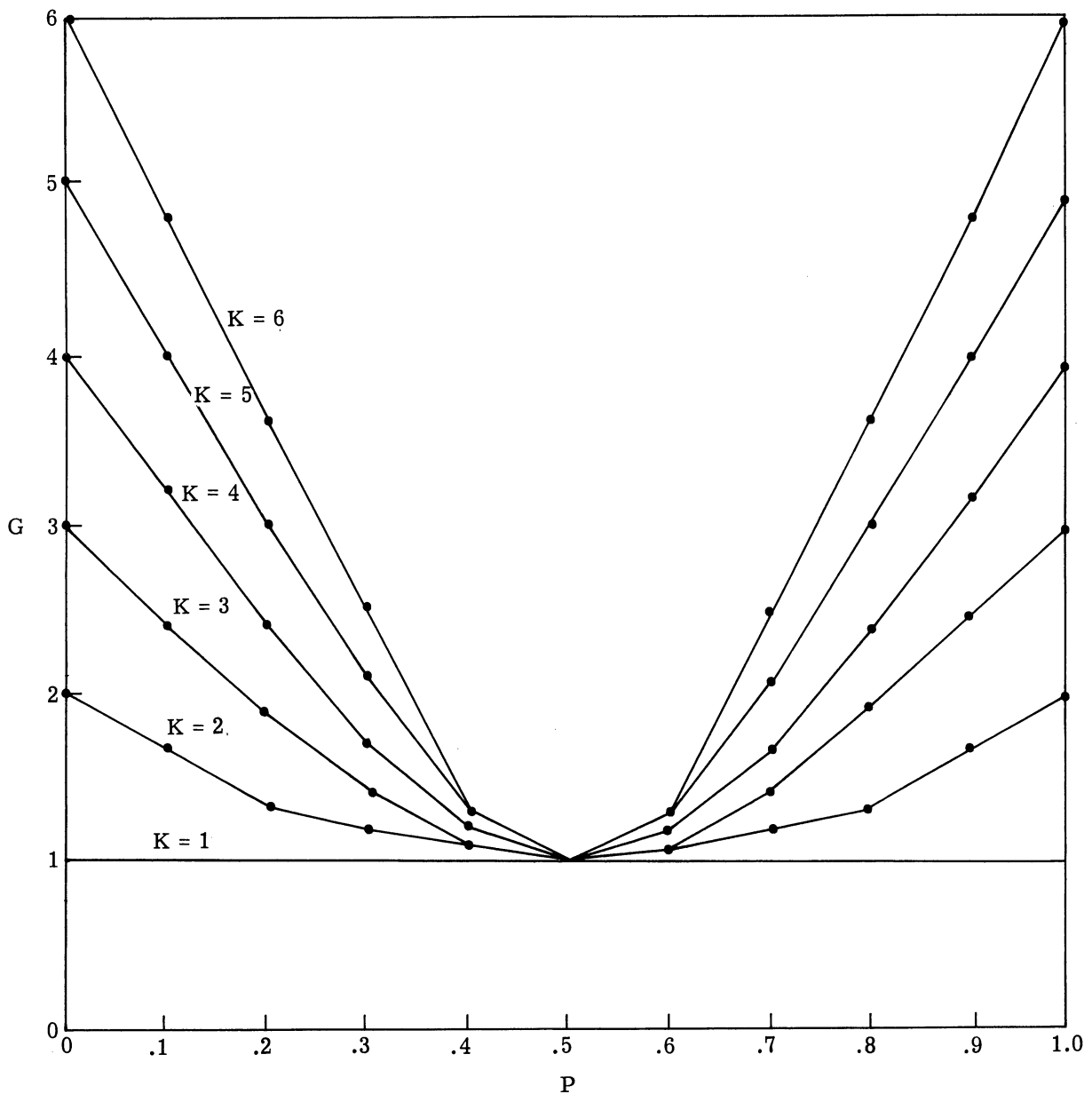


FIGURE 2. G AS A FUNCTION OF p FOR VALUES OF K = 1, 2, ..., 6.

From the partition in the statement of the theorem one obtains

$$\text{cov}(X_1 + \dots + X_{N-1}, X_N) = \sum_{i \in \bar{S}} \text{cov}(X_i, X_N) + \sum_{i \in S} \text{cov}(X_i, X_N). \quad (4.11)$$

According to the definition of the partition, the first sum has the value 0 since the random variables $X_i |_{i \in \bar{S}}$ are uncorrelated with X_N .

To evaluate the second sum, consider that according to the definition of the partition there are $E-1$ particles in \bar{S} for $N > E$ and fewer than $E-1$ particles in \bar{S} for $N \leq E$. Thus, the second sum consists of at most $E-1$ terms of the form $pq \rho_{X_i X_N}$. Let

$$\hat{\rho} = \text{Max}_i \rho_{X_i X_N}. \quad (4.12)$$

Then, from the preceding argument, for all N

$$\text{cov}(X_1 + \dots + X_{N-1}, X_N) \leq (E-1) pq \hat{\rho}. \quad (4.13)$$

From Eqs. (4.10) and (4.13) it follows that

$$\sigma_1^2(X_1 + \dots + X_N) \leq Npq + 2(N-1)(E-1) pq \hat{\rho}. \quad (4.14)$$

Since $\hat{\rho} \leq 1$, for any geometry

$$\sigma_1^2(X_1 + \dots + X_N) \leq Npq + 2(N-1)(E-1) pq, \quad (4.15)$$

hence

$$D = \frac{\sigma_1^2}{\sigma_0^2} \leq \frac{Npq(2E-1)}{Npq} \leq 2E. \quad (4.16)$$

Thus,

$$G = \frac{E}{D} \geq \frac{E}{2E}, \quad (4.17)$$

hence

$$G \geq 1/2. \quad (4.18)$$

However, when $P(X_i = 1 | X_j = 1) \leq 1/2$, a stronger result can be obtained as follows:

By definition

$$\text{cov}(X_i, X_j) = \sigma_{X_i} \sigma_{X_j} \rho_{X_i X_j} = \text{Ex}[(X_i - p)(X_j - p)], \quad (4.19)$$

where \bar{X}_k is the expected value of X_k and is equal to p for all k . It follows that

$$pq\rho_{X_i X_j} = \text{Ex}(X_i X_j - pX_j - pX_i + p^2) \quad (4.20)$$

$$= p \cdot P(X_i = 1 | X_j = 1) - p^2. \quad (4.21)$$

Applying the above constraint one obtains

$$\rho_{X_i X_j} \leq \frac{p(1/2 - p)}{p(1 - p)} = \frac{1 - 2p}{2 - 2p} \leq 1/2. \quad (4.22)$$

Thus, for this case

$$\hat{\rho} \leq \frac{1}{2}. \quad (4.23)$$

Applying this in the earlier result, one obtains

$$\sigma_1^2(X_1 + \dots + X_N) \leq Npq + 2(N-1)(E-1)pq/2 \leq NEpq \quad (4.24)$$

Therefore,

$$D = \frac{\sigma_1^2}{\sigma_1} \leq \frac{NEpq}{Npq} = E, \quad (4.25)$$

and

$$G = \frac{E}{D} \geq \frac{E}{E}, \quad (4.26)$$

$$G \geq 1$$

for any space which obeys the conditions outlined above.

QED

Corollary

If, for the new algorithm,

$$\sigma_1^2(X_1 + \dots + X_N) \leq NEpq, \quad (4.27)$$

then $G \geq 1$.

Proof

The proof follows immediately from Eq. (4.24) in the previous proof.

In the theorem the value chosen for $\hat{\rho}$ in Eq. (4.12) is that of the maximum of all particle correlations. In the same intuitive sense that the partition exists, so does each of the $E - 1$

terms $\rho_{X_i X_N}$ (reflecting the dependence of particle N on particle i) actually take on value from $\hat{\rho}$ to some value much less than $\hat{\rho}$ depending on the portions of their path histories which overlap. If one can show that the average of the $\rho_{X_i X_N}$ for any N is less than or equal to $1/2$, then, by the following corollary, $G \geq 1$.

Corollary

If the partition defined previously exists and

$$\bar{\rho} = \frac{\sum_{i \in S} \rho_{X_i X_N}}{E - 1} < \frac{1}{2}, \quad (4.28)$$

then $G \geq 1$.

Proof

This follows from substituting $\bar{\rho} = \frac{1}{2}$ for $\hat{\rho}$ in (4.14), yielding

$$\sigma_1^2(X_1 + \dots + X_N) \leq N pq + (N-1)(E-1) pq. \quad (4.29)$$

Clearly, now

$$\sigma_1^2(X_1 + \dots + X_N) \leq NE pq \quad (4.30)$$

which satisfies the condition of the previous corollary.

4.2.2 Comments on the Theorem of 4.2.1. The condition that $P(X_i = 1 | X_j = 1) < \frac{1}{2}$ stipulated for the proof that $G \geq 1$ can be shown to exist for many cases. For example consider symmetric random walks in two-dimensional spaces where the set T consists of a single point on a convex curve such that the boundary lies inside the curve. Figure 3 illustrates such a case. In such a case, one grid line on which this point lies may be continued to infinity in either direction without partitioning the space.

Let I_s be the event that particle i is at any arbitrary grid point s when particle j has been absorbed at T. Then

$$P(X_i = 1 | X_j = 1) = \sum_s P(X_i = 1 | X_j = 1 \text{ and } I_s) \cdot P(I_s) \quad (4.31)$$

and clearly

$$P(X_i = 1 | X_j = 1) \leq \text{Max}_s P(X_i = 1 | X_j = 1 \text{ and } I_s). \quad (4.32)$$

After particle j is absorbed at T, the remainder of the path of particle i is independent of the path of particle i except for the fact that it starts at s. Therefore

$$P(X_i = 1 | X_j = 1) < \text{Max}_s P(X_i = 1 | I_s). \quad (4.33)$$

This maximum is attained when s is W , the grid point adjacent to T , yielding

$$P(X_i = 1 | X_j = 1) \leq P(X_i = 1 | I_w). \quad (4.34)$$

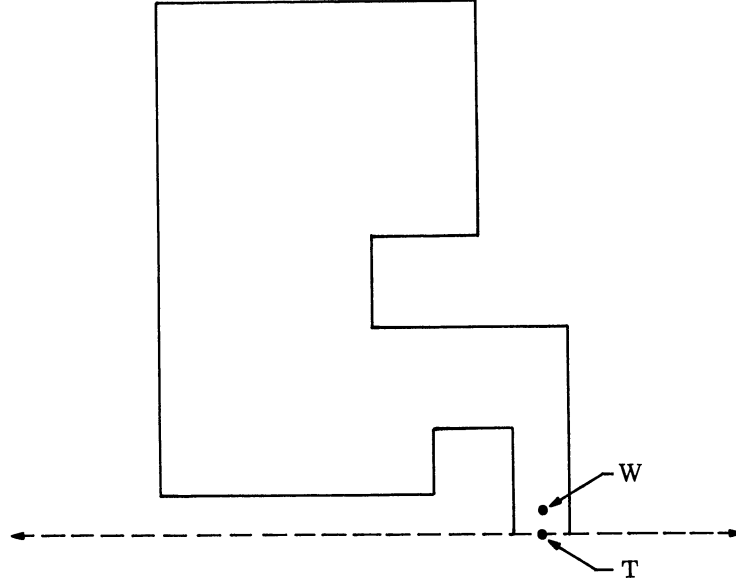


FIGURE 3. EXAMPLE OF A BOUNDARY POINT FOR WHICH THE PROBABILITY OF PARTICLE ABSORPTION IS $\leq 1/2$

For T as defined in the first paragraph, the upper limit on $P(X_i = 1)$ for a random walk initiated at W occurs when the absorbing boundary is the infinite line defined earlier and the space is the resulting infinite half-plane. From a result of McCrea and Whipple (14), in this case

$$P(X_i = 1 | X_j = 1) \leq P(X_i = 1 | I_w) < .37 < \frac{1}{2}, \quad (4.35)$$

which is the necessary condition.

Offering strong support for the validity and generality of the hypothesis that $G \geq 1$, subject to no constraints whatsoever, is the following argument.

Consider the conventional random walk simulation of N independent particles in a period of time t . Consider further the addition of N more particles whose path histories are determined by the same values of the random variable which direct the motion of the original particles and whose life times all occur within the period t . The expected absorption probabilities of these

particles around the boundary are the same as the first N particles by the argument presented in the second chapter.

Let us define the absorption of an independent particle to yield an expected I bits of information. Then N particles will yield a total of NI bits of information in t units of time. If each of the additional particles is totally dependent upon one of the original particles, then its termination yields no information. Thus, in t units of time the information yield is exactly the same as in the original N particle simulation; $G = 1$. If, however, only a portion of the history of every added particle is dependent on any one original particle, then they are each only partially correlated with any one particle and are partially independent in some sense—yielding some information upon absorption. Thus, $G \geq 1$ for this system of particles.

4.3 An Estimate of G Based on Geometry of Two-Dimensional Spaces

The results of the previous section, while offering support for the hypothesis that $G \geq 1$ for any geometry, do not offer any means for estimating how much more efficient A_1 might be than A_0 for a particular space. The following proposed estimate for G is based on a target set of one point in a two-dimensional space.

As defined earlier,

$$G = E/D. \quad (4.36)$$

An estimate for E is available for some cases (Section 3.2.2), but since it is not generally applicable to all two-dimensional spaces, a desire for a more general result has led to the development of a direct method for predicting the value of this parameter for any two-dimensional geometry. This will be presented below.

Obtaining an estimate for D is also necessary to estimate G . As a measure of the imprecision of A_1 , D is related to the number of particles active in the space E . This follows from the following facts.

1. When $E = 1$, $D = 1$; i.e., in a conventional random walk there is no imprecision resulting from the correlation of any particle with any other.
2. When $E > 1$, then $D > 1$. Since there is more than one particle in the space at one time, they will be correlated with one another.

However, it is hypothesized that D will not increase as rapidly as E since the average physical proximity of particles, hence the probability of their absorption in the same target set, will decrease as E increases. Based on data from experiments performed (see Section 5.3.3), it is hypothesized that

$$D \sim E^{\frac{1}{2}} \quad (4.37)$$

yielding

$$G \sim E^{\frac{1}{2}}. \quad (4.38)$$

This hypothesis is later supported using the experimental results of Chapter V which also presents a related technique for estimating a lower bound on G.

To obtain a simple estimate for the parameter E, the following method is offered. E is related to the amount of non-absorbing space in the geometry and also to the relationship of the location of the source point to the absorbing boundary. Consider a partition of the space into four subspaces by extending horizontal and vertical lines through the source (see Fig. 4). Since particles do not tend to reach isolated parts of the space, it is hypothesized that E_e , an estimate for E_1 , is proportional to the geometric mean of the areas a_i of the four largest rectangles each containing the source and each lying entirely within one of the four subspaces:

$$E_e \approx (a_1 \cdot a_2 \cdot a_3 \cdot a_4)^{\frac{1}{4}}. \quad (4.39)$$

This hypothesis is supported by experimental data in Chapter V, thus suggesting its use in (4.38), resulting in the relationship

$$G \approx (a_1 \cdot a_2 \cdot a_3 \cdot a_4)^{\frac{1}{8}}. \quad (4.40)$$

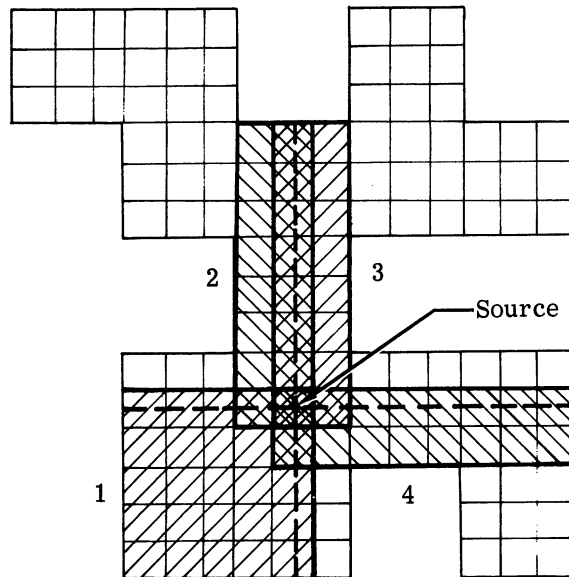


FIGURE 4. PARTITION OF A SPACE FOR PURPOSE OF ESTIMATING G. (The numbers identify the four shaded subspaces.)

4.4 Summary of Theoretical Results

The research reported shows that for a symmetric random walk according to A_1 in a one-dimensional space with centrally located source, $G = 1$. For the cases where the random variable governing particle motion is asymmetrically distributed $G > 1$, and in this case, the value of G increases as the number of non-absorbing points on the line is increased.

It has also been shown that for any space, subject to the existence of an intuitively plausible partition, $G \geq \frac{1}{2}$. Furthermore, given that X_i is the random variable associated with the absorption of particle i in the target set, then if $P(X_i = 1 | X_j = 1) \leq \frac{1}{2}$, $G \geq 1$; or if the average correlation of one particle with all those previously emitted but unabsorbed can be shown to be $\leq \frac{1}{2}$, then $G \geq 1$. On the basis of this work, it is conjectured that $G \geq 1$ for all spaces. The next chapter describes a set of experiments designed to support this hypothesis for symmetric random walks and to demonstrate the validity of the prediction method of the preceding section.

CHAPTER V

EXPERIMENTAL ANALYSIS OF TWO-DIMENSIONAL SPACES

Having obtained a lower bound on G which strongly suggested that $G \geq 1$ for all cases, it was decided to determine G for two-dimensional spaces by simulating symmetric random walks using algorithm A_1 , and in this way to obtain both estimates of G and the precision of these estimates. Analysis of a single space would yield results which were applicable to only that case; however, a series of experiments would allow inference leading to more general conclusions.

Two particular objectives were set forth, each leading to a separate set of experiments. The first was to determine the behavior of G as a function of the size of the space only, and to this end experiments were performed on various sized square spaces with the source in the center. This work is reported in Section 5.1. The second objective was to establish that A_1 can be used to advantage in a space consisting of a simply connected but irregular boundary. To this end experiments were performed on a particular subset of spaces chosen at random from all spaces which can be contained in a 17 by 17 envelope. The random selection assures no bias is acting when inference is drawn to all spaces. These experiments are reported in Section 5.2. Since G is a function of the particular set of points for which the absorption probability is desired (called the target set), analyses were carried out for target sets of various kinds.

It was also important to estimate the variance of the value of each G obtained so that a level of confidence could be determined for any experimental G as an estimate of the true G

for that case. This was done by obtaining the variance of all values of G which represents an average across target sets. Since G is made up, in part, of D which is itself a variance, the variance of G is a variance of a variance.

The results of these experiments are summarized and discussed in Section 5.3.

The particular computer programs employed in this work are included in Appendix C. Section 3.2 and Appendix D contain discussions of their contents and what might be considered novel techniques employed.

5.1 Experiments on L by L Square Spaces with Central Source Location

5.1.1 Experimental Design. In order to estimate

$$G = E/D \tag{5.1}$$

for some target set, it is necessary to obtain values for the two factors E and D which constitute it. As discussed earlier in Section 3.2, it is possible to obtain the expected values E, D and k for a rectangular space using the results of McCrea and Whipple (Ref. 14). To this end a computer program ECOMP has been developed.

To obtain values for

$$D = \frac{\sigma_1^2}{\sigma_0^2} \tag{5.2}$$

for some target set, one needs to evaluate its components σ_0^2 and σ_1^2 . In Section 3.2.4 it is also noted that the true absorption probability for any boundary point in any rectangular space can be obtained without simulation, and the program PROBS has been written to do this. Given p, the probability of absorption in a particular target set and using Eq. 3.1,

$$\sigma_0^2 = Npq,$$

(where N is the number of particles simulated and $q = 1 - p$) one can then compute the denominator of D.

This leaves only σ_1^2 to be determined in order to estimate G. By generating some number R of replications of N-trial simulations of random walks in L by L square spaces, σ_1^2 can be calculated directly from the number of terminations in the target set chosen. The program NDEP and its associated subroutines were used to obtain the data necessary in the form of the number of absorptions x_i at point i. The probability p_i is then estimated by x_i/N .

Results for target sets of more than one point are obtained by combining results for individual points.

Figure 5 outlines the flow of information from the specification of the results desired to obtain an estimate for G for some target set. The program COVAN is constructed to accept the specification of target sets, and inputs from NDEP and PROBS on punched cards. This results in estimates of D for the specified target sets. A desk calculator is then used to compute G from the results of COVAN and ECOMP. Separation of NDEP and COVAN allows the computation of G based on an alternate target set without repeating a simulation by maintaining the simulation results on punched cards.

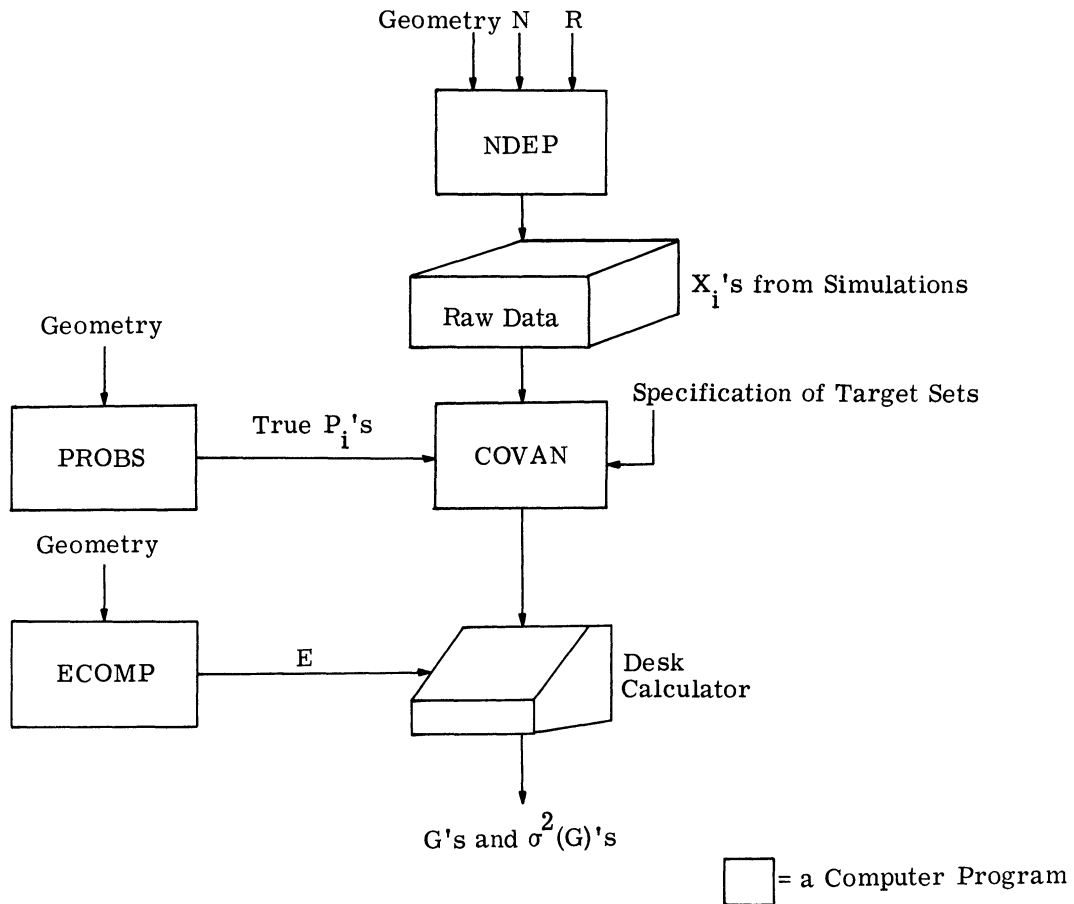


FIGURE 5. FLOW OF INFORMATION IN THE EXPERIMENTS ON SQUARE SPACES

Taking advantage of the fact that a square with the source at the center has four axes of symmetry, one can divide the points on the boundary of an L by L square (including absorbing points) into $L^* = \frac{L-1}{2}$ symmetry classes. Let c_1 contain those points closest to the corners, c_2 the next closest, . . . , and c_{L^*} those points closest to the center of the sides. Then $n(c_1)$,

the number of points in each class will be 8, except that $n(c_{L^*}) = 4$ because the requirement that the source be at the center demands L be odd. This provides a built-in set of replications since the absorption probabilities are the same for all points in any equivalence class.

It was decided that a single measure for D should be defined which in some sense would represent every individual point on the boundary. For this purpose let us define

$$D_{c_i} = \frac{\sum_{j \in c_i} D_j}{n(c_i)} \quad (5.3)$$

which is the average of the D_j for one symmetry class, and

$$\bar{D} = \frac{\sum_{i=1}^{L^*} D_{c_i}}{L^*} \quad (5.4)$$

the single measure to represent all of the D_j .

The following derivation supplied the estimate of the variance of G as based on E and \bar{D} . By expanding E/D about E/\bar{D} and retaining terms through fourth order one obtains

$$\sigma^2(G) = \frac{E^2}{\bar{D}^4} \left[\frac{3[\sigma^2(D)]^2}{\bar{D}^2} + \sigma^2(D) + \text{Ex} \left[\frac{(D - \bar{D})^4}{\bar{D}^2} \right] - 2 \text{Ex} \left[\frac{(D - \bar{D})^3}{\bar{D}} \right] \right] \quad (5.5)$$

where Ex indicates expected value. Assuming that the variance of D_j is the same for every point i , one can use the following approximation of the variance of \bar{D}

$$\hat{\sigma}^2(\bar{D}) = \frac{\sum_{i=1}^{L^*} \left[\sum_{j \in c_i} (D_j - D_{c_i})^2 \cdot [n(c_i) - 1] \right]}{\sum_{i=1}^{L^*} [n(c_i) - 1]} \quad (5.6)$$

An estimate of $\sigma^2(G)$ called $\hat{\sigma}^2(G)$ was calculated for the results obtained by substituting $\hat{\sigma}^2(\bar{D})$ for $\sigma^2(D)$ in Eq. (5.5).

5.1.2 Preliminary Experiments. Based on the estimate $\hat{\sigma}^2(G)$ and the desire to present results for as large a number of spaces as possible consistent with the use of a reasonable amount of computer time, it was decided to choose the number of replications R large enough to allow $\hat{\sigma}^2(G) \leq 0.2 G$. The number of random walks run in one simulation need only be large enough to

overshadow the earliest and latest periods in which there are fewer than E particles in the grid.

To choose standard values for N and R, random walks were simulated in a 13 by 13 square for various combinations of these values. The data for \bar{D} and $\hat{\sigma}^2(\bar{D})$ for these trials are presented in Tables Ia and Ib, respectively. From the data of Table Ia, \bar{D} was assumed to be independent of N for $N \geq 1500$. The dependence for $N < 1500$ can be explained by the greater importance of the early and late periods in the simulation when these account for a large part of the total simulation time. From Table Ib it was assumed that $\hat{\sigma}^2(\bar{D})$ is approximately inversely proportional to the square root of R. By extrapolating these data and based on Eqs. (5.5) and (5.6) it was determined that $R \geq 200$ to meet the criterion set forth above for $\hat{\sigma}^2(G)$. Thus the values $N = 1500$ and $R = 200$ were chosen for this work.

TABLE Ia. \bar{D} as a Function of N and R for a 13 by 13 Square Space

R	N (in hundreds)								
	1	5	10	15	25	50	100	250	Ave.
10	2.20	2.79	2.31	2.51	2.53	2.43	2.83	2.51	2.51
15	2.76	2.60	2.27	2.24	2.34	1.90	2.76	2.22	2.39
25	2.42	2.62	2.05	2.58	2.98	2.78	2.54	2.69	2.58
35	2.46	2.67	2.14	2.58	2.81	2.67	2.66	2.63	2.60
50	2.53	2.66	2.18	2.45	2.68	2.43	2.67	2.49	2.51
Ave.	2.48	2.67	2.19	2.47	2.66	2.44	2.69	2.56	2.52

TABLE Ib. $\hat{\sigma}^2(\bar{D})$ as a Function of N and R for a 13 by 13 Square Space

R	N (in hundreds)								
	1	5	10	15	25	50	100	250	
10	2.33	1.59	0.838	1.61	1.52	0.924	2.25	0.840	
15	1.99	1.16	0.382	0.618	0.514	0.472	0.846	0.256	
25	0.567	0.590	0.470	0.610	1.02	1.16	0.349	0.380	
35	0.276	0.295	0.313	0.537	0.417	0.574	0.360	0.256	
50	0.285	0.163	0.201	0.287	0.337	0.266	0.288	0.220	

Table II shows the results of using the program ECOMP to obtain E, and also the expected source covering factor k and the expected path length D, each as a function of L. These data are shown plotted in Fig. 6. By plotting $1/1-k$ versus $\log_{10} L$ as in Fig. 7 it is shown that, although k approaches 1 as L becomes large, indicating an eventual inability of the source to emit new particles, for all cases of practical use ($L < 10^6$) $k < .9$.

TABLE II. k , P , and E as a Function of L

L	k	P	E
3	0.0	1.0	1.0
7	0.435	10.38	5.87
9	0.489	18.63	9.53
11	0.524	29.24	13.93
13	0.549	42.20	19.04
15	0.568	57.53	24.85
25	0.624	169.51	63.75
51	0.680	736.5	235.5
75	0.704	1613.4	477.9
101	0.720	2946.6	825.8

5.1.3 Experimental Results. Experiments were performed for L equal to 7, 9, 11, 13, 15 and 25. Table III presents

E from Table II

$$\bar{D}$$

$$G = E/\bar{D}$$

D_{\max} , the maximum of the D_i

$$G_{\min} = E/D_{\max}$$

$$\hat{\sigma}^2(\bar{D})$$

$$\hat{\sigma}^2(G) \text{ based on Eq. 5.5 and } \hat{\sigma}^2(\bar{D}).$$

These results, except for the last two items, are shown plotted versus L in Fig. 4. These data represent G for a target set consisting of a single point.

TABLE III. G and G_{\min} as a Function of L

L	E	\bar{D}	G	D_{\max}	G_{\min}	$\hat{\sigma}^2(\bar{D})$	$\hat{\sigma}^2(G)$
3	1.0	1.0	1.0	1.0	1.0		
7	5.87	1.67	3.51	1.77	3.31	0.031	0.172
9	9.53	1.90	5.02	1.95	4.89	0.024	0.217
11	13.93	2.14	6.52	2.19	6.37	0.026	0.312
13	19.04	2.45	7.77	2.65	7.19	0.057	0.785
15	24.85	2.68	9.28	2.84	8.75	0.061	1.02
25	63.75	4.14	15.41	4.70	13.55	0.170	2.91

Two new target sets were defined, one consisting of the points along two adjacent sides (one-half the total points) and the other consisting of two adjacent half sides including the center points (see Figs. 9a and b). There are four sets of each type for any square. Table IV contains data for each of these averaged over the four symmetry replications; $D_{1/2}$ for the first target set and $D_{1/4}$ for the smaller, second one. $G_{1/2}$ and $G_{1/4}$ were computed from $D_{1/2}$ and $D_{1/4}$ and E .

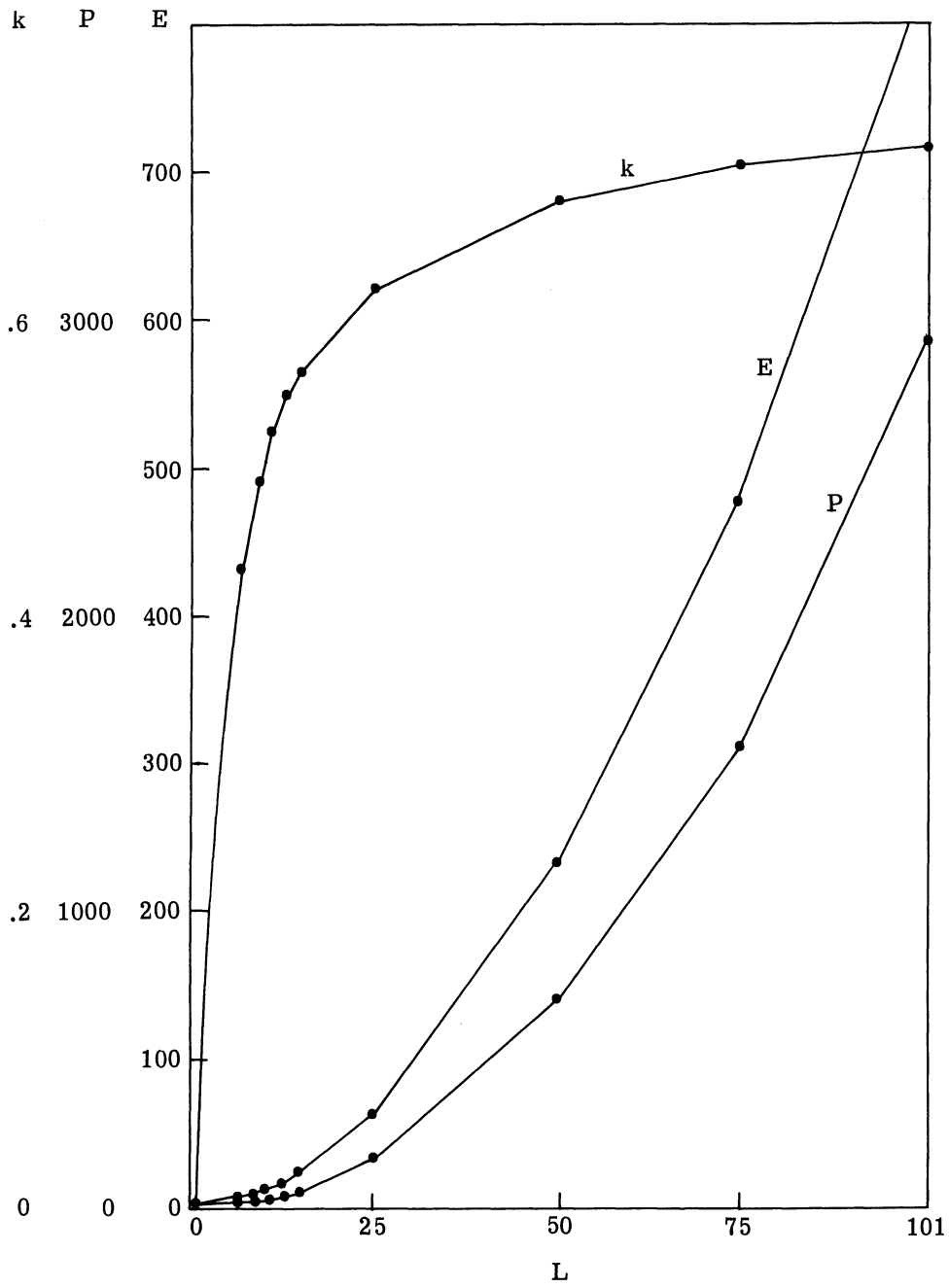


FIGURE 6. k, E AND P AS A FUNCTION OF L

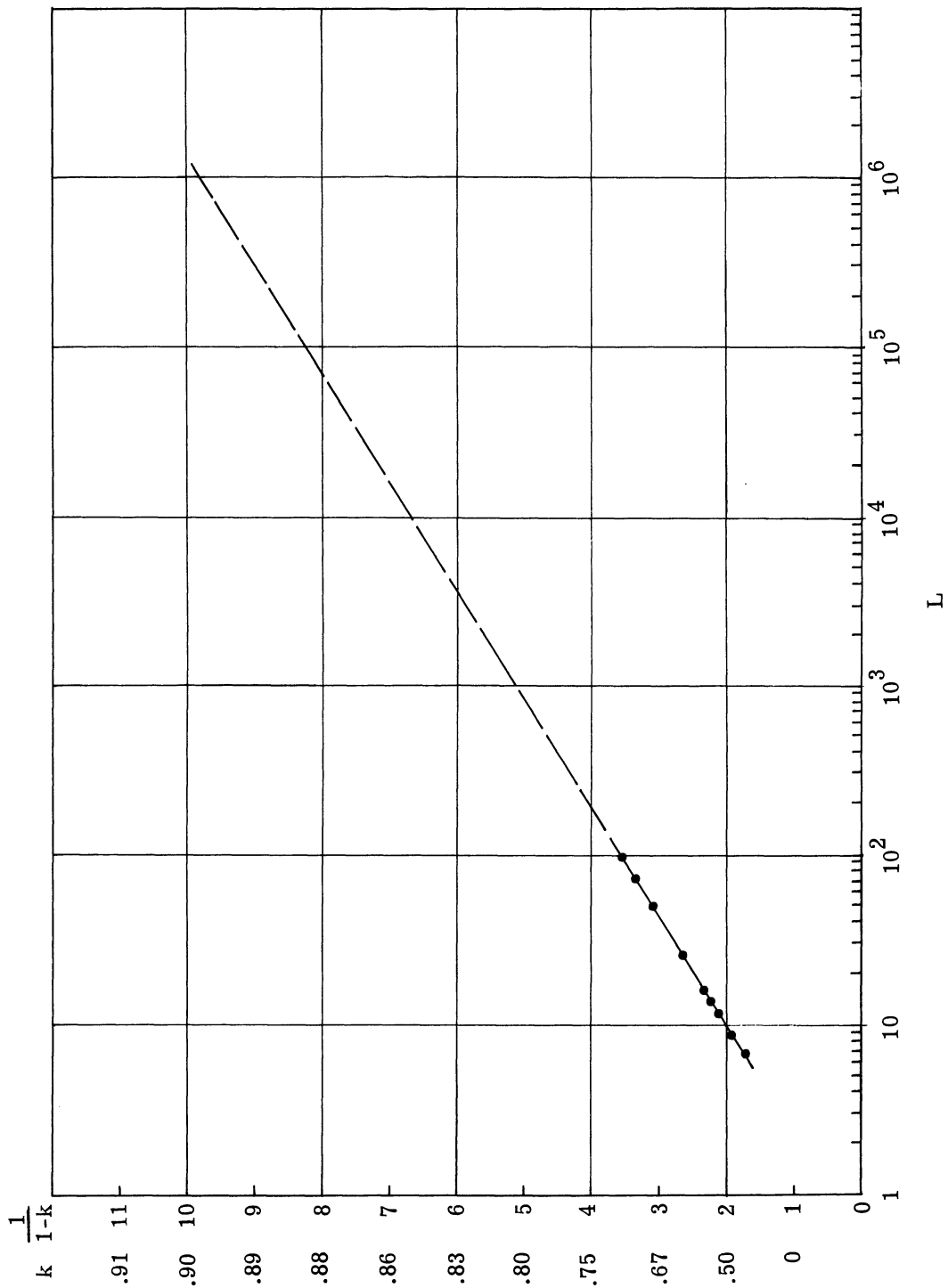


FIGURE 7. $\frac{1}{1-k}$ AS A FUNCTION OF L . (L is scaled logarithmically and corresponding values of k are shown. Dashed line indicates extrapolation.)

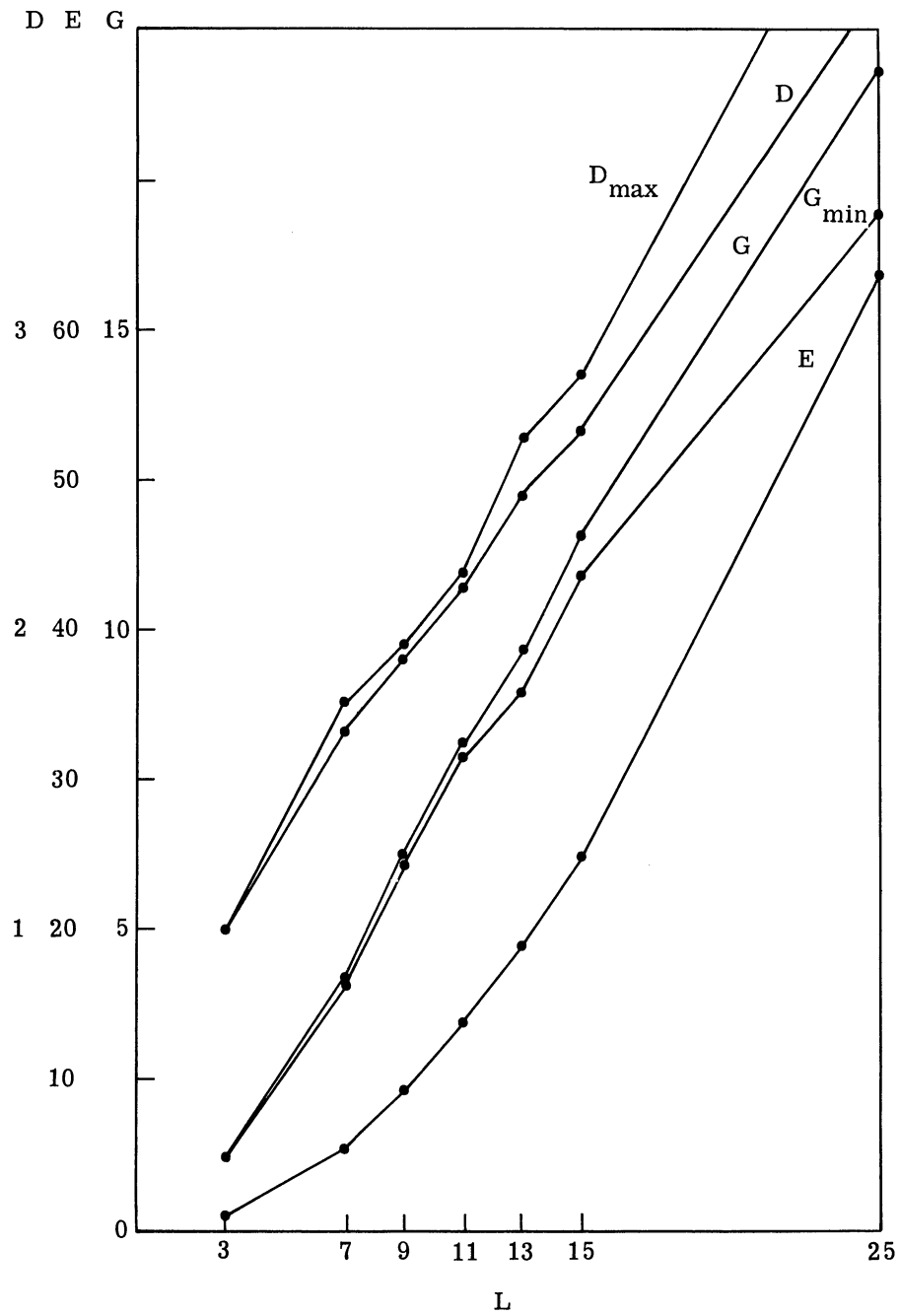


FIGURE 8. RESULTS FOR SINGLE POINT TARGET SETS IN SQUARE SPACES

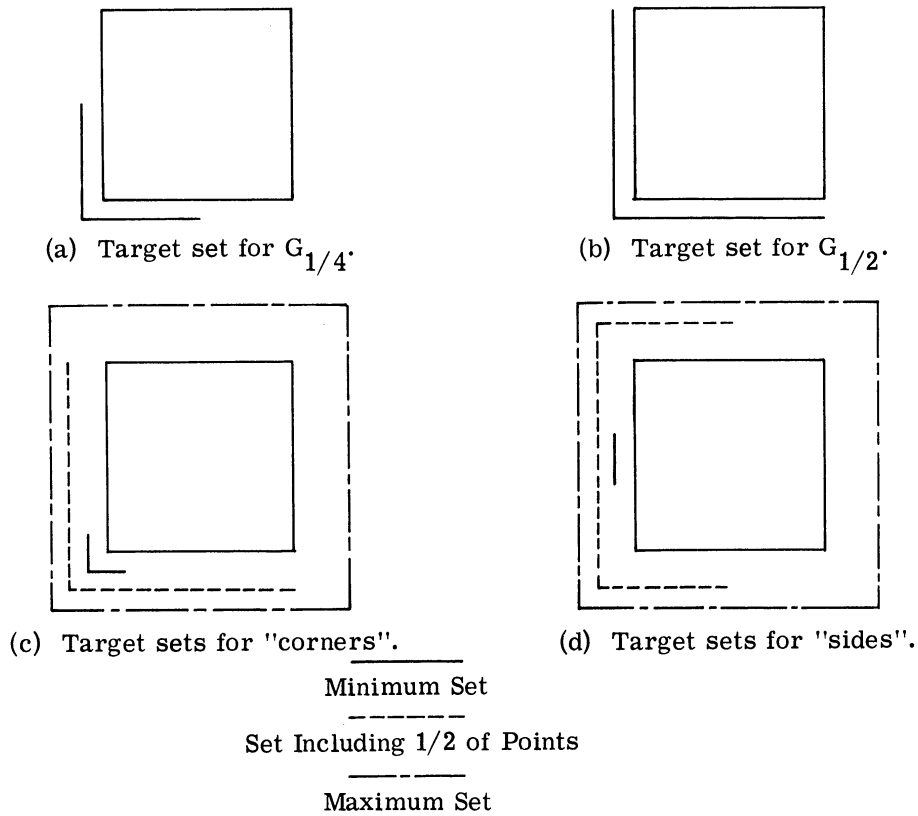


FIGURE 9. VARIOUS TARGET SETS USED

TABLE IV. $G_{1/4}$ and $G_{1/2}$ as a Function of L

L	E	$D_{1/4}$	$G_{1/4}$	$D_{1/2}$	$G_{1/2}$
7	5.87	4.05	1.45	4.73	1.24
9	9.53	6.29	1.51	7.57	1.26
11	13.93	8.74	1.60	10.91	1.28
13	19.04	12.70	1.50	16.07	1.19
15	24.85	16.62	1.50	19.36	1.28
25	63.75	44.62	1.43	56.48	1.13
Average			1.50		1.27

For the 13 by 13 square several target sets were defined in order to determine how G is related to the number of points in the set and their relative proximity. The sets were specified as either "corner" sets or "side" sets; the former consisting of an even number of points half on either side of a corner, and the latter of an odd number of points including one in the center of a side [see Figs. 9(c) and (d)]. Two further distinctions were made; "closed" sets consisted of contiguous points and "alternate" sets consisted of points separated from one another by a single gap. Table V and Fig. 10 present G for these sets with the number of points included in any set (including gaps) varying from 1 to 27 for "sides" data and from 2 to 28 for "corners" data.

TABLE V. G as a Function of the Number of Points in "Corners" and "Sides" Target Sets

Number of Points in Set	Sides		Corners	
	Closed	Open	Closed	Open
1	5.12			
2			3.44	
3	2.00	3.29		
4			2.13	
5	1.48	13.04		
6			1.72	3.04
7	1.33	2.84		
8			1.52	
9	1.31	2.96		
10			1.37	3.13
11	1.34	3.29		
12			1.28	
13	1.37	3.64		
14			1.26	3.34
15	1.37	4.35		
16			1.28	
17	1.34	4.82		
18			1.37	4.51
19	1.31	5.61		
20			1.52	
21	1.33	5.88		
22			1.72	10.01
23	1.48	7.82		
24			2.13	
25	2.00	14.9		
26			3.44	22.6
27	5.12	33.9		
28				

5.2 Experiments on Spaces with Irregular Boundaries

5.2.1 Experimental Design. Since the spaces used in this part of the research were to be representative of all possible spaces of a certain size, they had to be selected at random from the total population. In this way the results could be used in drawing inference to the entire population of spaces without fear that some bias used in selecting the spaces would affect the results. The selection of spaces is discussed in detail in Section 5.2.2.

Since the spaces studied in this investigation were not rectangles, the parameter E and the true absorption probabilities p which were obtained theoretically in the previous experiments had to be determined experimentally. The values used for the p's were the estimates x/N , where x is the number of particles absorbed in the selected target sets in an N-trial simulation. In each replication it was necessary to measure E directly and to include it as output along with the data that were obtained in the previous experiments. The covering factor k was recorded in a similar manner.

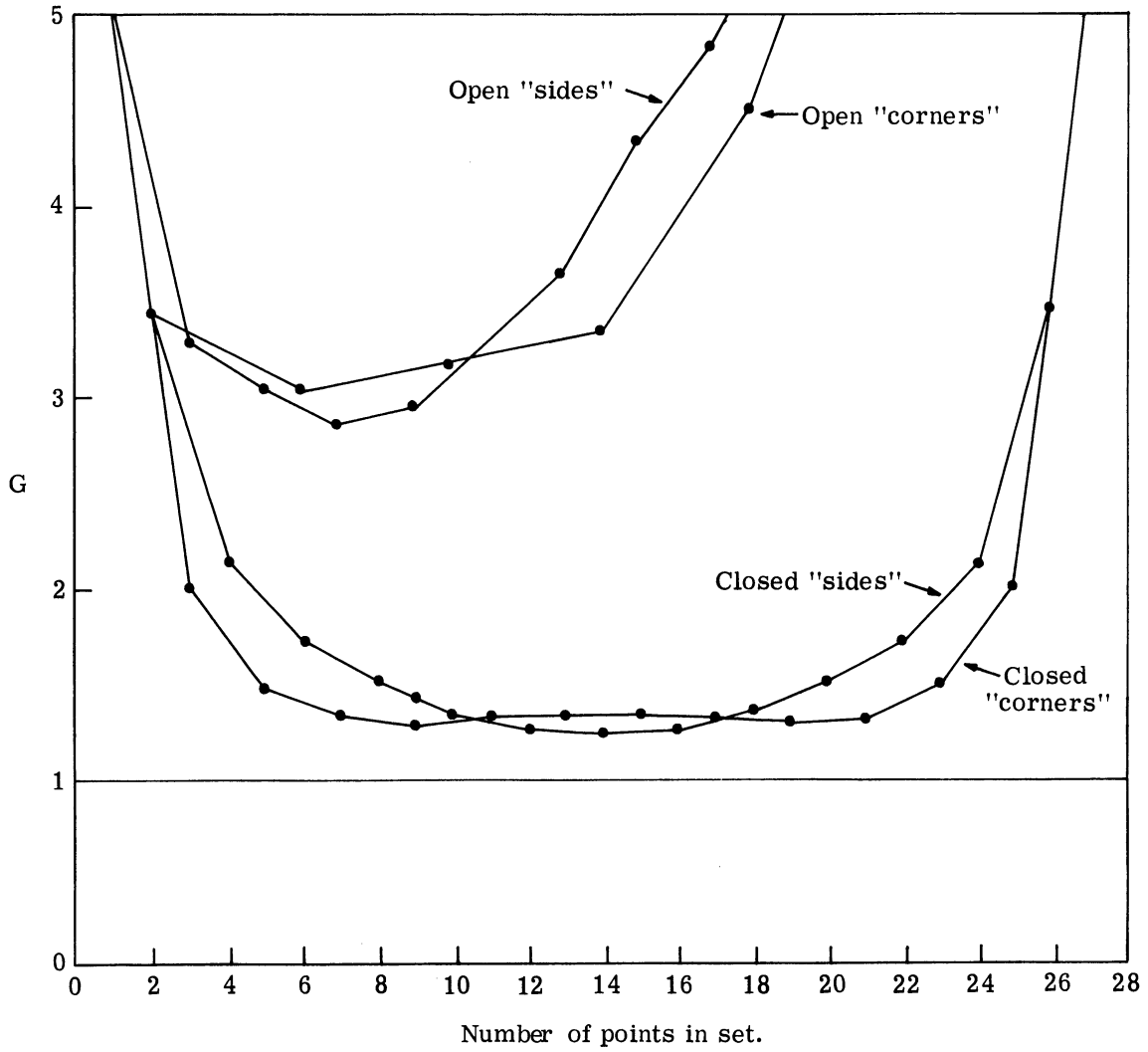


FIGURE 10. G AS A FUNCTION OF THE NUMBER OF POINTS IN "CORNERS" AND "SIDES" TARGET SETS

In order to obtain $\sigma^2(G)$ it was decided to divide the total set of replications, each of N simulations, into R sets of r replications each. As is indicated in Fig. 11, a value of G can be obtained for each of the $R \times r$ replications by using the set of equations

$$\sigma_0^2 = N \left(\frac{x}{N} \right) \left(1 - \frac{x}{N} \right), \quad (5.7)$$

$$\sigma_1^2 = \sigma^2(x), \quad (5.8)$$

$$D = \frac{\sigma_1^2}{\sigma_0^2} = \frac{\sigma^2(x) \cdot N}{x(N-x)} \quad (5.9)$$

and

$$G = \frac{E}{D} = \frac{E x(N-x)}{\sigma^2(x) \cdot N}. \quad (5.10)$$

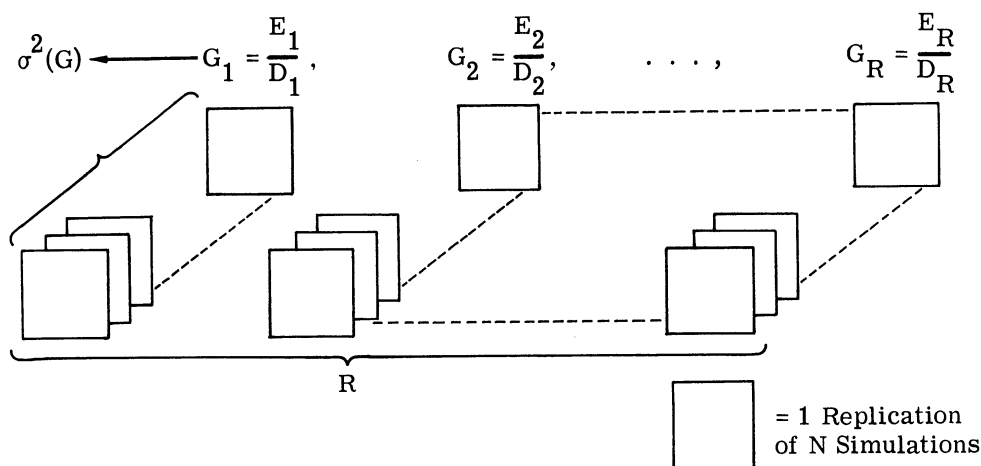


FIGURE 11. DESIGN FOR EXPERIMENTS ON IRREGULAR SPACES

To this end an information flow, similar to that shown in Fig. 5, was defined for these experiments (Fig. 12). A program RDEP was written to perform the simulations, the program COVAN was changed to allow the use of x/N as an estimate for p , and a new program DANAL was written to yield a value of G_i for each boundary point, a value G computed from the average of the D_i 's, and a value G_Σ corresponding to the set of points to be defined at the end of the next section. RDEP and DANAL are included in Appendix C.

5.2.2 Specification of the Spaces. Along with the requirement that the spaces be a random sample of all spaces, the following additional criteria were established for the reasons stated below:

1. In order to have data which would be comparable to the results of the previous sections, the number of non-absorbing points in the space should be between 100 and 200 and the points should lie within a 15 by 15 envelope.
2. In order to assure that most of the boundary points would have an absorption probability reasonably greater than 0, at least one-half of the grid points in the 15 by 15 square containing the space should be non-absorbing points in the irregular space defined.
3. For the same reason as above, no branch or arm of the space should be fewer than three grid units wide.

To this end, the following approach was formulated and used. A table consisting of 5 by 5 arrays of random digits was selected (Ref. 22). On a piece of squared paper, a 5 by 5 grid was marked off and an X was placed in each square of the grid which corresponded to a location for which the digit value was 5 or greater in a selected one of the 5 by 5 random digit arrays.

Figure 13 shows the result of such an operation with the digits recorded in their corresponding array locations. The largest possible envelope, so as to include every grid square, completely enclosed (horizontally and vertically) by marked squares was then constructed. Unmarked squares along the border of the array were not included. Figure 14 shows the results of the application of this rule to the array in Fig. 13. If the number of enclosed squares was greater than 12 (i.e., greater than half the possible number), then the result was reserved. If not, then a new envelope was constructed by applying the same rule, but with respect to unmarked squares.

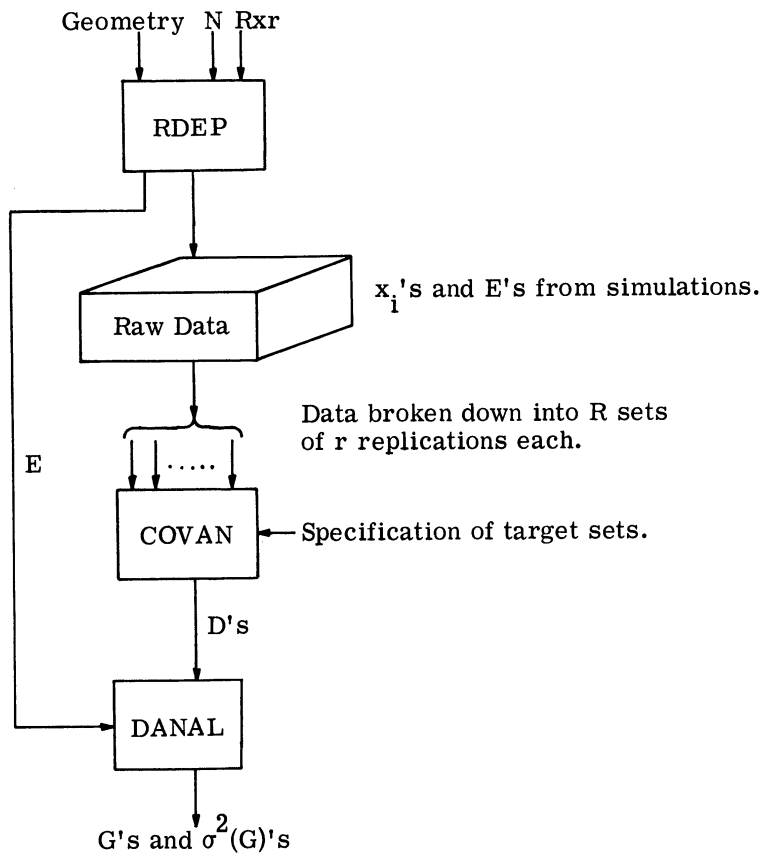


FIGURE 12. INFORMATION FLOW FOR EXPERIMENTS ON SPACES WITH IRREGULAR BOUNDARIES

x 6	x 6	0	x 7	1
x 9	3	x 8	4	x 8
4	x 6	x 9	0	2
0	3	x 6	3	4
x 7	x 5	x 6	x 9	x 5

FIGURE 13. FIRST STAGE IN THE DEFINITION OF AN IRREGULAR SPACE

x	x		x	
x		x		x
	x	x		
		x		
x	x	x	x	x

FIGURE 14. SECOND STAGE IN THE DEFINITION OF AN IRREGULAR SPACE

The result was a reserved set of fifteen 5 by 5 sub-arrays. Each grid point was then expanded into a 3 by 3 array resulting in new sub-arrays of size 15 by 15. The arrays thus derived were defined to be the non-absorbing points of the grid, with the nodes external to these the absorbing ones. Further use of the table of random digits resulted in a random placement of the source in each space by selecting two two-digit integers and using these as coordinates in an overlaid 100 by 100 space. The end result of these operations based on space shown in Figures 13 and 14 is shown in Fig. 15. The 14 other arrays are shown in smaller scale in Fig. 16. In each case "S" or a heavy dot indicates the source location.

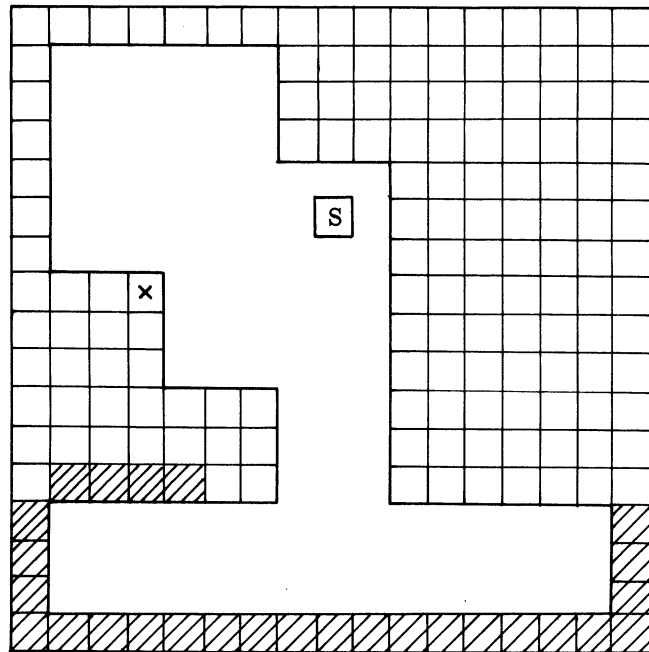


FIGURE 15. FINAL STAGE IN DEFINITION OF IR-REGULAR SPACE NO. 1. (S indicates the source location, x corresponds to the location for which G_{\min} was attained and the shaded area indicates the target set for G_{Σ} .)

In each of these figures it will be noted that some of the absorbing points are shaded. Each shaded area represents a subset of 25 points which defines a target set for which G_{Σ} was determined in the experiments. These groups of points were selected as follows. Consider an ordering applied to the absorbing points such that those in the lower rows precede those in upper rows, and, within rows, those on the left precede those on the right. Then the points indicated as belonging to the marked target sets are the first 25. The absorbing point for which G was found to be least (i.e., corresponding to G_{\min}) is marked with an X.

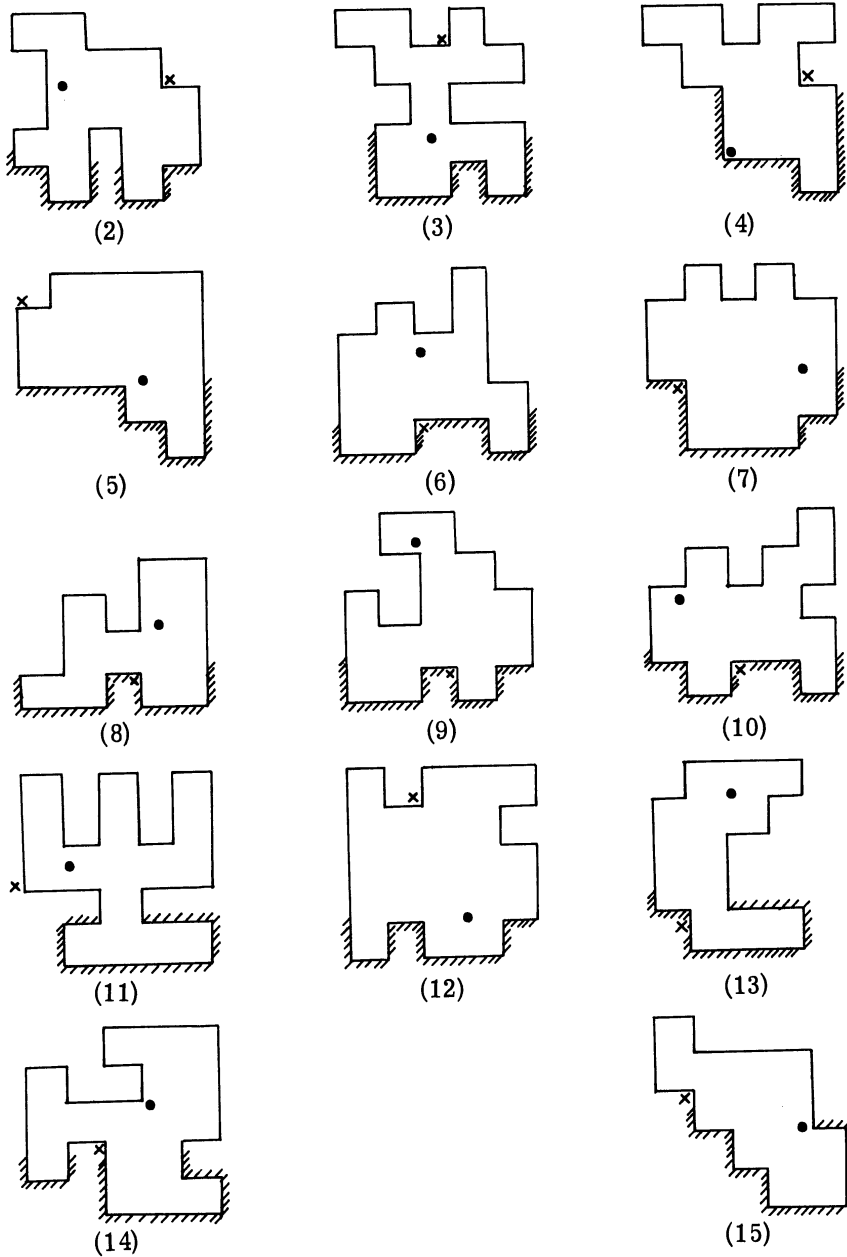


FIGURE 16. ADDITIONAL SPACES DEFINED. (The notation is as before except a large dot indicates the source.)

5.2.3 Experimental Results. From a preliminary investigation similar to that noted in Section 5.1.2, it was decided that the following parameters would provide suitable precision for this work:

N (number of simulations per replication) = 1500,

r (number of replications to determine G) = 20,

R (number of sets of replications to determine an average G and $\sigma^2(G)$) = 10.

Since it was desirable to have data for as many spaces as possible consistent with reasonable use of computer time, it was decided not to obtain $\sigma^2(G)$ for some of the spaces but to perform a total of $2r = 40$ replications in these cases. Thus, $\sigma^2(G)$ is reported for only 5 of the 15 spaces used.

Table VI contains the results of the experiments on spaces with irregular boundaries. In this table:

E is the average of the values of E obtained experimentally across all replications.

D is the average of the D_i for each point obtained across either 10 sets of 20 replications, or 40 replications.

k is the average of the values of k obtained experimentally.

G is E/D

G_{\min} is $\min(G_i = E/D_i)$ across boundary points i.

p_{\min} is the probability estimate x_i/N at point i defined above.

G_{Σ} is the ratio of E to the value of D based on the set of 25 points defined in the previous section.

$R \times r$ is the number of replications performed.

5.3 Interpretation of Results

This section contains a discussion of the results of the experiments described in Sections 5.1 and 5.2. Section 5.3.1 contains a general discussion of the parameter G. Sections 5.3.2-5.3.4 amplify the results as they affect the relationships between the geometry of the space in which the random walks were simulated and the parameters E, D, and G, respectively.

5.3.1. General Comments on G. In the experiments conducted, $G > 1$ for target sets of individual points (G in Tables III and VI, and G_{\min} in Table VI). For target sets of more than one point in square spaces $1 < G < 2$, except when the number of points in the set is close to the minimum or maximum possible, or there are gaps in the set ($G_{1/2}$, $G_{1/4}$ in Table IV and G in Table V). In the spaces with irregular boundaries the results for target sets of 25 points (G_{Σ} in Table VI) show that $1 < G < 4$ within one standard deviation (based on the assumption that variances for spaces 6-15 are similar to those obtained for spaces 1-5).

TABLE VI. Results of Experiments on Irregular Spaces

Space	G	G _{min}	G _Σ	E	D	k	Pat min	R × r
1	6.206 (0.209) ⁴	4.212 (4.446)	1.362 (0.102)	10.100 (0.161)	1.625 (0.0137)	0.456 (0.000202)	0.0357	200
2	7.008 (0.350)	3.190 (0.933)	1.052 (0.114)	13.971 (1.148)	2.005 (0.0257)	0.492 (0.000233)	0.0130	200
3	6.270 (0.353)	4.099 (0.1573)	1.140 (0.123)	8.918 (0.0308)	1.434 (0.0155)	0.470 (0.00840)	0.0031	200
4	3.716 (0.0246)	1.426 (0.114)	0.878 (0.0308)	4.146 (0.159)	1.117 (0.00204)	0.180 (0.000451)	0.0057	200
5	7.127 (0.438)	4.531 (7.754)	1.131 (0.197)	15.213 (0.828)	2.149 (0.0289)	0.499 (0.000289)	0.0130	200
6	6.630	3.551	0.921	12.133	1.830	0.466	0.0632	40
7	7.800	3.458	0.924	17.022	2.160	0.510	0.0184	40
8	3.723	1.862	1.357	5.008	1.345	0.292	0.0009	40
9	5.743	1.997	0.818	9.797	1.706	0.447	0.0306	40
10	4.878	1.684	3.641	7.488	1.535	0.404	0.0394	40
11	4.392	1.991	2.189	5.745	1.308	0.390	0.0199	40
12	5.478	2.203	1.104	9.499	1.734	0.473	0.0028	40
13	5.351	2.652	2.295	9.423	1.761	0.442	0.0136	40
14	6.924	2.958	1.573	11.023	1.592	0.430	0.0426	40
15	5.362	2.309	1.715	9.507	1.773	0.324	0.0058	40

⁴Numbers in () are variances of the data above.

The data in Table V (plotted in Fig. 10) show that, for closed sets of points in square spaces, G is distributed symmetrically with respect to the number of points in the set. This is because G varies only with D in these cases and, as shown in Section 3.1, D is dependent on the product pq. That the product is symmetric about $p = q = \frac{1}{2}$ offers an explanation of these results.

The data for open sets in Table V show that, after the set spans about one-fourth of the points (including only alternate points in the actual target set), G increases quite rapidly. This can be explained in the following way. Since adjacent pairs of points are not included, maximum correlational effects are not present; but as the set expands and includes more remote points, the amount of negative correlational effect increases greatly.

In summary, within one standard deviation, $G > 1$ for all experiments conducted, and for many cases $G \gg 1$.

5.3.2 E as a Function of Geometry. In Section 4.3 a simple method for estimating E directly from the geometry of the space is hypothesized. The estimate E_e is proposed to be proportional to the geometric mean of the area of the largest interior rectangles (including the source) included in each of the four subspaces defined by drawing lines through the source point. (See Section 4.3 and Fig. 3.) Figure 17 is a plot of the experimentally obtained E as a

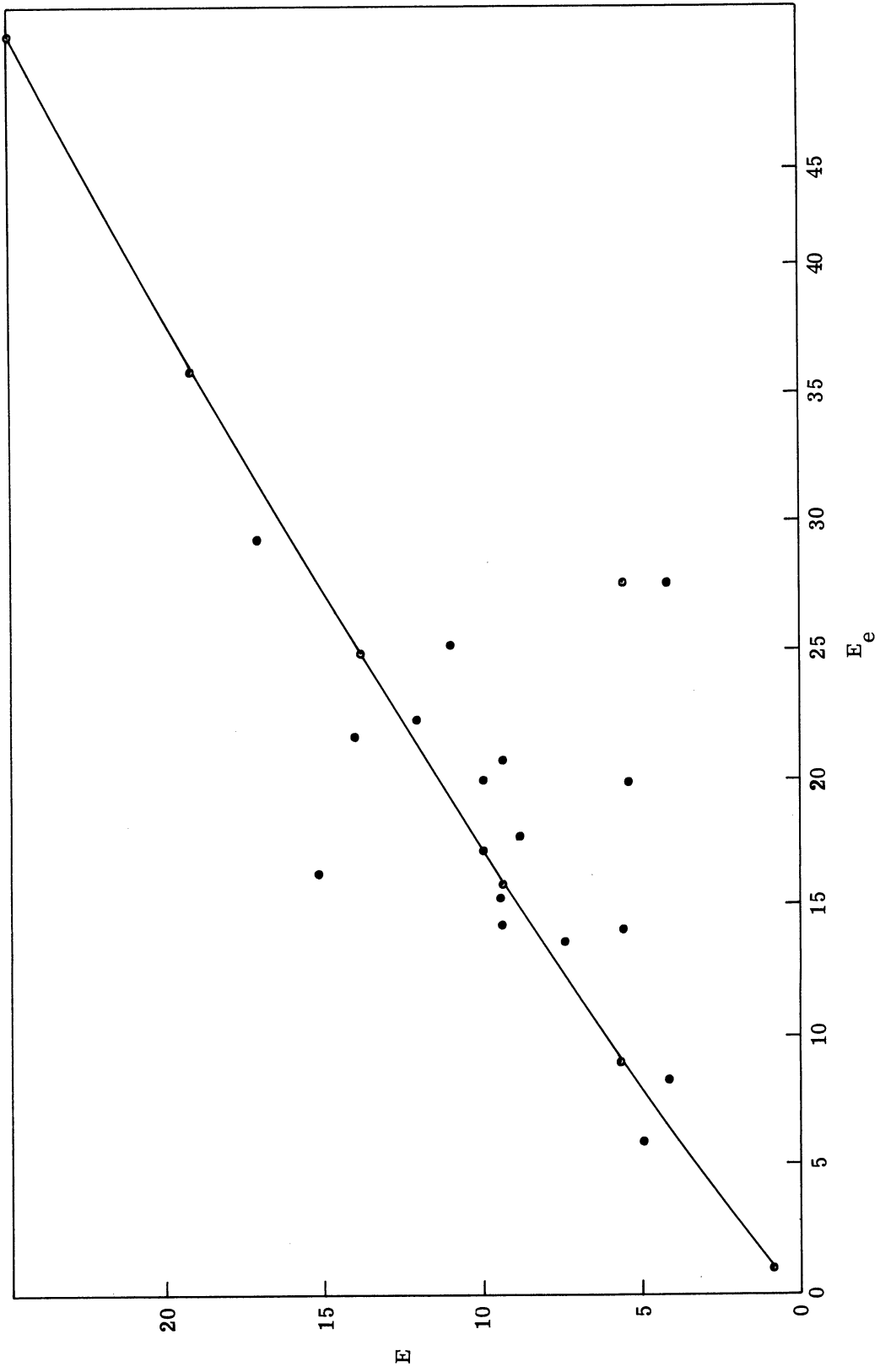


FIGURE 17. E AS A FUNCTION OF E_e . (Points connected by the line represent data from square spaces.)

function of the estimate E_e . The line in the figure connects the points from the data in Table II while the other points depict the data in Table VI. The computed correlation coefficient for the values of E and E_e for irregular spaces in Fig. 17 is

$$\rho_{E,E_e} = .979, \quad (5.11)$$

with standard deviation

$$\sigma_{\rho_{E,E_e}} = .005. \quad (5.12)$$

Thus, the hypothesis that $E \sim E_e$ as originally defined in Section 3.3 is supported for these spaces.

However, from evaluations of E and E_e in long, narrow rectangles (see Section 3.2.2 and Fig. 18), it is clear that the hypothesized prediction method is not valid for these spaces as the rectangle lengthens. In Fig. 19 it is noted that the prediction approximates the data adequately for rectangles up to at least 5 by 11. (The prediction line is that for the square spaces, Fig. 17.) Thus, it can be stated that the prediction method is valid in cases where the length of the rectangle is greater than at most 3 times the width, measured across non-absorbing points. The upper limit 3 of this ratio is conservatively low, especially since it was established for a space only 3 non-absorbing grid units wide. The data for the irregular spaces further support the conclusion that E_e is a useful estimate of E for the vast majority of spaces.

5.3.3 D as a Function of Geometry. In Section 3.3 it was also hypothesized that

$$D \sim E^{1/2}. \quad (5.13)$$

Figure 20 is a plot of D versus $E^{1/2}$ for the experiments performed. These data for irregular spaces correlate with coefficient

$$\rho_{D,E^{1/2}} = .999 \quad (5.14)$$

and standard deviation

$$\sigma_{\rho_{D,E^{1/2}}} = .0008. \quad (5.15)$$

Thus, these data support this hypothesis.

The data for square spaces (the line in the figure) form an upper bound on D as a function of $E^{1/2}$. This is reasonable since it is expected that an irregular boundary will tend to absorb a single one of a pair of particles more often than will be regular boundary due to the convex corners which arise in these spaces, thus decreasing correlation.

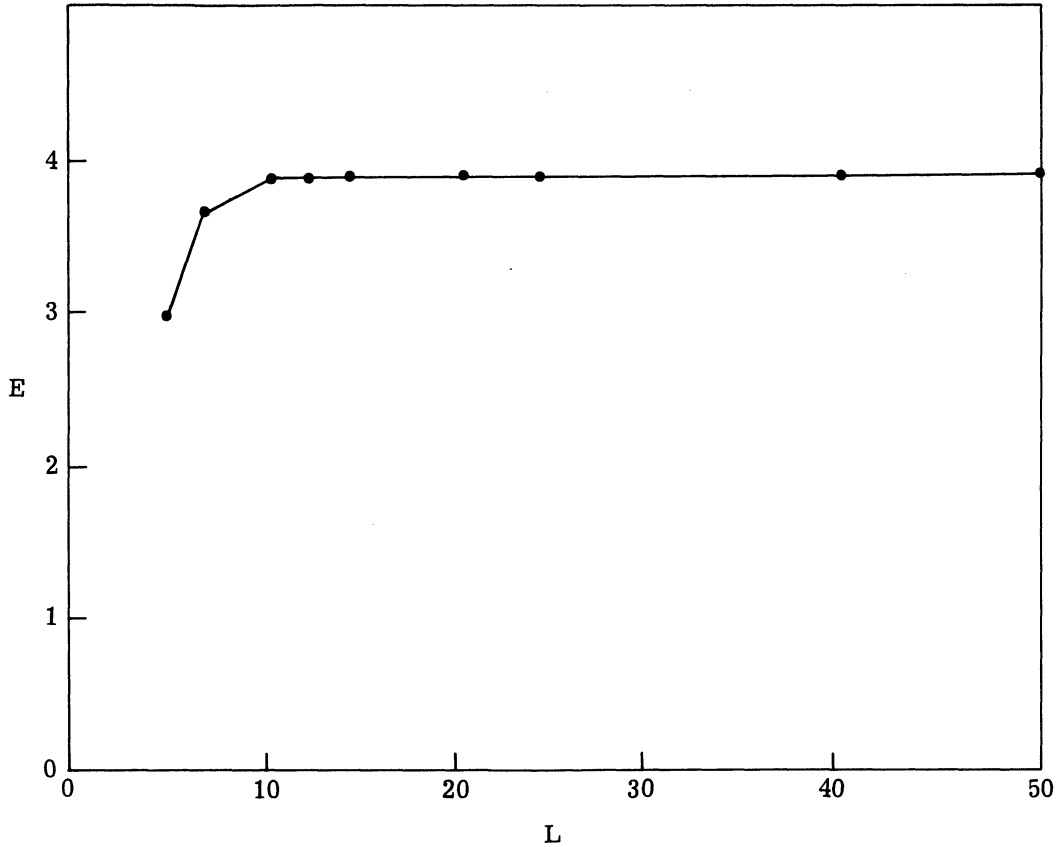


FIGURE 18. E AS A FUNCTION OF L FOR 5 BY L RECTANGLES WITH SOURCE AT POINT (3, 3)

5.3.4 G as a Function of Geometry. In Section 3.3 it was further hypothesized that

$$G \sim E^{1/2}, \quad (5.16)$$

and that E_e could be substituted for E in Eq. (5.13) to yield

$$G \sim E_e^{1/2}. \quad (5.17)$$

Figure 21 is a plot of G versus $E_e^{1/2}$ for the experiments discussed previously. Using the data from irregular spaces, one can correlate these parameters with coefficient

$$\rho_{G, E_e^{1/2}} = .996, \quad (5.18)$$

and standard deviation

$$\sigma_{\rho_{G, E_e^{1/2}}} = .0019. \quad (5.19)$$

Thus, for the spaces used in all experiments the hypothesis is confirmed.

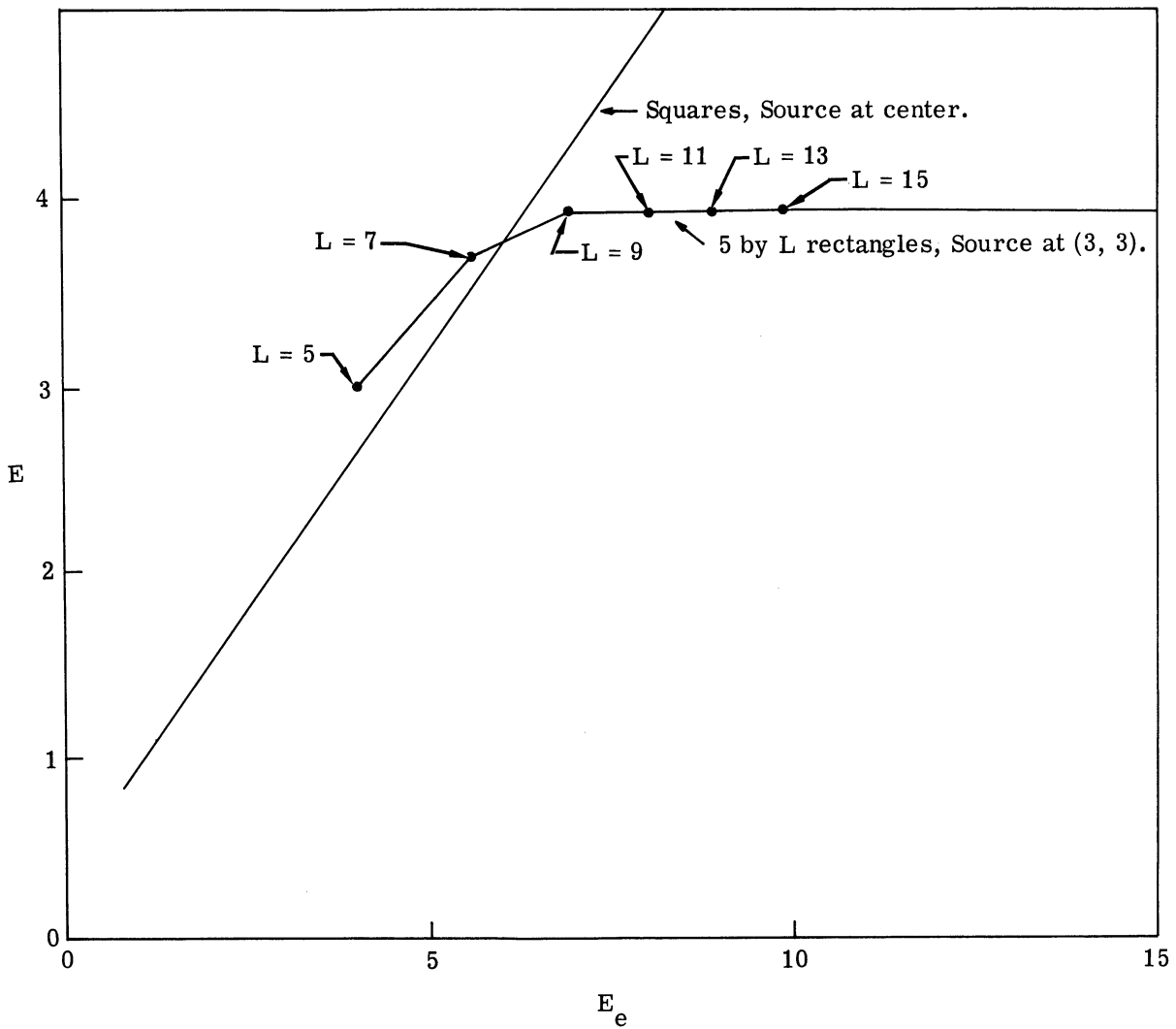


FIGURE 19. E AS A FUNCTION OF E_e FOR SQUARES AND 5 BY L RECTANGLES

From the figure it can be further hypothesized that $E_e^{1/2}$ for the square spaces establishes a lower bound on G . This can be justified by considering the argument of the previous section that D for square spaces should be an upper bound on all D 's and that G is inversely proportional to D .

Therefore, by establishing the coefficients of the line in Fig. 21, a lower bound on G is

$$g = (1.3 \cdot E_e^{1/2} - .3) \leq G. \quad (5.20)$$

This is applicable in all cases where E_e is a suitable estimate for E .

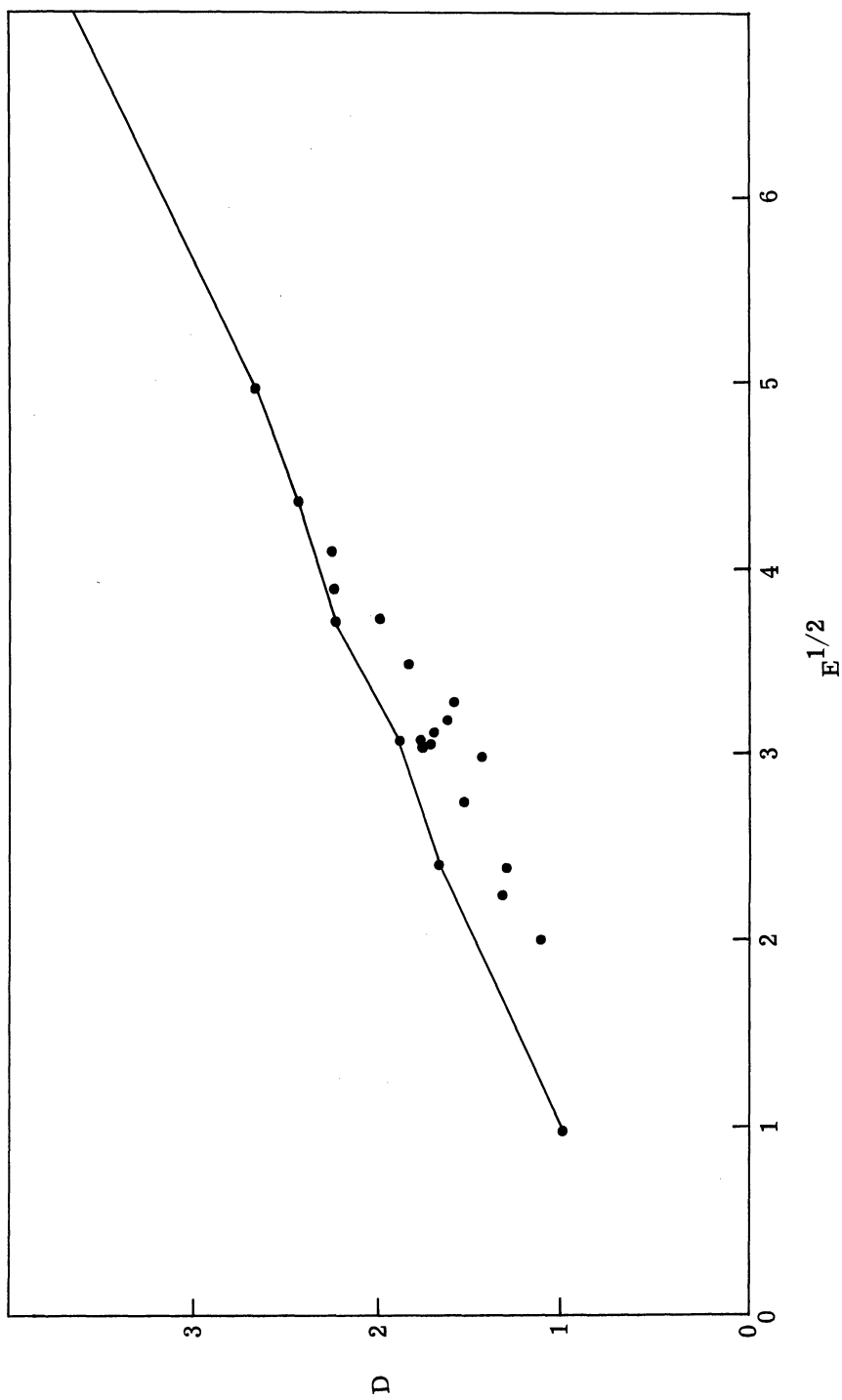


FIGURE 20. D AS A FUNCTION OF $E^{1/2}$. (Points connected by the line represent data from square spaces.)

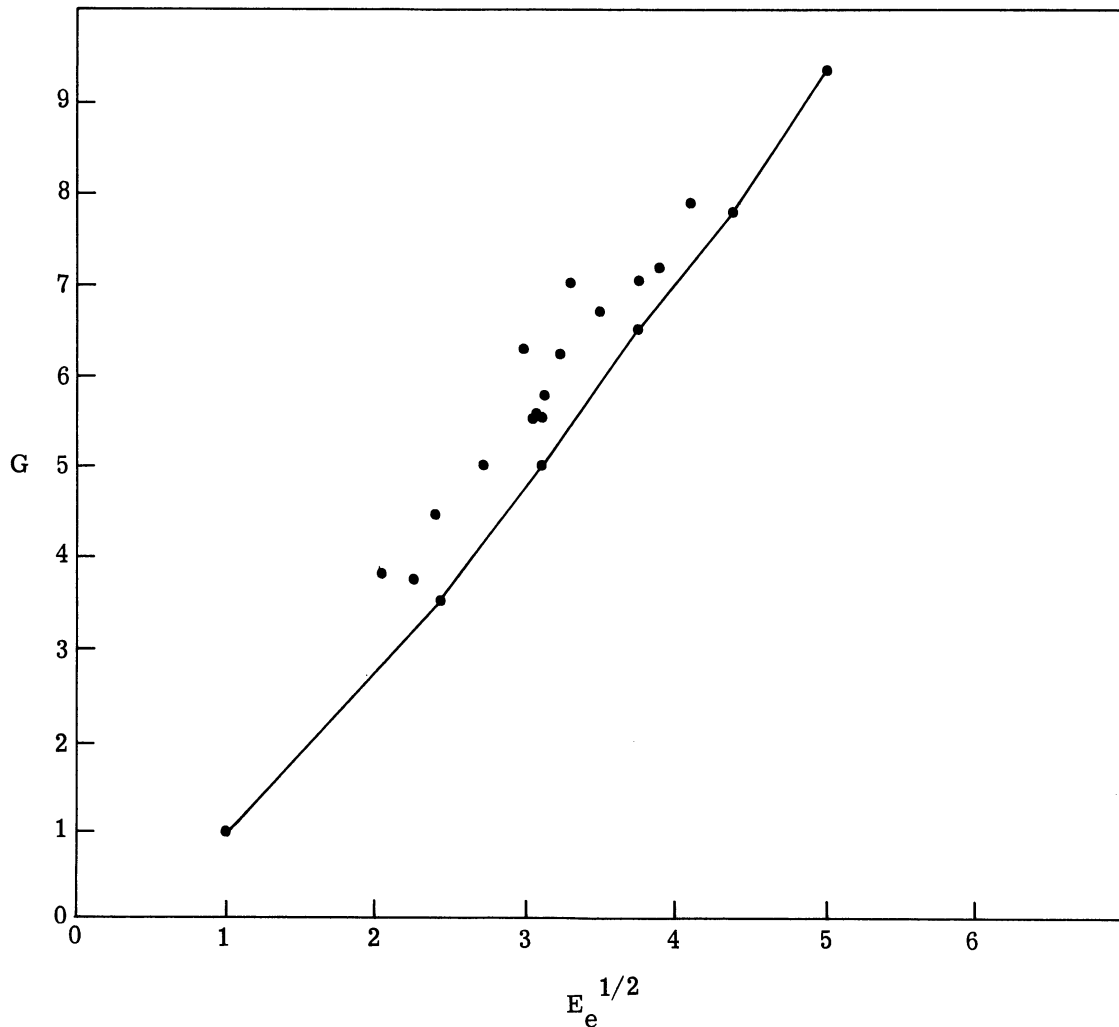


FIGURE 21. G AS A FUNCTION OF $E_e^{1/2}$. (Points connected by the line represent data from square spaces.)

CHAPTER VI CONCLUSIONS

6.1 The Use of the Algorithm A_1 on Highly Parallel Computers

The following conclusions are based on the results reported in the sections cited in each case. The conclusions concern the relative effectiveness of G (Section 3.1) of algorithms A_0 and A_1 (Section 2.1) when used to simulate absorbing boundary random walks on highly parallel computers (Section 1.1). They pertain to random walks simulated in spaces of dimension two, unless stated otherwise. The first six conclusions result from experimental work on symmetric random walks while the last two are based upon general theoretical investigation.

a) Within one standard deviation, the experimental results show $G \geq 1$ for any target set (Section 5.3.1).

- b) For target sets consisting of single points, G increases as the size of the non-absorbing space surrounding the source increases (Sections 5.1.3 and 5.3.3).
- c) For target sets consisting of unbroken chains of points whose total absorption probability lies between $1/4$ and $3/4$, $G > 1$ and, generally $G < 4$ (Section 5.3.1).
- d) For target sets consisting of alternate points along the boundary $G \gg 1$ (Section 5.1.3).
- e) For all but very long thin rectangles it is possible to predict G directly from the geometry of any two-dimensional space such that the prediction correlates 0.99 with the values of G obtained experimentally (Section 5.3.3).
- f) Based on the prediction cited above, for all but very long thin rectangles, there exists a lower bound on G based only on the spatial geometry;

$$G > 1.3(a_1 \cdot a_2 \cdot a_3 \cdot a_4)^{1/8} - .3 \quad (6.1)$$

where the a_i 's are measures of area more precisely defined in Section 4.3 (Section 5.3.3).

- g) For symmetric random walks in one-dimensional spaces, $G = 1$ (Section 4.1). For asymmetric walks, $G > 1$.
- h) Subject to one condition, which is intuitively plausible, for any space of any dimensionality $G \geq \frac{1}{2}$, and subject to one of two other conditions $G \geq 1$ (Section 4.2).

In summary, these conclusions indicate that it would be generally advantageous to use A_1 instead of A_0 for simulating random walks on a highly parallel computer. Even for target sets of least advantage, $G \approx 1.25$ which represents a 20-percent saving in computer time for a set of simulations. Considering that the use of random walk techniques often requires a very large number of particle simulations, hence a large amount of computer time, this 20-percent saving can be important. In the cases where the target set is a single point and the space large, much greater savings can be expected. For example, using the prediction formula (6.1), for a geometry of the same shape as that in Fig. 15 but 5 times larger in each dimension, by using A_1 one can expect to obtain results in 1/30th the time required for similar results using A_0 .

Personal communication with staff members at the Westinghouse Corporation indicates that these results may be of immediate use when applied to the SOLOMON computer (Section 1.1). In their investigations they have concluded that:

- a) Simulation of an independent random walk by each processing element (module) of SOLOMON would require more storage per element than available.
- b) Simulation of a single grid point by a single processing element would result in more than one particle occupying some grid points, which in turn demands a processing element of too high a level of sophistication.

As discussed in Section 2.1, the algorithm A_1 studied here avoids each of these drawbacks, yet, as shown in this research, it does show a distinct advantage over A_0 in most cases. In Section 1.1 it was mentioned that for certain combinations of spatial geometries and sizes of arrays of processing elements, SOLOMON is not so effective as the more general highly parallel computer. However, this drawback in many cases of practical interest would not be so severe as to discourage the use of A_1 on this computer.

6.2 The Use of the Algorithm A_1 on Contemporary Computers

From the experiments performed the following conclusions can be drawn. One time-step of a random walk simulation on a contemporary computer using A_0 can be accomplished by updating a pair of registers containing the coordinates of the particle being simulated and comparing this to a table of coordinates of absorbing points. The necessary search can be accomplished in as few as two indirect addressing cycles and is very fast.

The techniques developed for simulating A_1 in this research have resulted in some computational concurrency on a conventional computer by using the contents of each of the n bit positions of the computer word to represent the absence or presence of a particle. The use of shifting and masking techniques allows rapid processing of these data.

However, to be able to use this technique it is required that the space be partitioned into units which can be represented by one machine word, and there must be a program to link these words as defined by the geometry. As the size of the space increases, the number of words and complexity of linking increase. The lack of true parallelism and the added drawback for large spaces leads to the conclusion that the use of A_1 on contemporary computers will not result in any advantage over the use of A_0 .

6.3 Suggestions for Further Research

The investigation reported here leads to the following suggestions for further research:

- a) The assumptions upon which the theorem in Section 4.2 is based should be substantiated.
- b) Variations on the random walk defined in Section 2.1 should be investigated. Suggested variations are:
 1. asymmetrical distribution of the values of the random variable used to determine direction of particle motion,
 2. random walk with spontaneous particle emission,
 3. spaces with isolated absorbing points, and
 4. spaces of dimension 3 and higher.

The results of Section 4.1 indicate that research on Item b1 above will lead to the conclusion that algorithm A_1 is at least as advantageous when the transition probabilities are asymmetric as it is in the symmetric case.

The results of Section 4.2 indicate that $G \geq \frac{1}{2}$ and probably $G \geq 1$ for spaces of dimension 3 and higher. In these cases the fraction of time that the source is covered k will be less than in the two-dimensional case. Since there is no reason to believe that D will increase to offset the resultant increase in E , research under item b4 should prove A_1 to be of even greater advantage in these cases than in the two-dimensional case.

c) Since it is likely that highly parallel computers will become available in the near future and, as shown in this research, solving some problems by using conventional algorithms does not always take full advantage of such systems, there is a large number of applications for which novel algorithms based on these new organizations should be developed and verified.

Appendix A-1

A SPECIAL PURPOSE COMPUTER FOR PARTIAL DIFFERENTIAL EQUATION SOLUTION AND OTHER ITERATIVE ALGORITHMS

A.1.1 Introduction

The computer organization described below is an extension of a concept of Leondes and Rubinoff (Ref. 10), although it was invented independently by this author ten years later. The original machine consisted of a drum store and a single processor capable of sequencing its access from one drum track (array column) to another at the end of each revolution. In this way it could process one iteration of a Laplace equation in n drum revolutions where n is the number of columns of the array.

The present author has extended this idea by allowing one processor per drum track. It will be shown in this paper that a computer of this type can be produced and used in a very economical way without extension of current technology. Other possible applications of such a device will also be discussed.

A.1.2 The Computer

Partial differential equation solutions usually require a large number of identical iterations over a large array of data. There are computer organizations proposed which tend to be highly suitable for the algorithms (Refs. 4, 5, 8). However, it is often the case that the parallel nature of the computer results in its being costly. The chief reason for this appears to be the generality of the individual processing units and the expense of their associated stores.

The approach to this problem which is considered here involves a slow, thus inexpensive, store and a set of processors whose power and flexibility are well below that of any general purpose computer. As is done in many parallel machine organizations, the store is to be considered as a two-dimensional array. Unlike many such designs, there is not one processor for each element of store, but merely a one-dimensional line of processors. Thus, we should achieve a design for a computer which is less flexible and slower in executing basic operations over the plane than would be a system of more general character.

It is appropriate to ask how a line of processors can operate on a plane of storage modules. One solution is to move the processors about. But, at electronic operating speeds, mechanical movement seems impractical. However, considering moving mass storage devices brings to mind the magnetic disc and the magnetic drum which have the characteristic that the store is constantly in motion. To simplify problems of design, the drum was chosen as the basic store of this machine.

The processors themselves are considered to be small computers, one per drum track, fed by read heads from the tracks on the drum and putting out information through write heads. As the drum turns through one revolution every word of data (stored serially) comes into the reach, hence within processing power, of one or more processors (Fig. 22).

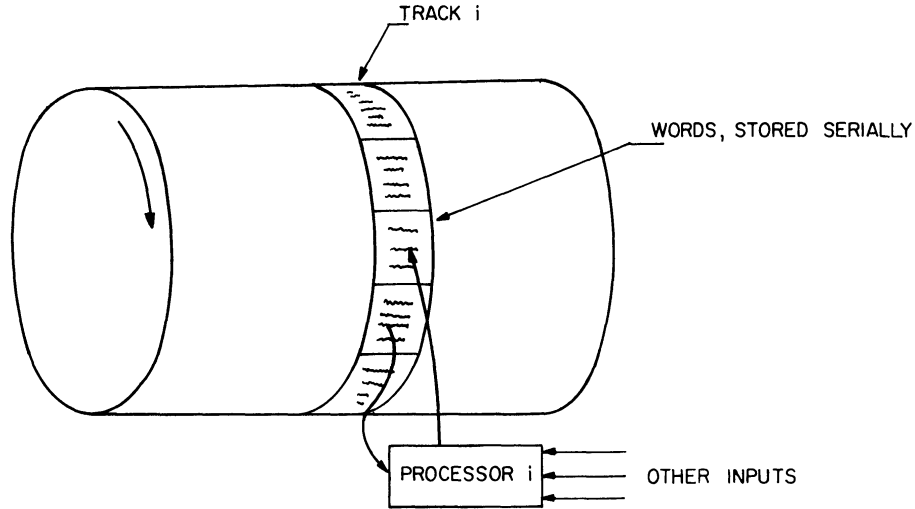


FIGURE 22. GENERAL ORGANIZATION OF THE COMPUTER

A.1.3 A Primary Application

Consider the solution of partial differential equations; using finite difference methods, a solution to the Laplace equation with Dirichlet boundary conditions can be formulated in the following way:

$$V_p^{t+1} = \frac{1}{4} (V_n^t + V_s^t + V_e^t + V_w^t) \quad (\text{A.1})$$

where V_p^{t+1} is the value of the point p at time $t + 1$ and $V_n^t, V_s^t, V_e^t,$ and V_w^t are the values of the four neighboring points in the north, south, east, and west directions, respectively, at time t (Refs. 22, 23). If this computation is executed for all points under consideration in the space until $\left| V_p^{t+1} - V_p^t \right| \leq \epsilon$ for all p , then the resultant values at each point describe the solution to the finite difference problem within the error described by ϵ . The classical diffusion problem is solved in this manner.

In our new organization, each track on the drum represents a column of the array; the line of processors works on one row at a time with one processor per column. The appropriate values for processor input are read from its own and immediately adjacent tracks and delayed appropriately by delay line (Fig. 23). A comparison for termination criteria is also made. In each revolution of the drum one new value for each point is computed; thus, one revolution is equivalent to one iteration over the plane.

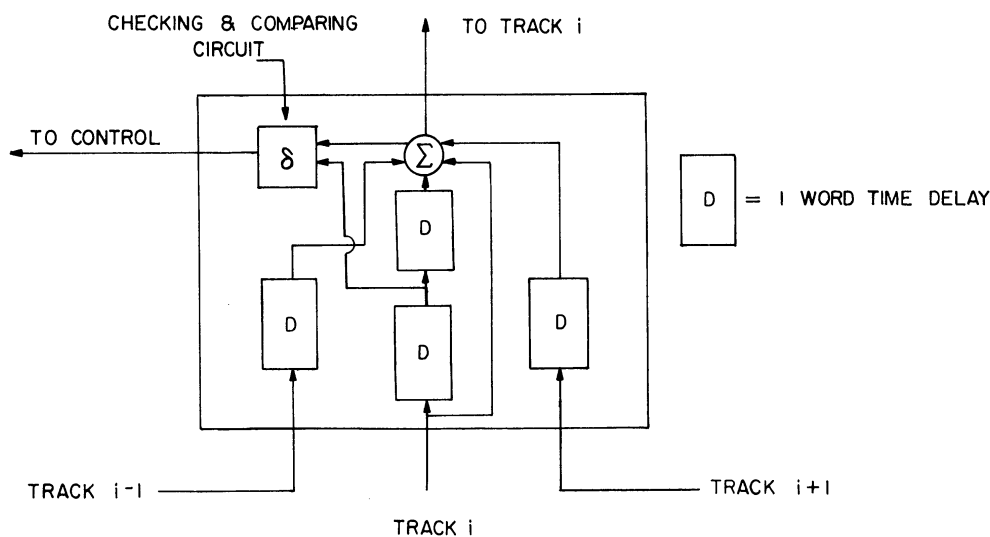


FIGURE 23. INDIVIDUAL PROCESSOR FOR LAPLACE EQUATION

Three-dimensional diffusion problems can also be carried out using a modified version of this algorithm, hence using this organization. This application would involve the weighted sum of six inputs, with layers being represented by sets of tracks. The solution of equations in greater than two dimensions appear to be very important to some areas of study. Such solutions are currently extremely expensive to obtain.

From communications with numerical analysts⁵ it appears that the power of this form of organization could be enhanced if the algorithm executed were of a more general nature. First of all, the method of optimum over-relaxation allows solutions to Laplace's equation in a fraction of the number of iterations required by the algorithm of Eq. (A.1). This over-relaxation algorithm for any point, p , takes on the form

$$V_p^{t+1} = V_p^t + \omega \left[\frac{1}{4} (V_n^t + V_s^t + V_e^t + V_w^t) - V_p^t \right].$$

At first the multiplication by the constant, ω , appears to be particularly formidable in a serial machine of this type. However, $0 < \omega \leq 2$, and further investigation indicated that approximation of ω to within one-eighth of its optimum value for any set of equations yields solution in at most 110 percent of the optimum number of iterations. A number of this form can be represented as a 4-bit integer. Multiplication of a number by a 4-bit constant is no more difficult

⁵ Particularly, Prof. Jim Douglas, Jr., of Rice Institute, Prof. Richard Varga of Case Institute, and A. Downing of Oak Ridge National Laboratory.

than building a 4-input adder with appropriate unit delays. This is accomplished using a set of gates controlled by the value of ω and applying the sum (in parentheses above) at each of the adder inputs. (See Fig. 24.)

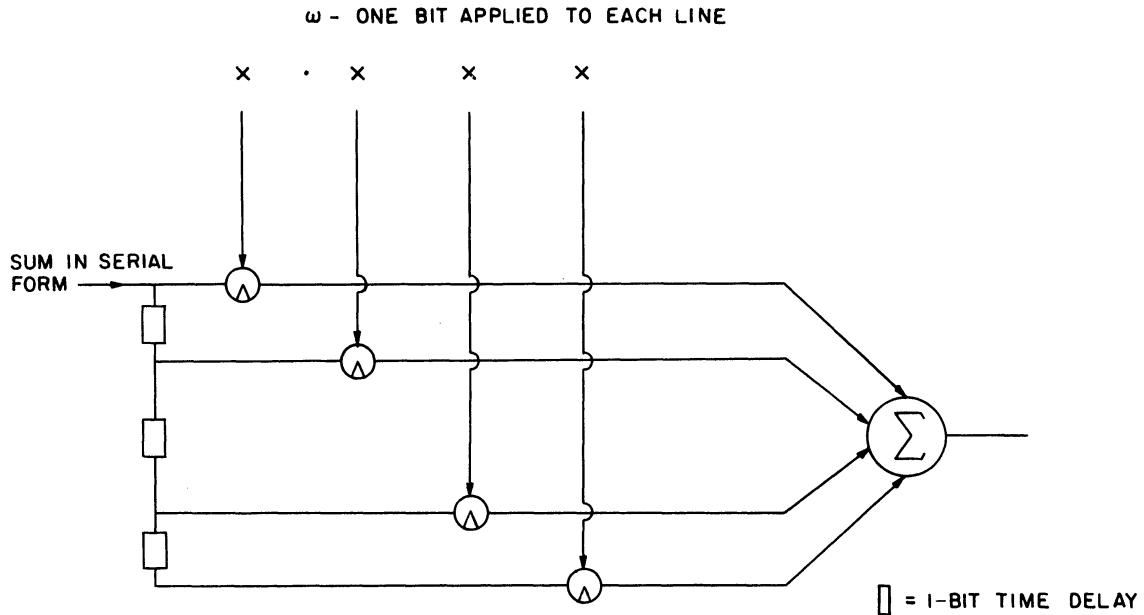


FIGURE 24. FOUR-BIT MULTIPLIER FOR OPTIMUM OVER-RELAXATION MODULE

A second improvement can be made if equations more general than the Laplace form could be solved. This would be accomplished if, in the computation of a new value for any grid point, the values representing the neighboring points can be multiplied, each by a particular, and generally different, weighting constant. For example:

$$V_p^{t+1} = V_N^t w_N + V_S^t w_S + V_E^t w_E + V_W^t w_W$$

If the four w 's for one point can be stored in a single machine word, then this result can be achieved by halving the effective amount of storage to 375×500 . However, four fast multipliers per module are necessary and this could be an excessive expense. On the other hand, if the w 's can be kept to as few as 6 or 8 bits, then the technique employed earlier might be applied.

A.1.4 Evaluation

A large drum spinning at 1500 rpm can carry out 1500 iterations on this problem per minute. Drums are currently available which are large enough to carry 375 by 1000, 40-bit

word arrays and can revolve at these speeds.⁶ On arrays of this size the organization would allow a 2000-iteration solution of Laplace's equation in approximately two minutes including loading and unloading the drum. If storage limitations are neglected, this computation would require approximately ten hours on a conventional computer such as the IBM 7090. Smaller arrays would involve the use of smaller drums capable of angular velocities at least 10 times greater than the 1500 rpm quoted. There would also be commensurate savings in overall time due to the fewer iterations required to solve such problems.

It is naive to consider such a machine being useful by itself. Our current thought leads us to believe that this device should be considered functioning as a special purpose input-output device on a fast general purpose machine whose purpose would be to load and unload the drum store and to perform desired computations necessary for preparing input and analyzing output. Costs for the 3.75×10^6 word device break down as follows:

2 Drums	\$ 30,000
Read/write heads and amplifiers at \$300 per track	300,000
Processors at \$1000 per track	1,000,000
General purpose computer	<u>3,000,000</u>
	\$4,330,000

The estimate for track processor cost is very generous and the other estimates are based on available equipment.⁷ The total price of the system is such that it appears to be well within range of eclipsing the effectiveness of any available or proposed system when measured in computations per dollar. But, so far in this presentation, this is true for only one problem, and this drawback is serious.

A.1.5 Other Applications

In general, to be truly useful any computer system must be operating for several hours a day or its great expense may be wasted since a less expensive, slower system should suffice for the available workload. In the case of the system under discussion, it has an immediate advantage in that over 50 percent of its cost is represented by a general purpose computer, the operation of which is in no way impaired by the attachment of the drum device and which may be used in a conventional way at times when the drum is inactive due to lack of need, disrepair, etc.

⁶Current solutions involve only about 30,000 points, due mainly to the size of the stores of the machines.

⁷These figures arise from personal communication with the Bryant Division, Ex-Cell-O Corporation, and "Computer Characteristics Quarterly," Charles Adams Associates (1962).

On the other hand, seeing that a new application involves no more than a change of the individual processors, we asked ourselves for what other applications might this organization be useful. One which came immediately to mind through its use in solution of partial differential equations is the random walk. To this end, a new, highly parallel method for simulation random walks is currently under active investigation. It has been suggested that implicit methods for solution of partial differential equations could also be developed to take advantage of this organization.

Another application involves parallel-search memories (associative memories)(Ref. 24). Both ordered-multiple-occurrence and single-occurrence memories can be simulated with fixed word length required in the former case. The search and access time would be relatively high (on the order of 50 μ sec to 1 msec) depending on the size of the store, but the relative low cost of this device makes it worth consideration for this application. The uses to which search memories can be put include information retrieval applications, bubble chamber photo-analysis, etc.

Many other applications can be considered. A most interesting example is the pattern recognition computer of McCormick and Divilbiss (Ref. 9) which apparently can be simulated in this organization. The advantage here again appears to be low cost.

A.1.6 Conclusion

The opinion held by some that special purpose computers can be organized to be more efficient and economical for certain applications than more general purpose computers prompted the investigation reported here. Our results are quite favorable in supporting this opinion and comment from other sources has been generally favorable.

Appendix A-2

IMPLEMENTATION OF THE ALGORITHM A_1 ON THE COMPUTER OF APPENDIX A-1

Hardware or program implementation of the algorithm A_1 on some machine capable of a large degree of concurrent processing is relatively simple. For example, Figs. 25 and 26 show a detailed logical design for a processor of such capability for the drum machine described in Appendix A-1. (These figures do not include the additional circuitry to load and unload the drum between simulations.) Briefly, the drum contains, stored serially in one word, the information describing the state of the grid point simulated by that word (i.e., is this a boundary point, is this a source point, was this point occupied at previous time step) and the contents of a counter to be used to count absorptions or insertions at that point. With many words stored on each drum track and many tracks and processors across the drum, as the drum revolves each processor simulates the activity of each grid point in a particular column for one time-step. Considering a space of maximum size 200 by 200 grid points and less than 10^{10} random total walks desired per simulation, a 200-track drum holding two hundred 40-bit words per track would be suitable for this simulation.

The processor itself consists of a counter and some simple circuitry to distribute particles (i.e., to set appropriate "occupied" bits). Using delay line storage, it is possible to develop a new storage configuration representing some module and delay its output so as to have simultaneous serial reading and writing as the drum revolves with no danger of premature erasure. A central control device capable of counting total absorptions and issuing random variable values must also be provided. The module as depicted here contains fewer than 30 active elements (gates) and 16 delay lines of various lengths, including the hardware necessary for the flip-flops depicted.

The counter module is used for forming the unit increment or unit decrement of the previous count depending on whether the module is simulating a boundary point or source point. Again, due to the bit-wise serial operation of the system, the circuitry for such devices can be rather elementary. Flip-flop No. 1 in the counter is used for retaining information defining a source simulation; No. 2 is used to detect simulation of an occupied boundary point or a non-zero source count, and the third to complement the low-order bits of the counter in performing the unit change.

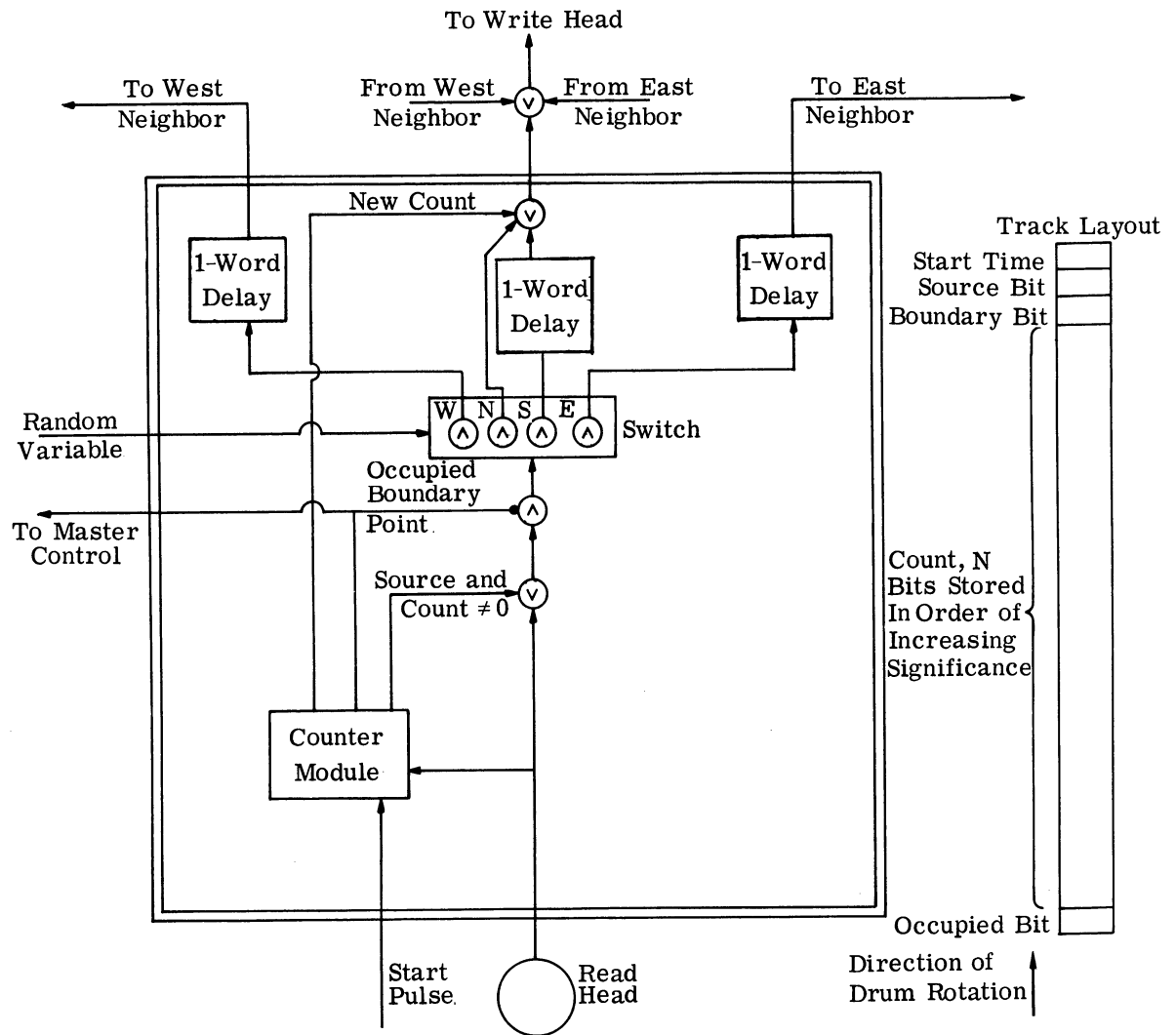


FIGURE 25. CONCURRENT RANDOM WALK PROCESSOR FOR DRUM MACHINE

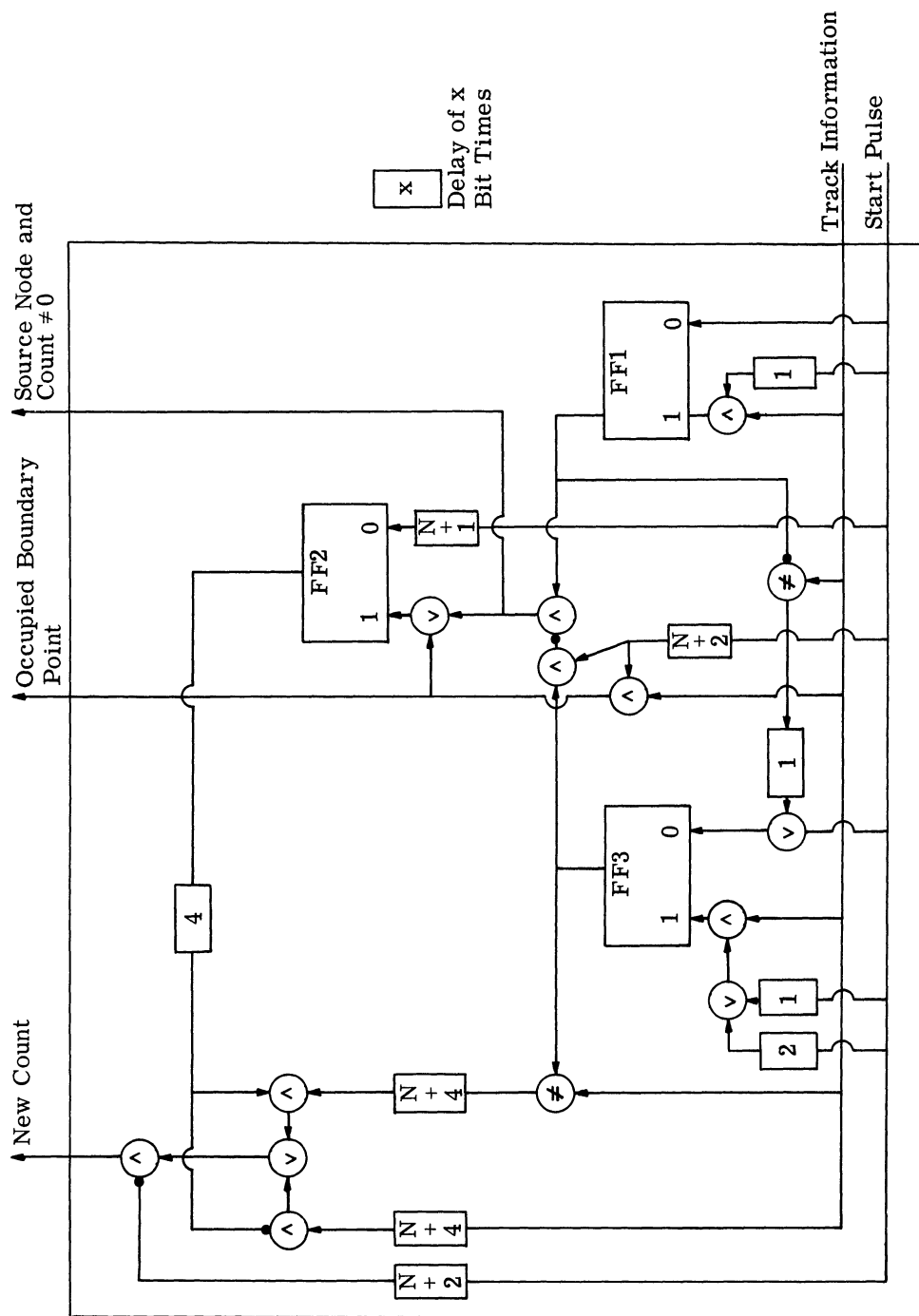


FIGURE 26. COUNTER MODULE FOR RANDOM WALK PROCESSOR

Appendix B

CONCURRENT RANDOM WALKS IN A ONE-DIMENSIONAL SPACE
WITH THE SOURCE AT THE CENTER

by James Dickey

The cells are ordered from left to right, 1, 2, . . . , 2K + 1; cells 1 and 2K + 1 are absorbing, and cell K + 1 is the particle source. At each step in the process either all unabsorbed particles move to the right by one cell with probability p, or all unabsorbed particles move to the left by one with probability q = 1 - p. A new particle is born at cell K + 1 whenever a step leaves it unoccupied (concurrent random walks). (For non-concurrent random walks, a new particle is born whenever a step leaves all the cells unoccupied.)

The process is observed until N particles have been absorbed. We wish to describe the random variables N_ℓ and N_r , the number of particles absorbed at cells 1 and 2K + 1, respectively.

To that end we treat the process as a Markov chain with two states, ℓ and r, corresponding to "most recent absorption was on the left" and "most recent absorption was on the right." The matrix of transition probabilities is either

$$\begin{array}{cc} & \begin{array}{c} \ell \\ \text{(to)} \end{array} \\ \begin{array}{c} \ell \\ \text{(from)} \end{array} & \frac{K}{K+1} \\ & \begin{array}{c} r \\ \text{(to)} \end{array} \\ & \frac{1}{K+1} \\ \begin{array}{c} r \\ \text{(from)} \end{array} & \frac{1}{K+1} \\ & \begin{array}{c} r \\ \text{(to)} \end{array} \\ & \frac{K}{K+1} \end{array}$$

with stationary probabilities $(\frac{1}{2}, \frac{1}{2})$ if $p = q = \frac{1}{2}$, or

$$\begin{array}{cc} 1 - \frac{1 - \frac{q}{p}}{1 - (\frac{q}{p})^{K+1}} & \frac{1 - \frac{q}{p}}{1 - (\frac{q}{p})^{K+1}} \\ \\ \frac{1 - \frac{p}{q}}{1 - (\frac{p}{q})^{K+1}} & 1 - \frac{1 - \frac{p}{q}}{1 - (\frac{p}{q})^{K+1}} \end{array}$$

with stationary probabilities $(\frac{p^K q^K - q^{2K}}{p^{2K} - q^{2K}}, \frac{p^{2K} - p^K q^K}{p^{2K} - q^{2K}})$ if $p \neq q$. (The transition probabilities are solutions to the classical gamblers' ruin problem.)

(N_ℓ, N_r) is the frequency count of a partial realization of length N of an ergodic Markov chain. Hence, as N increases, the random vector (N_ℓ, N_r) has asymptotically a normal distribution with mean either $(\frac{1}{2}, \frac{1}{2})$ if $p = q = \frac{1}{2}$, or

$$\left(\frac{p^{\frac{K}{2}} q^{\frac{K}{2}} - q^{\frac{2K}{2}}}{p^{\frac{2K}{2}} - q^{\frac{2K}{2}}} \cdot N, \frac{p^{\frac{2K}{2}} - p^{\frac{K}{2}} q^{\frac{K}{2}}}{p^{\frac{2K}{2}} - q^{\frac{2K}{2}}} \cdot N \right)$$

if $p \neq q$, and covariance matrix either

$$\begin{array}{cc} \frac{1}{4}NK & -\frac{1}{4}NK \\ -\frac{1}{4}NK & \frac{1}{4}NK \end{array}$$

if $p = q = \frac{1}{2}$, or

$$\begin{array}{cc} A & -A \\ -A & A \end{array}$$

where A equals

$$\frac{(p^{\frac{K}{2}} q^{\frac{K}{2}} - q^{\frac{2K}{2}})(p^{\frac{2K}{2}} - p^{\frac{K}{2}} q^{\frac{K}{2}})}{(p^{\frac{2K}{2}} - q^{\frac{2K}{2}})^2} N \left| \frac{q^{K+1} + pq^{\frac{K}{2}} - qp^{\frac{K}{2}} - p^{K+1}}{q^{K+1} - pq^{\frac{K}{2}} + qp^{\frac{K}{2}} - p^{K+1}} \right|$$

if $p \neq q$. (See Eq. (9) of Ref. 25.) The mean of (N_ℓ, N_r) is, in fact, exact.

In the nonconcurrent case, the matrix of transition probabilities is either

$$\begin{array}{cc} \frac{1}{2} & \frac{1}{2} \end{array}$$

$$\begin{array}{cc} \frac{1}{2} & \frac{1}{2} \end{array}$$

if $p = q = \frac{1}{2}$, or

$$\begin{array}{cc} 1 - B & B \\ 1 - B & B \end{array}$$

where B equals

$$\frac{1 - \left(\frac{q}{p}\right)^K}{1 - \left(\frac{q}{p}\right)^{2K}},$$

which is equal to

$$\frac{p^{\frac{K}{2}} q^{\frac{K}{2}} - q^{\frac{2K}{2}}}{p^{\frac{2K}{2}} - q^{\frac{2K}{2}}} \frac{p^{\frac{2K}{2}} - p^{\frac{K}{2}} q^{\frac{K}{2}}}{p^{\frac{2K}{2}} - q^{\frac{2K}{2}}}$$

$$\frac{p^{\frac{K}{2}} q^{\frac{K}{2}} - q^{\frac{2K}{2}}}{p^{\frac{2K}{2}} - q^{\frac{2K}{2}}} \frac{p^{\frac{2K}{2}} - p^{\frac{K}{2}} q^{\frac{K}{2}}}{p^{\frac{2K}{2}} - q^{\frac{2K}{2}}}$$

if $p \neq q$. The stationary probabilities are found to be the same as in the concurrent case; hence the mean of (N_ℓ, N_r) is the same. As N increases, the approach of (N_ℓ, N_r) to normality is that of $(X, N - X)$ where X is binomially distributed with parameters N and $p_{\ell, \ell}$ and the exact covariance matrix is

$$\begin{array}{cc} \frac{1}{4}N & - \frac{1}{4}N \\ - \frac{1}{4}N & \frac{1}{4}N \end{array}$$

if $p = q = \frac{1}{2}$, or

$$\begin{array}{cc} C & -C \\ -C & C \end{array}$$

where C equals

$$\frac{(p^K q^K - q^{2K})(p^{2K} - p^K q^K)}{(p^{2K} - q^{2K})^2} \cdot N$$

if $p \neq q$.

If one used Monte Carlo methods to estimate the stationary probabilities by the random vector $(N_\ell/N, N_r/N)$, in the no-drift case ($p = q = \frac{1}{2}$) one's estimates using concurrent random walks would have K times the variances as when using non-concurrent walks, for the same large number N of particles. Since with the concurrent method K times as many particles can be absorbed than with the non-concurrent method for the same large amount of random numbers, and, since the variances of the estimates are linear in $1/N$, the two methods are equally commendable if $p = q$, and the only desideratum is economy of random numbers. In the drift case ($p \neq q$), the variances of the estimates with the concurrent method differ from the variances with the non-concurrent method, by a factor of

$$\frac{|q^{K+1} + pq^K - qp^K - p^{K+1}|}{|q^{K+1} - pq^K + qp^K - p^{K+1}|}$$

As K increases, this factor approaches

$$\frac{p+q}{|p-q|} = \frac{1}{|p-q|}$$

Appendix C

COMPUTER PROGRAMS USED IN THIS RESEARCH

Sections 5.1.1 and 5.2.1 contain descriptions of the use of the programs included in this appendix. The programs appear as they stood at the last time they were translated. They were written in the MAD and UMAP languages and processed in The University of Michigan Executive System. Unless otherwise noted, all variables are of integer mode.

C.1 RDEP

for random walk simulation in spaces with irregular boundaries

```

$COMPILE MAD, PRINT OBJECT ,PUNCH OBJECT                                RDEP0000
    NORMAL MODE IS INTEGER
    POOLEAN SET, Q, SAV
    FLOATING POINT K
    FLOATING POINT KT , KS ,ES,E,ET
    DIMENSION A(30),TX(900),TAB(1060),B(900),MASK(30)
    IOHSIZ.(3)
    NUM=0

START    QA=0
          COUNT =0
          Q=0B
          SAV=0B
          AV=0
          ES=0.
          ET=0.
          KT=0
          KS =0
          PRINT COMMENT $1$
          READ AND PRINT DATA XMAX,YMAX,NUM,TIMES,REP,COUNT,QA
          FORMAT VARIABLE F1,F2,F3
             F1=YMAX-ORGY+1
             F2=XMAX
             F3=ORGX-1
          WHENEVER NUM.NE.0,RANDS.(NUM)
          NUM=0
          SET.(A,XMAX,YMAX,MASK,TAB,TX,TOT)
          INSERS.(ORGX,ORGY,A,XMAX,YMAX)
          WHENEVER COUNT .NE.0
             SAV=1B
             AV=TIMES/COUNT
          END OF CONDITIONAL
          WHENEVER QA.NE.0,Q=1B
          ZERO.(TAB(1)...TAB(900))
          MSK=1.LS.XMAX
          L=1
          J=0
          THROUGH B1,FOR I=1,1,I.G.YMAX
             MASK(I)=0
          THROUGH B2,FOR L=L,1,B(L).E.99
             J=J+1
             TAB((I.LS.5)+B(L))=J
             X=B(L)
B2        MASK(I)=MASK(I).V.(MSK.RS.X)
B1        L=L+1
          GT=J
          THROUGH B3, FOR I=YMAX,-1,I.L.1
B3        PRINT RESULTS TAB(I.LS.5+1)...TAB(I.LS.5+XMAX)
          PRINT FORMAT OUT,(I=YMAX,-1,I.L.1,MASK(I).V.1K11)
          TT=1
          TRANSFER TO JMP
LOOPA    TT=TT+1

```

```

JMP      WHENEVER TT.G.REPS,TRANSFER TO START
        ZERO.(A...A(YMAX),TX(1)...TX(900))
        COUNT =AV
        BR=0
        TIMESS=TIMES
        INSERP.(TIMESS)
        KX=0
        TOT=0
        T=0
LOOP     INSERT.(0)

        WHENEVER Q
        BR=BR+1
        WHENEVER BR.E.QA
        BR=0
        LINE=LINE+3+YMAX
        WHENEVER LINE.G.60
        PRINT COMMENT $1$
        LINE=YMAX+2
        END OF CONDITIONAL
        PRINT FORMAT OUT,(I=YMAX , -1,I.L.1,A(I).V.1K11)
        VECTOR VALUES OUT=$1H-/'F1'(1H ,2BI'F2'/),1H+,S'F3',1HS/(1H
1 ,2BI'F2')*$
        END OF CONDITIONAL
        END OF CONDITIONAL

        T=T+1
        KX=KX+TIMES-TIMESS-TOT
        TRANSFER TO S(RAND.(0))
S(0)    MVPX.
        TRANSFER TO RCD
S(1)    MVMX.
        TRANSFER TO RCD
S(2)    MVPY.
        TRANSFER TO RCD
S(3)    MVMY.
        TRANSFER TO RCD

RCD     WHENEVER .NOT.SAV
        WHENEVER TOT.L.TIMES, TRANSFER TO LOOP
        PRINT COMMENT $2$
        PR.
        TRANSFER TO LOOPA

        OTHERWISE
        WHENEVER TOT.L.COUNT, TRANSFER TO LOOP
        PRINT COMMENT $2$
        COUNT = COUNT+AV
        PR.
        PRINT COMMENT $1GRAND TOTALS$
        TOT=COUNT
        PR.
        TRANSFER TO LOOPA
        END OF CONDITIONAL

        INTERNAL FUNCTION
        ENTRY TO PR.
        E=KX/(T+0.)
        ET=ET+E
        ES=ES+E*E
        K=1.-(TOT+0.)/T
        KT=KT+K
        KS=KS+K*K
        PRINT RESULTS TOT,T,E ,K ,TX(1)...TX(GT)
        PRINT RESULTS KT,KS ,ES,ET
        RPUNCH.(TX(1)...TX(GT))
        FUNCTION RETURN
        END OF FUNCTION
        END OF PROGRAM

```


Input parameters:

XMAX maximum dimension along x-axis, $XMAX \leq 36$.
YMAX maximum dimension along y-axis, $YMAX \leq 30$.
NUM odd, 12-digit octal number to prime random number generator. If not supplied, the random numbers continue from the previous simulation.
TIMES corresponds to N in the text.
REPS corresponds to R_{xr} in the text.
B(1). . . the x coordinates of the absorbing points in row 1 of the array, followed by 99, followed by the x coordinates of the absorbing points in row 2, etc., followed by 99.
QA When used, the program will print an image of the simulation space once in every QA time steps.
ORGX x location of source.
ORGY y location of source.

Output parameters:

TX(i). . . the total number of absorptions at boundary point i; the ordering on the boundary points as defined in 4.2 (also punched in BPUNCH format).
KT sum of covering factors for all replications.
KS sum of squares of covering factors for all replications.
ET sum of average number of active particles for all replications.
ES sum of squares of average number of particles for all replications.
TOT total number of particles replicated.
T total number of time-steps required for all replications.

RDEP also prints an image of the absorption mask and a list of the indices of the absorbing points.

C.2 MOVE

Subroutine for use with RDEP

```

$ASSEMBLE,PUNCH OBJECT                                MOVE0000
    ENTRY SET
    ENTRY MVMX
    ENTRY MVPX
    ENTRY MVMY
    ENTRY MVPY
SET  CLA      1,4
    STA     LOOP
    STA     LOOPA
    STA     A
    STA     AA
    STA     AAA
    STA     B
    STA     LOOP5
    STA     MB
    STA     MC
    STA     ME

```

	SUB	=1
	STA	MA
	STA	MD
	STA	MF
	CLA*	2,4
	ALS	18
	STD	XTRA
	ARS	18
	STA	*+2
	CLA	=1
	ALS	**
	STO	CON
	CLA*	3,4
	STA	TST
	STA	MVMX
	STA	MVPX
	SUB	=1
	STA	MVPY
	ALS	18
	STD	STD
	CLA	4,4
	STA	MSK
	CLA	5,4
	STA	TB
	CLA	6,4
	STA	T1
	STA	T2
	CLA	7,4
	STA	T3
	STA	T4
	TRA	8,4
MVMX	AXT	**,1
LOOP	CLA	**,1
	ALS	1
AA	STO	**,1
	TIX	LOOP,1,1
	TRA	TST
MVPX	AXT	**,1
LOOPA	CLA	**,1
	ARS	1
AAA	STO	**,1
	TIX	LOOPA,1,1
	TRA	TST
MVMY	AXT	1,1
MA	CLA	**,1
MB	STO	**,1
	TXI	*+1,1,1
STD	TXL	MA,1,**
ME	STZ	**,1
	TRA	TST
MVPY	AXT	**,1
MC	CLA	**,1
MD	STO	**,1
	TIX	MC,1,1
MF	STZ	**
TST	AXT	**,1
LOOP5	CAL	**,1
MSK	ANA	**,1
	TZE	LOOP4
	SLW	TMP
	CLA	CON
	STO	TL
XMAX	AXT	1,2
LOOP3	CLA	TL
	ARS	1
	STO	TL
	ANA	TMP
	TZE	LOOP2
A	ERA	**,1

```

B      SLW      **,1
      SXD      SV,2
      PXA      0,1
      ALS      5
      PAX      0,2
SV     TXI      **+1,2,**
TB     CLA      **,2
      PAX      0,2
T1     CLA      **,2
      ADD      =1
T2     STO      **,2
T3     CLA      **
      ADD      =1
T4     STO      **
      LXD      SV,2
LOOP2  TXI      **+1,2,1
XTRA   TXL      LOOP3,2,**
LOOP4  TIX      LOOP5,1,1
      TRA      1,4
TL     PZE
TMP    PZE
CON    PZE
      END

```

C.3 ISRR

Subroutine for use with RDEP

```

$ASSEMBLE, PUNCH OBJECT                                ISRR000
      ENTRY    INSERTS
      ENTRY    INSERTP
      ENTRY    INSERT
INSERTP CLA      1,4
      STA      ID
      STA      IC
      CLA      CLA
      STO      INSERT
      TRA      2,4
INSERTS CLA*     4,4
      SUB*     1,4
      STA      **+2
      CAL      =1
      ALS      **
      SLW      CON
      CLA      3,4
      SUB*     2,4
      STA      CLA
      STA      IB
      TRA      6,4
INSERT  CLA      **
      ANA      CON
      TNZ      2,4
      CAL      CON
IB      ORS      **
IC      CLA      **
      SUB      =1
ID      STO      **
      TNZ      2,4
      CLA      **+2
      STO      INSERT
      TRA      2,4
CON     PZE
CLA     CLA
      END

```

C.4 NDEP

for random walk simulation in square spaces

```

$COMPILE MAD, PRINT OBJECT, PUNCH OBJECT                                NDEP0000
NORMAL MODE IS INTEGER
POOLEAN SET, Q, SAV
FLOATING POINT K
FLOATING POINT KT , KS
DIMENSION A(30),SXM(30),SXX(30),SYM(30),SYX(30),TYMIN(30),
1 TYMAX(30),TXMIN(30),TXMAX(30)
IOHSIZ.(3)
NUM=0

START      QA=0
           COUNT =0
           Q=0B
           SAV=0B
           AV=0
           W'R TT.G.REPS , PRINT RESULTS KT,KS
           KT=0
           KS =0
           PRINT COMMENT $1$
           READ AND PRINT DATA XMAX,YMAX,NUM,TIMES,REP,COUNT,QA
           MAX=YMAX
           WHENEVER XMAX.G.YMAX,MAX=XMAX
           FORMAT VARIABLE F1,F2,F3
           F1=YMAX-ORGY-1
           F2=XMAX-2
           F3=ORGX-2
           WHENEVER NUM.NE.0,RANDS.(NUM)
           NUM=0
           MVXS.(A,XMAX,YMAX)
           MVYS.(A,XMAX,YMAX)
           INSERS.(ORGX,ORGY,A,XMAX,YMAX)
           WHENEVER COUNT .NE.0
           SAV=1B
           AV=TIMES/COUNT
           END OF CONDITIONAL
           WHENEVER QA.NE.0,Q=1B
           TT=1
           TRANSFER TO JMP

LOOPA     TT=TT+1
JMP      WHENEVER TT.G.REPS,TRANSFER TO START
           ZERO.(A...A(YMAX))
           ZERO.(SYM...SYM(MAX),SXM...SXM(MAX),SXX...SXX(MAX),SYX...SYX
1 (MAX),TYMIN...TYMIN(MAX),TXMIN...TXMIN(MAX),TYMAX...TYMAX(MA
2 X),TXMAX...TXMAX(MAX),RMX,RMY,RPX,RPY,TOT,LINE,T)
           COUNT =AV
           BR=0
           TIMESS=TIMES
           INSERP.(TIMESS)

LOOP      TOT=0
           INSERT.(0)

           WHENEVER Q
           BR=BR+1
           WHENEVER BR.E.QA
           BR=0
           LINE=LINE+1+YMAX
           WHENEVER LINE.G.60
           PRINT COMMENT $1$
           LINE=YMAX+2
           END OF CONDITIONAL
           PRINT FORMAT OUT,(I=YMAX-2,-1,I.L.1,A(I),V.1K11)
           VECTOR VALUES OUT=$1H-/'F1'(1H ,2BI'F2'/),1H+,S'F3',1HS/(1H
1 ,2BI'F2')*$
           END OF CONDITIONAL
           END OF CONDITIONAL

```

```

T=T+1
TRANSFER TO S(RAND.(0))
S(0) Y=MVPX.(0)
WHENEVER Y.NE.0
    TXMAX(Y)=TXMAX(Y)+1
    TOT=TOT+1
    SET=1B
    TRANSFER TO S(0)
OTHERWISE
    RPX=RPX+1
    WHENEVER SET,TRANSFER TO RECORD
    TRANSFER TO LOOP
END OF CONDITIONAL
S(1) Y=MVMX.(0)
WHENEVER Y.NE.0
    TXMIN(Y)=TXMIN(Y)+1
    SET=1B
    TOT=TOT+1
    TRANSFER TO S(1)
OTHERWISE
    RMX=RMX+1
    WHENEVER SET, TRANSFER TO RECORD
    TRANSFER TO LOOP
END OF CO'NDITIONAL
S(2) Y=MVPY.(0)
WHENEVER Y.NE.0
    TYMAX(Y)=TYMAX(Y)+1
    TOT=TOT+1
    SET=1B
    TRANSFER TO S(2)
OTHER WISE
    RPY=RPY+1
    WHENEVER SET, TRANSFER TO RECORD
    TRANSFER TO LOOP
END OF CONDITIONAL
S(3) Y=MVMY.(0)
WHENEVER Y.NE.0
    TYMIN(Y)=TYMIN(Y)+1
    TOT=TOT+1
    SET=1B
    TRANSFER TO S(3)
OTHERWISE
    RMY=RMY+1
    WHENEVER SET, TRANSFER TO RECORD
    TRANSFER TO LOOP
END OF CONDITIONAL

RECORD SET=0B
WHENEVER .NOT.SAV
WHENEVER TOT.L.TIMES, TRANSFER TO LOOP
PRINT COMMENT $2$
PR.
TRANSFER TO LOOPA

OTHERWISE
WHENEVER TOT.L.COUNT, TRANSFER TO LOOP
PRINT COMMENT $2$
COUNT = COUNT+AV
PR.
THROUGH RS, FOR I=1,1,I.G.MAX
SYM(I)=SYM(I)+TYMIN(I)
SYX(I)=SYX(I)+TYMAX(I)
SXM(I)=SXM(I)+TXMIN(I)
SXX(I)=SXX(I)+TXMAX(I)
RS ZERO.(TYMIN...TYMIN(MAX),TYMAX...TYMAX(MAX),TXMIN...TXMIN(MAX)
1 ),TXMAX...TXMAX(MAX))
WHENEVER TOT.L.TIMES,TRANSFER TO LOOP
MOVER.(SYM...SYM(MAX),TYMIN...TYMIN(MAX),SYX...SYX(MAX),TYMAX

```

```

1 ...TYMAX(MAX),SXM...SXM(MAX),TXMIN...TXMIN(MAX),SXX...SXX(MA
2 X),TXMAX...TXMAX(MAX))
PRINT COMMENT $1GRAND TOTALS$
TOT=COUNT
PR.
TRANSFER TO LOOPA
END OF CONDITIONAL

INTERNAL FUNCTION
ENTRY TO PR.
K=1.-(TOT+0.)/T
KT = KT + K
KS = KS + K*K
PRINT RESULTS TOT,T,K,TXMIN(1)...TXMIN(YMAX),TYMAX(1)...TYMAX
1 (XMAX),TXMAX(YMAX)...TXMAX(1),TYMIN(XMAX)...TYMIN(1)
2 ,RMX,RPX,RYM,RPY
BPUNCH .( TXMIN(2)...TXMIN(YMAX-1),TYMAX(2)...TYMA
1 X(XMAX-1),TXMAX(YMAX-1)...TXMAX(2),TYMIN(XMAX-1)...TYMIN(2))
FUNCTION RETURN
END OF FUNCTION
END OF PROGRAM

```

Input parameters

XMAX	as in RDEP
YMAX	as in RDEP
NUM	as in RDEP
TIMES	as in RDEP
REPS	as in RDEP
QA	as in RDEP
ORGX	as in RDEP
ORGY	as in RDEP

Output parameters:

TXMIN(i)	number of absorptions at point i on left-hand border
TXMAX(i)	number of absorptions at point i on right-hand border
TYMIN(i)	number of absorptions at point i on bottom border
TYMAX(i)	number of absorptions at point i on top border

The above data all also punched according to BPUNCH format.

KT	as in RDEP
KS	as in RDEP
TOT	as in RDEP
T	as in RDEP

C.5 MVX

Subroutine called by NDEP

\$ASSEMBLE,PUNCHOBJECT		MVX00000
	ENTRY	MVMX
	ENTRY	MVPX
	ENTRY	MVXS
MVXS	CLA	1,4
	STA	LOOP

```

        STA     LOOPA
        STA     AA
        STA     AAA
        STA     OUT+1
        CLA*    3,4
        SUB     =2
        STA     MVPX
        STA     MVMX
        STA     MVA
        STA     MV
        CLA*    2,4
        SUB     =3
        STA     **+2
        CLA     =2
        ALS     **
        STO     CON
        TRA     4,4
MVMX   AXT     **,1
LOOP   CLA     **,1
        ALS     1
AA     STO     **,1
        ANA     CON
        TNZ     OUT
        TIX     LOOP,1,1
MVA    AXT     **,1
        SXA     MVMX,1
        ZAC
OUT    TRA     2,4
        COM
        ANS     **,1
        TXI     **+1,1,-1
        SXA     MVMX,1
        PXA     0,1
        ADD     =2
        TRA     2,4
CON    PZE
MVPX   AXT     **,1
LOOPA  CLA     **,1
        LGR     1
AAA    STO     **,1
        TQP     **+2
        TRA     OUTA
        TIX     LOOPA,1,1
MV     AXT     **,1
        SXA     MVPX,1
        ZAC
OUTA   TRA     2,4
        TXI     **+1,1,-1
        SXA     MVPX,1
        PXA     0,1
        ADD     =2
        TRA     2,4
        END

```

C.6 MVY

Subroutine called by NDEP

```

$ASSEMBLE, PUNCH OBJECT
        ENTRY   MVYS
        ENTRY   MVPY
        ENTRY   MVMY
MVYS   CLA     1,4
        STA     AA
        STA     AB
        STA     ABB
        ADD     =1
        STA     AMB

```

MVY00000

	STA	AM
	SUB	=2
	STA	AAB
	CLA*	2,4
	STO	MAX
	SUB	=2
	STA	MXA
	STA	*+2
	CLA	=1
	ALS	**
	ARS	1
	STO	CON
	CLA*	3,4
	SUB	=2
	STA	MX
	ADD	=1
	ALS	18
	STD	STD
	CLA	NOP
	STO	MVPY
	STO	MVMY
	TRA	4,4
MVPY	NOP	
MX	AXT	** ,1
AA	CLA	** ,1
	STO	TST
AM	CLA	** ,1
AB	STO	** ,1
	TIX	*-2,1,1
	CLA	TST
	TZE	2,4
	CLA	TR
	STO	MVPY
TEST	CAL	CON
	SLW	CONA
MXA	AXT	** ,1
	TRA	*+2
MXB	AXT	** ,1
	CLA	CONA
	LDI	TST
	TIO	UP
	ARS	1
	TIX	*-2,1,1
	CLA	NOP
	STO	MVPY
	STO	MVMY
	ZAC	
	TRA	2,4
UP	ARS	1
	TNZ	*+2
	CLA	=K1K11
	STO	CONA
	PXA	0,1
	CHS	
	ADD	MAX
	TXI	*+1,1,-1
	SXA	MXB,1
	TRA	2,4
TR	TRA	MXB
NOP	NOP	
CON	PZE	
TST	PZE	
CONA	PZE	
MAX	PZE	
MVMY	NOP	
	AXT	1,1
AAR	CLA	**
	STO	TST
TXI	TXI	*+1,1,1


```

ABB   CLA   **,1
AMB   STO   **,1
STD   TXL   TXI,1,**
      CLA   TST
      TZE   2,4
      CLA   TR
      STO   MVMY
      TRA   TEST
      END

```

C.7 INSRA

Subroutine called by NDEP

```

$ASSEMBLE, PUNCH OBJECT                                INSRA000
      ENTRY  INSERS
      ENTRY  INSERP
      ENTRY  INSERT
INSERP CLA   1,4
      STA   ID
      STA   IC
      CLA   CLA
      STO   INSERT
      TRA   2,4
INSERS CLA*  4,4
      SUB*  1,4
      STA   **+2
      CAL   =1
      ALS   **
      ARS   1
      SLW   CON
      CLA   3,4
      SUB*  2,4
      ADD   =1
      STA   CLA
      STA   IB
      TRA   6,4
INSERT CLA   **
      ANA   CON
      TNZ   2,4
      CAL   CON
IB     ORS   **
IC     CLA   **
      SUB   =1
ID     STO   **
      TNZ   2,4
      CLA   **+2
      STO   INSERT
      TRA   2,4
CON    PZE
CLA    CLA
      END

```

C.8 RNCSA

random number generator called by NDEP and RDEP, returns a random number between 0 and 3 inclusive

```

$ASSEMBLE ,PUNCH OBJECT                                RNCAS000
      ENTRY  RAND
      ENTRY  RANDS
RANDS  CLA*  1,4
      STO  NUM
      SXA  SV4,4
      CALL ,PRINT
      FMT  FORM
      ENDIO
SV4    AXT  **,4
      TRA  2,4
FORM   BCI  *,H* METHOD 7-23-63**

```

```

RAND  LDQ NUM
      MPY CON
      STQ NUM
      XCA
      CAS =K2K11
      TRA GR
      TRA GR
      CAS =K1K11
      TRA ONE
      TRA ONE
ZERO  ZAC
      TRA 2,4
ONE   CLA =1
      TRA 2,4
GR    CAS =K3K11
      TRA TWO
      TRA TWO
      CLA =3
      TRA 2,4
TWO  CLA =2
      TRA 2,4
NUM  PZE
CON  OCT 343277244615
      END

```

C.9 ECOMP

computes E, k, P for any rectangular space

```

$COMPILE MAD, PRINT OBJECT, PUNCH OBJECT                                ECOMP000
      VECTOR VALUES PI=3.14159265
      INTEGER M,N,R,P,Q
      DIMENSION SPR(5000),C1(5000),C2(5000),B(5000)

      INTERNAL FUNCTION SINH.(X)=(EXP.(X)-EXP.(-X))/2.

START  FTRAP.
      READ AND PRINT DATA A,B,N,M
      M=M-2
      N=N-2
      A=A-1
      B=B-1
      PIM=PI/(M+1.)
      NP=N+1.
      THROUGH SS, FOR R=1,1,R.G.M
          T=2.-COS.(R*PIM)
          B(R)=ELOG.(T+SQRT.(T*T-1.))
          AR= SIN.(A*R*PIM)
          WHENEVER .ABS.(NP*B(R)).G. 87.2
              C1(R)=AR/(FXP.(2.*B(R))-1.)
              C2(R)=0.
          OTHERWISE
              C =  SINH.(B(R))*SINH.(NP*B(R))
              C1(R)=AR*SINH.((NP-B)*B(R))/C
              C2(R)=AR*SINH.(B*B(R))/C
SS     END OF CONDITIONAL

      V=0.
      THROUGH S, FOR P=1,1,P.G.M
          THROUGH S1, FOR R=1,1,R.G.M
              SPR(R)=SIN.(P*R*PIM)
              THROUGH S, FOR Q=1,1,Q.G.N
                  SUM=0.
                  WHENEVER Q.LE.B
                      THROUGH SA, FOR R=1,1,R.G.M
                          WHENEVER C2(R).E.0.
                              T=1-Q-B
                              W'R T.L. 87.3
                              SUM= SUM+SPR(R)*C1(R)*EXP.(((1+Q-B)*B(R))

```

```

O'E
  SUM=SUM+SPR(R)*C1(R)*(EXP.((1+Q-B)*B(R))-FXP.(T*B(R)))
E'L
OTHERWISE
  SUM=SUM+SPR(R)*C1(R)*SINH.(Q*B(R))
SA  END OF CONDITIONAL
      OTHERWISE
      THROUGH SB, FOR R=1,1,R.G.M
WHENEVER C2(R).E.0.
  T=1-Q-B
  W'R T.L. 87.3
  SUM= SUM+SPR(R)*C1(R)*EXP.((1+B-Q)*B(R))
O'E
  SUM=SUM+SPR(R)*C1(R)*(EXP.((1+B-Q)*B(R))-FXP.(T*B(R)))
E'L
OTHERWISE
SB  SUM=SUM+SPR(R)*C2(R)*SINH.((NP-Q)*B(R))
      END OF CONDITIONAL
      END OF CONDITIONAL
SUM  =SUM*8./(M+1.)
S  WHENEVER P.E.A.AND.Q.E.B,WAB= SUM
      V=W+SUM

K =(WAB-1.) /WAB
E = W*(1./WAB)
PRINT RESULTS W ,K,E
TRANSFER TO START
END OF PROGRAM

```

Input parameters:

M	maximum x dimension.
N	maximum y dimension.
A	x-coordinate of source.
B	y-coordinate of source.

Output parameters:

E	as in text.
W	P in text.
K	k in text.

C.10 PROBS

determines absorption probabilities for any rectangular spaces

```

$COMPILE MAD, PUNCH OBJECT, PRINT OBJECT                                PROBS000
R PROBABILITY OF BORDER TERMINATION GIVEN DIMENSIONS OF RECT-
R ANGLE AND COORDINATES OF SOURCE
DIMENSION TOP(100),BOT(100),LH(100),RH(100)
DIMENSION CON(100)
INTEGER M,N,A,B,BNBCD.,R,P
INTEGER Y
VECTOR VALUES PI=3.14159265
R
INTERNAL FUNCTION
ENTRY TO SETUP.
MP=M+1.
NP=N+1.
PIM=PI/MP
THROUGH S1, FOR R=1,1,R.G.M
  T=2.-COS.(R*PIM)
B1=ELOG.(T+SQRT.(T*T-1.))
WHENEVER .ABS.(NP*B1).G.87.2
  CON(R)=SIN.(A*R*PIM)*EXP.(-B*B1)
OTHERWISE
  CON(R)=SIN.(A*R*PIM)*SINH.((NP-B)*B1)/SINH.(NP*B1)

```

```

S1      END OF CONDITIONAL
        FUNCTION RETURN
        END OF FUNCTION
R
INTERNAL FUNCTION SINH.(X)=(EXP.(X)-EXP.(-X))/2.
R
INTERNAL FUNCTION(X)
ENTRY TO COMP.
THROUGH S2, FOR P=1,1,P.G.M
SUM=0.
    THROUGH S3, FOR R=1,1,R.G.M
S3      SUM=SUM+CON(R)*SIN.(P*R*PIM)
        WHENEVER SUM.LE.0.,SUM=0.
S2      X(P)=(SUM*2.)/MP
        FUNCTION RETURN
        END OF FUNCTION
R
START   FTRAP.
        PRINT COMMENT $1$
        READ AND PRINT DATA M,N,A,B
        M=M-2
        N=N-2
        A=A-1
        B=B-1
        EXECUTE SETUP.
        EXECUTE COMP.(LH)
        WHENEVER B.E.N+1-B
S4      THROUGH S4, FOR R=1,1,R.G.M
        RH(R)=LH(R)
        OTHERWISE
        B=N+1-B
        EXECUTE SETUP.
        EXECUTE COMP.(RH)
        B=N+1-B
        END OF CONDITIONAL
        Y=M
        M=N
        N=Y
        Y=A
        A=B
        B=Y
        EXECUTE SETUP.
        EXECUTE COMP.(TOP)
        WHENEVER B.E.N+1-B
S5      THROUGH S5, FOR R=1,1,R.G.M
        BOT(R)=TOP(R)
        OTHERWISE
        B=N+1-B
        EXECUTE SETUP.
        EXECUTE COMP.(BOT)
        END OF CONDITIONAL
        Y=M
        M=N
        N=Y
        OUTA(2)=BNBCD.(7*N)
        WHENEVER N.G.14,OUTA(2)=BNBCD.(98)
        PRINT FORMAT OUT, TOP(1)...TOP(N)
        PRINT FORMAT OUT,(R=1,1,R.G.M,LH(R),RH(R))
        PRINT FORMAT OUT,BOT(1)...BOT(N)
        TRANSFER TO START
        VECTOR VALUES OUTA=$1H0,F5.4,S          ,S2,F5.4*$
        VECTOR VALUES OUT=$1H0, //S6,16(S2,F5.4)*$
        END OF PROGRAM

```

Input parameters:

Same as in ECOMP.

Output parameters:

Graphical printout of absorption probabilities, with the exception that for spaces with $M > 14$ the top and bottom rows are printed on more than one line and the right side is foreshortened.

C.11 COVAN

```
$COMPILE MAD, PRINT OBJCT , PUNCH OBJECT          COVAN000
  DIMENSION  FORMAT(20)
  DIMENSION L(200)
  DIMENSION COM(3)
  VECTOR VALUES QQ0003 = 2,1,100
  DIMENSION COV(10000,QQ0003),R(10000,QQ0003),D(100),B(100)
  INTEGER B
  EXECUTE ZERO.(P...P(100))
  TRANSFER TO UP
QQ0002  READ FORMAT SETUP,N,IFWT,NFMT,CT,COM(1)...COM(3)
        INIT.
        SETEOF.(QQ0026)
        VECTOR VALUES SETUP=$S8,3I8,F8.0,S6,3C6*$
        WHENEVER N.G.100, TRANSFER TO QQ0006
        TRANSFER TO QQ0007
QQ0006  PRINT FORMAT QQ0008,N,COM(1)...COM(3)
        VECTOR VALUES  QQ0008 = $ 54H4      THIS PROGRAM WILL HANDL
        IF A MAX. OF 100 VARIABLES./31H0     YOU ARE ATTEMPTING TO RUN
        I13,22H VARIABLES ON RUN           3C6*$
        EXECUTE SYSTEM .( 0)
QQ0007  N1=N+1
        WHENEVER NFMT .NE. 0, TRANSFER TO QQ0010
QQ0009  N2 = QQ0011
        STATEMENT LABEL N2
        TRANSFER TO QQ0012
QQ0010  N2 = QQ0013
        STATEMENT LABEL N2
QQ0012  THROUGH QQ0014 ,FOR I      = 1      ,1, I      .G.
        IN1
        THROUGH QQ0014 ,FOR J      = 1      ,1, J      .G.
        1I
QQ0014  R(I,J)=0.0
        WT=1.0
        WTSUM=0.0
        NCC=0
        D(N1)=1.0
        W/R NFMT .E. 0, T'O QQ2
        NFMT=NFMT*12
        READ FORMAT QQ0015,FORMAT(1)...FORMAT(NFMT)
        VECTOR VALUES  QQ0015 = $ 12C6  *$
QQ2     SETERR.(Q)2A)
        LOOK AT FORMAT QQ3, ITYPE
        VECTOR VALUES QQ3=$C1*$
        WHENEVER ITYPE.E.-$=$,TRANSFER TO QQ0026
QQ2A    SETERR.(0)
        TRANSFER TO N2
        QQ0013  READ FORMAT FORMAT(1),D(1)...D(N)
        TRANSFER TO QQ0025
QQ0011  BREAD.(B(1)...B(N))
        THROUGH QQ11A, FOR B=1,1,B.G.N
QQ11A   D(B)=B(B)
QQ0025  NCC=NCC+1
        WTSUM=WTSUM+WT
        THROUGH QQ0027 ,FOR I      = 1      ,1, I      .G.
        IN1
        THROUGH QQ0027 ,FOR J      = 1      ,1, J      .G.
        1I
QQ0027  F(I,J)=R(I,J)+D(I)*D(J)*WT
        TRANSFER TO QQ2
```

```

QQ0026      NWT=WTSUM
            PRINT FORMAT SETOUT,COM(1)...COM(3) ,N,NCC,NWT
            VECTORVALUES SETOUT=$1H4,3C6/1H0,S7,19HNO. OF VARIABLES = I5/
1           1H0,S4,22HNO. OF OBSERVATIONS = I5/1H0,S13,13HWEIGHT SUM =
2           I5/1H2,S4,H*VAR NO.*,S8,H*SUM*,S12,H*SUM SQUARES*,S11,H*VAR
3           IANCE*,S13,H*EXP*,S18,1HD,S9,H*PROB TH PROB**$
            FN1=WTSUM-1.
            THROUGH      QQ0029 ,FOR I          = 1          ,1, I          .G.
1N
            THROUGH      QQ0029 ,FOR J          = 1          ,1, J          .G.
1I
QQ0029      COV(I,J)=(R(I,J)-R(N1,I)*R(N1,J)/WTSUM)/FN1

            QQS=0.
            SUM=0.
            P=0.
            WHENEVER P(1).E.0,SPRAY.(0,P(1)...P(N))
            THROUGH      QQ0030 ,FOR I          = 1          ,1, I          .G.
1N
            SS=R(I,I)
            AVE=R(N1,I)/WTSUM
            D(I)=SQRT      .(          .ABS          .( SS/WTSUM-AVE*AVE))
            THROUGH      QQ0031 ,FOR J          = 1          ,1, J          .G.
1I
QQ0031      F(I,J)=((R(I,J)-AVE*R(N1,J))/WTSUM)/(D(I)*D(J))
            E=D(I)*D(I)*WTSUM/(WTSUM-1.)
            AV=AVE/CT
            P=P+AV
            SUM=SUM+R(N1,I)
            WHENEVER P(I).E.0.,P(I)=AV
            PR=P(I)*(1.-P(I))*CT
            D=E/PR
            QQSS=SETN.(0)
            INTEGER SETN.
            STA(QQSS)=STA(QQSS)+D
            ST(QQSS)=ST(QQSS)+D*D
            STN(QQSS)=STN(QQSS)+1
            INTEGER QQSS
            QQS=QQS+D
            L(I)=D
QQ0030      PRINT FORMAT QQ1,I,R(N1,I),SS,E,PR,D,AV,P(I)
            VECTOR VALUES QQ1=$1H0,I9,1P5E19.7,2F8.4*$
            PRINT FORMAT QQ4,SUM,QQS/N,P
            VECTOR VALUES QQ4=$1H-,S4,H*TOTAL*,1PE19.7,S57,1PF19.7,F8.4*$

            INTEGER      N      ,      N1      ,      IFWT      ,      I
            INTEGER      J      ,      NCC      ,      NFMT      ,      QQ0017
            INTEGER      QQ0020,      QQ0024,      ITYPE      ,      NWT
            INTEGER      MXOUT ,      NXOUT

            SETEOF.(0)
UPA
            SS=0.
            MN=0.
UP
            READ FORMAT IN,OP,NI,A(1)...A(NI)
            WHENEVER OP.E.$PROB$
                READ AND PRINT DATA P(1)...
            OR WHENEVER OP.E.$CORR$
                EXECUTE MXOUT.(R,N)
            PRINT COMMENT $1$
            OR WHENEVER OP.E.$COVS$
                EXECUTE NXOUT.(COV,N)
            PRINT COMMENT $1$
            OR WHENEVER OP.E.$SUM$
                P=0.
                THROUGH S, FOR I=1,1,I.G.NI
                    P=P+P(A(I))
                    EXP=P*(1.-P)*CT
                    WHENEVER EXP.E.0.,EXP=1.
                    SUM=0.
S

```

```

        THROUGH S1, FOR I=1,1,I.G.NI
        THROUGH S1, FOR J=1,1,J.G.I
        WHENEVER I.E.J
            SUM=SUM+COV(A(I),A(I))
        OR WHENEVER A(I).L.A(J)
            SUM=SUM+ 2.*COV(A(J),A(I))
        OTHERWISE
            SUM=SUM+2.*COV(A(I),A(J))
S1      END OF CONDITIONAL
        D=SUM/EXP
        NN=NN+1.
        PRINT RESULTS SUM,EXP,D,A(1)...A(NI)
        SS=SUM+SS
        ORWHENEVER OP.E.$DPCH$
            BPUNCH.(L(1)...L(N))
        OR WHENEVER OP.E.$AVG$
            AV=SS/N.I
            D=AV/EXP
        PRINT COMMENT      $0*****$
        PRINT RESULTS AV,EXP,D
        PRINT COMMENT      $ *****$
        OR WHENEVER OP.E.$ERASE$
            TRANSFER TO UPA
        OR WHENEVER OP.E.$DATA$
            TRANSFER TO QQ0002
        O'R OP.E.$DC$
            ST=0.
            STA=0.
            THROUGH SDC, FOR I=1,1,I.G.PS
                STA(I)=STA(I)/STN(I)
                ST(I)=(ST(I)-STA(I)*STA(I)*STN(I))/(STN(I)-1)
                STA=STA + STA(I)
SDC     ST=ST+ST(I)
            STA=STA/PS
            ST=ST/PS
        PRINT COMMENTS-ST IS D, STA IS AV ... BOTH AVERAGED AND BY
1      POINTS
        PRINT RESULTS ST...ST(PS),STA...STA(PS)
        END OF CONDITIONAL
        TRANSFER TO UP
        DIMENSION A(100),P(100)
        VECTOR VALUES IN =$ S1,C6,S3,I5,15I3/(S1,23I3)*$
        INTEGER A,NI,OP

INTERNAL FUNCTION
ENTRY TO SETN.
T'O SETL(SM)
SETL(1) SI=SI+1
        W'R SI.GE.PS
            W'R EQ,SM=2
            W'R SI.E.PS+1
                SI=SI-1
                SM=2
        E'L
        E'L
        FUNCTION RETURN SI
SETL(2) SI=SI-1
        W'RSI.E.0
            SI=1
            SM=1
        F'L
        FUNCTION RETURN SI
        ENTRY TO INIT.
        SI=0
        FQ=0B
        SM=1
        PS=N/8

```

```

V'R PS*8,NE,N
  EQ=1B
  PS=PS+1
F'L
ZERO.(ST...ST(20),STN...STN(20),STA...STA(20))
FUNCTION RETURN
END OF FUNCTION
INTEGER SI,PS,SM,STN
DIMENSION STN(20),STA(20),ST(20)
BOOLEAN EQ

END OF PROGRAM

```

Control parameters:

\$DATA followed by

1. control card
column 16, 1.
columns 17-24, the number of following cards containing format information.
columns 25-32, N for these data.
columns 47-64, a comment to precede the output for these data.
2. format card
none for binary data,
otherwise prepared according to rules for MAD I-O formats.
3. data cards
according to format defined in 2, or punched as column binary using BPUNCH in the U. of M. system library.

\$CORR

prints out the correlation matrix for the last preceding set of data.

\$COV

prints out the covariance matrix for the last preceding set of data.

\$DC

computes \bar{D} and $\sigma^2(\bar{D})$ for square spaces.

\$DPCH

punches the D vector in BPUNCH format.

\$PROB

followed by one or more cards in MAD "Simple I-O" format containing absorption probabilities. If $P(1) = 0$, then the experimentally estimated probabilities are used.

\$SUM

followed on the same card by a number punched columns 13-15, the number of points to be read in to define one target set; every following set (up to 15) of 3 columns containing the numbers (see RDEP and NDEP) of the points in the set; followed by as many cards as necessary to contain additional points in the target set punched in 3 column fields after skipping the first column.

\$AVG

averages the values for the target sets in the preceding \$SUM cards following the last occurrence of an

\$ERASE card.

Output parameters:

\$DATA produces the number of absorbing points on the boundary, the number of replications in the data read in, a table of the sum, sum of squares, variance V, expected variance EXP, $D = V/EXP$, estimated absorption probability, true absorption probability for each point, and the average of the D's, and the sum of the sums and estimated probability.

\$CORR and \$COV produce the lower triangular correlation and covariance matrices for the last data read in.

\$SUM produces the variance of the set of points in the target set, the expected variance and the resultant D.

\$AVG produces the same output as \$SUM for the preceding occurrences of \$SUM occurring before \$ERASE.

C.12 NXOUT

is used by COVAN to print lower triangular matrices

```
$COMPILE MAD, PRINT OBJECT , PUNCH OBJECT          NXOUT000
R      SUBROUTINE TO PRINT COVARIANCE  MATRIX IN MATRIX FORM
R
R
R
      EXTERNAL FUNCTION (R,N)
      ENTRY TO MXOUT.
      F1=18
      F2=6
      F3=2
      F4=0
      K=18
      TRANSFER TO QQ1
      ENTRY TO  NXOUT .
      K=9
      F1=9
      F2=12
      F3=0
      F4=1
QQ1    I=N/K
      FORMAT VARIABLE F1,F2,F3,F4
      INTEGER F1,F2,F3,F4
      WHENEVER (N-I*K) .G.0, TRANSFER TO QQ0003
      TRANSFER TO QQ0004
QQ0003 I=I+1
QQ0004 ITOTAL=(I*(I+1))/2
      !PAGE=0
      I1=1-K
QQ0005 !1=I1+K
      !2=I1+K-1
      WHENEVER (I2-N).G.0, TRANSFER TO QQ0006
      TRANSFER TO QQ0007
QQ0006 I2=N
QQ0007 J1=1-K
```

```

QQ0008      J1=J1+K
             J2=J1+K-1
             WHENEVER (J2-N).G.0, TRANSFER TO QQ0009
             TRANSFER TO QQ0010
QQ0009      J2=N
QQ0010      ?PAGE=IPAGE+1
             PRINT FORMAT QQ0011,IPAGE,ITOTAL,(J=J1,1,J.G.J2,J)
             VECTOR VALUES QQ0011=$'F4'(1H1,36(1H),47HC O V A R I A N C E
1            C O E F F I C I E N T S ,19(1H),9HPAGE NO. ,I2,4H OF
2            ,I2//12HOCOLUMN = ,9I12*),1H1,36(1H),47HC O R R E L A T I
4            O N C O E F F I C I E N T S ,19(1H),9HPAGE NO. ,I2,4H OF
5            ,I2//12HOCOLUMN = ,18I6*$
             WHENEVER (J2-I2) .L.0, TRANSFER TO QQ0014
             TRANSFER TO QQ0015
QQ0014      !SWTCH=2
             TRANSFER TO QQ0016
QQ0015      !SWTCH=1
QQ0016      THROUGH QQ0017 ,FOR I = I1 ,1, I .G.
1I2
             TRANSFER TO QQ0019( !SWTCH)
QQ0019(01) TRANSFER TO QQ0018
QQ0019(02) TRANSFER TO QQ0017
QQ0018      J2=I
QQ0017      PRINT FORMAT QQ0020,I,R(I,J1)...R(I,J2)
             VECTOR VALUES QQ0020=$1H /8H0 ROW = ,I3,1H , 'F1'( 'F2'.'F3')
1            *$
             TRANSFER TO QQ0025( !SWTCH)
QQ0025(01) TRANSFER TO QQ0024
QQ0025(02) TRANSFER TO QQ0008
QQ0024      WHENEVER (I2-N).L.0, TRANSFER TO QQ0005
QQ0026      FUNCTION RETURN
R            * * * * * T H E E N D * * * * *
R
INTEGER QQ0002, K , I , N
INTEGER ITOTAL, IPAGE , I1 , I2
INTEGER J1 , J2 , QQ0013, !SWTCH
INTEGER QQ0023
INTEG ER J
QQ0001      END OF FUNCTION

```

C.13 DANAL

computes averages and variances of D and G for irregular spaces

```

$COMPILE MAD, PRINT OBJECT ,EXECUTE ,DUMP DANAL000
INTEG ER J,I,N,R ,L
DIMENSION D(200),S(200),SS(200),V(200) ,L(200) ,U(200),W(200)

START      READ AND PRINT DATA N,R
            ZERO.(S...S(200) , SS...SS(200),L...L(200))
            ZERO.(U...U(200),W...W(200))
            THROUGH S1, FOR I=1,1,I.G.R
            BREAD.(D(1)...D(N-2))
            READ AND PRINT DATA E,D(N)
            THROUGH S1, FOR J=1,1,J.G.N
            WHENEVER D(J).NE.0.
            L(J)=L(J)+1
            T=E/D(J)
            S(J)=S(J)+T
            SS(J)=SS(J)+T*T
            U(J)=U(J)+D(J)
            W(J)=W(J)+D(J)*D(J)
S1         END OF CONDITIONAL
            THROUGH S2, FOR J=1,1,J.G.N
            U(J)=U(J)/L(J)
            S(J)=S(J)/L(J)
            W(J)=(W(J)-U(J)*U(J)*L(J))/(L(J)-1)

```

```

S2      V(J)=(SS(J)-S(J)*S(J)*L(J))/(L(J)-1)
        PRINT COMMENT $-S IS G,V IS V(G), L IS DF,U IS D, W IS V(D)$
        PRINT RESULTS S(1)...S(N),V(1)...V(N) ,L(1)...L(N)
        PRINT RESULTS U(1)...U(N),W(1)...W(N)
        TRANSFER TO START
        END OF PROGRAM

```

Input parameters

N the number of variables to be processed by replication

R the number of replications

D(1)...D(N - 2) the values of D_i for the $n(=N - 2)$ absorbing points (from COVAN)

E the experimental estimate of E

D(N - 1) $D = \frac{\sum D_i}{n}$

D(N) D_Σ

Output parameters:

5 tables, each one value per input variable

S_i mean of G_i

V_i $\sigma^2(G_i)$

L_i the number of non-zero D_i 's in S_i

U_i mean of D_i

W_i $\sigma^2(U_i)$

Appendix D

PHILOSOPHICAL AND PRACTICAL INSIGHTS INTO THE USE OF A COMPUTER AS A RE-SEARCH TOOL

This thesis is inspired in many ways by the concept of a computer. The problem concerns a class of hardware computers, a classical computer application, and the selection and evaluation of a computer algorithm. In addition, a computer is used as a tool in carrying out this research.

It is common to take advantage of many University facilities in performing research. The library is certainly applicable in almost every case. The experimental physicist may use a cyclotron, bubble chamber, or reactor. The linguist will often employ a tape recorder or spectrum analyzer. Almost every discipline, even if only scientific in a weak sense, appears to make some use of the digital computer in research.

One of the clearest examples of the advantage of computer use in research occurs in the factor analysis, a complicated statistical device used by behavioral scientists. Whereas, before the availability of computers one or two hand calculated factor analyses might constitute a doctoral dissertation, now such calculations are routine tasks for a computer. The important result is that the scholar is free to contribute to his discipline in a more creative and enlightening way.

The experimental work proposed for this investigation appears to be a case of the dual evil in use of a computer; that is, the work consisted of using a computer to generate data which were then analyzed on the same computer.

The use of a computer as a data generator has, however, become quite justifiable as the art and science of simulation have developed. Truly, any random walk run on a computer is a simulation, and it results in what may be described as computer generated data. But researchers in the various science and engineering disciplines have become quite adept at computer simulation for a large variety of systems; from those as small as the microcomponents of living cells, to the fuel tanks of the SATURN I space booster. Thus, this research is not involved with the use of the computer as a data generator, but as a simulator. The data reduction and analysis function in this work is by no means an unusual computer application.

Random walk simulations have only been fully realizable in the period since the development of the contemporary computer. The results of McCrea and Whipple (Ref. 14), which we can apply now in a matter of fractions of seconds, would have taken hours and days to use by hand with very little chance of error free computation at the time of their development. This point is brought out when one realizes that in Eq. 3.21 as N becomes large, $\sinh((N + 1)B)$ be-

comes very large, and the use of these results in a computer is apparently out of the question due to overflow problems. However, appropriate approximations can be made, allowing for general use of the classic results of McCrea and Whipple.

The computer programs involved in this research are included in Appendix C. However, some techniques and approaches which were used in this work will be discussed here.

With respect to the simulation, the program is of general iterative nature and written in high level (algebraic) language. When approaching the implementation of a highly iterative algorithm, decisions must be made as to what type of language to use. A flexible executive system allows a combination of language forms through the use of general subroutine linkage. With the advent of refined symbol manipulation operations in the language involved here, it is conceivable that the entire simulator could well have been written in that language. However, these new developments followed in time the construction of this program, and this was never considered.

When making decisions as to which language to use, the enormity of some numbers, especially the number of iterations required, should not overshadow the expense of using an assembly language. For example; in this problem one must replicate a 1500-particle random walk 200 times. Each particle replication requires approximately 1.5 basic time-steps. Thus, one is confronted with a simulation requiring 4.5×10^5 basic operations, and this simulation will be repeated for several geometries. A first thought might be to use assembly language and the accompanying ability to use index-registers, if available, in a most effective manner. However, consider the case in which using these high speed machine features can save only 1/100 of the total time involved in one iteration, which is the case here. The result is a saving of perhaps 6 seconds in 10 minutes of computation at the cost of many added "debugging" runs and an extended period of time for program checkout. It is probable that the total saving in this case might be negative.

On the other hand, there is required in each time step a spatial displacement of every particle in the grid and an examination of the boundary points for any possible absorptions. This also is an iterative process, and we are now confronted with an algorithm which itself accounts for over 50% of the time required to simulate a single time-step. Routines for simulation of the source node and for simulation of particle motion and absorption in each of the four directions were written in assembly language with appropriate calls occurring in the main, compiler language program. The total result is a working program which is efficient to use and yet efficient to debug.

Results are printed on an output sheet and also punched on densely packed binary cards for use in the analysis program, thereby reducing greatly the effort in linking the programs.

The analysis program is so general in scope that it was decided to maintain it and the simulation results separately. In this way the binary cards could be subjected to various analyses, not all of which could necessarily be determined at the time the simulation was carried out. The added effort and expense in performing these two tasks independently and maintaining a file of results on punched cards has proven well worthwhile.

One problem in carrying out the simulations was solved in what might be a novel, though very simple way. Generation of random values "North", "South", "East", and "West" (0, 1, 2 and 3) can be attempted in several ways using uniformly distributed psuedo-random number generators. However, some of these can be quite costly or even theoretically unsound. The problem is really one of obtaining a two-bit integer, but most available methods for generating random numbers yield a psuedo-random fraction in the range between 0 and 1. Selecting two bits in some systematic way from numbers of this type does not generally offer numbers which can be considered psuedo-random.

The method normally employed is to scale the fraction by multiplying, in this case by 4, and to use the rounded result. However, another scheme can be used which is more nearly optimum in terms of speed of operation. To obtain a psuedo-random integer between 0 and $n - 1$ from a psuedo-random number between 0 and 1, first generate a table of fractions equal to $1/n, 2/n, \dots, n - 1/n$. After obtaining the original number by some well-known technique such as the power residue method (Ref. 26), perform a tree-like series of transfers based on "greater-than-or-equal" comparisons, ending up with the loading of the appropriate integer constant and return to the calling program. In general, when n is small, this will result in an efficient program.

As an example, consider the case where $n = 4$. The MAD language code for this operation is as follows:

```
WHENEVER RNUM .LE. 2k10
    WHENEVER RNUM .LE. 1K10
        R = 0
    OTHERWISE
        R = 1
    END OF CONDITIONAL
OTHERWISE
    WHENEVER RNUM .LE. 3K10
        R = 2
    OTHERWISE
        R = 4
    END OF CONDITIONAL
END OF CONDITIONAL
```

In this program RNUM is the original full word value between 0 and 1 and the constants of the form $xK10$ are full word fractions representing $1/4$, $1/2$, and $3/4$. Note that this algorithm requires only two comparisons per number which amounts to a much faster execution than an integer multiplication.

It is worthwhile to mention the technique used to store the image of the random walk space for the simulation program. Borrowing an idea from Samuel's Checker Player (Ref. 27), the space, limited to 36 by 36, is stored in 36 or fewer words of memory, each word representing one row of the lattice and each bit position across words representing one column. As is customarily done, a 1 in any position represents the presence of a particle and a 0 represents an unoccupied grid point. Thus, a move of all particles to the right is accomplished by a corresponding shift of each word, and noting and erasing any bits where absorption has taken place. Motion in a vertical direction is accomplished by a cyclic replacement of words, filling in an all-zero word at the appropriate end of the chain and examining the last word at the opposite end for any absorptions.

No estimate has been made of the amount of time saved by using this programming trick, but it is probably considerable when compared with a more conventional approach. Note that only n basic operations are necessary for the simulation of one time step in an n by n grid. Compared to methods which record and update coordinates of all active particles, removing some, adding others, and performing bookkeeping for these lists, this general scheme appears to be conservative of both storage and time.

Several alternative methods were available for transmitting to the simulation program RDEP the information necessary to specify an irregular absorbing boundary. These ranged from using as data the coordinates of the points on the boundary, to merely reading in the matrix of random digits used to select some space (see Section 5.2.2). The method ultimately chosen for this purpose is to use as input a vector consisting of successive sets of column coordinates for each boundary in a row followed by "99", once for each row. The program then generates a mask for each row and prints out an image representing the absorbing boundary and the source location. This device necessitates a minimum of keypunching and yet retains complete generality and, further, allows use of the "simple" input routine available in the Michigan Executive System reducing potential format errors.

With respect to the analysis program, rather than have several programs to accomplish this task, depending on the nuances involved in a particular analysis, this program has been developed as a small executive system in which control cards, recognized by a "\$" in the first column, are used to transfer control to appropriate routines. Thus, the user selects the analyses to be performed in what seems to the user to be a very natural way. Possibly the most important concept discovered here for the author is that, rather than complicating his task, this

organization makes both the use and the programming of the routine quite straightforward.

The MAD language and very flexible and powerful executive and input-output systems available at The University of Michigan Computing Center have allowed the completion of this project in relatively short time. What might have been added frills in some other environments were built in as a matter of course in this set of programs. Thus, one can examine periodic pictorial printouts of the simulation space by setting a programmed data switch, the absorption probability computation pictorially prints its results, yielding a meaningful display of these values, and the analysis program reads control cards containing commands punched in English, not as digits. This research provides one illustration of the value of such a software system.

REFERENCES

1. Morrison, P., and Morrison, E., Charles Babbage and his Calculating Engines, Dover, New York, 1961, see especially p. 225.
2. Corbato, F., et al., The Compatible Time-Saving System, MIT Press, Cambridge, 1963.
3. von Neumann, J., "The Theory of Automata; Construction, Reproduction, Homogeneity," (to be published by U. of Ill. Press).
4. Holland, J., "Iterative Circuit Computers," Proc. WJCC, AFIPS 1960, pp. 259-265.
5. Squire, J., and Palais, S., "Programming and Design Consideration of a Highly Parallel Computer," Proc. SJCC, AFIPS 1963, pp. 395-400.
6. Gonzalez, R., "A Multilayer Iterative Circuit Computer," IEEE TEC, December 1963, pp. 781-790.
7. Comfort, W., "Highly Parallel Machines," in Barnum and Knapp eds., Workshop on Computer Organization, Spartan, Baltimore, 1963.
8. Slotnick, D., Borck, and Reynolds, "The Solomon Computer," Proc. EJCC, AFIPS 1962, pp. 97ff.
9. McCormick, B., and Divilbiss, "A Pattern Recognition Computer," Proc. Engineering Summer Conferences, The University of Michigan, Ann Arbor, Michigan, 1962.
10. Leondes and Rubinoff, "DINA, A Digital Analyzer for Laplace, Poisson, Diffusion, and Wave Equations," Trans. AIEE, 1952, pp. 303ff.
11. Todhunter, I., Mathematical Theory of Probability, Macmillan, Cambridge, 1865, and Chelsea Co., New York, 1949, Articles 34, 35 and 107.
12. Feller, W., An Introduction to Probability Theory and its Applications, Vol. 1, Wiley, New York, 1957, pp. 311-334.
13. Courant, Friederichs and Lewy, "Differenzgleichungender Mathematischen Physik," Math. Ann.; Vol. 6, 1938, pp. 22-74.
14. McCrea and Whipple, "Random Paths in Two and Three Dimensions," Proc. Roy. Soc. Edinburgh, Vol. 60, 1940, pp. 281-298.
15. Rosenstock, H. B., "Random Walks with Spontaneous Emission," J. SIAM, Vol. 9, No. 2, June 1961, pp. 169-188.
16. Levison, N., "Emission Probability in a Random Walk," J. SIAM, Vol. 10, No. 3, Sept. 1963, pp. 442-447.
17. Brown, G. W., "Monte Carlo Methods" in Beckenback ed., Modern Mathematics for Engineers, McGraw-Hill, New York, 1956, pp. 291-302.
18. Berger, M. J., "Monte Carlo Calculation of the Penetration and Diffusion of Fast Charged Particles," in Alder, ed., Methods in Computational Physics, Academic Press, New York, 1963, pp. 135-139.
19. Ibid., Wall, Widwer and Gans, "Monte Carlo Methods Applied to Configurations of Flexible Polymer Molecules," pp. 217-243.
20. Kemeny and Snell, Finite Markov Chains, Van Nostrand, New York, 1960.

21. From Morse and Kimball, Methods of Operation Research, Technology Press, Cambridge, Mass., 1951.
22. Forsythe and Wasow, Finite-difference Methods for Partial Differential Equations, Wiley, New York, 1960.
23. Milne, Numerical Solution of Differential Equations, Wiley, New York, 1960.
24. Rosin, R. F., "An Organization of an Associative Cryogenic Computer," SJCC, AFIPS, 1962.
25. Good, I. J., "The Frequency Count of a Markov Chain, etc.," Ann. Math. Stat., 1961, pp. 41-48.
26. IBM Corp., "Reference Manual; Random Number Generating and Testing," No. c20.8011, White Plains, New York, 1959.
27. Samuel, A. H., "Some Studies in Machine Learning Using the Game of Checkers," IBM J. Res. and Dev., Vol. 3, July 1959, pp. 211-219.

DISTRIBUTION LIST

- | | | |
|----|--|-----|
| 1. | John McLean, EMIF
RADC
Griffiss AFB
Rome, New York | (1) |
| 2. | Lt. L. W. Odell, EMICA
RADC
Griffiss AFB
Rome, New York | (2) |
| 3. | Pat Langendorf, EMIIT
RADC
Griffiss AFB
Rome, New York | (1) |
| 4. | Fred Dion, EMIIT
RADC
Griffiss AFB
Rome, New York | (1) |
| 5. | Rome Air Development Center
Griffiss AFB
Rome, New York
Attn: RAT | (1) |
| | RAALD | (1) |
| | RAAPT | (1) |
| | RAL | (1) |
| | RALC | (1) |
| | RALS | (1) |
| | RALT | (1) |
| | RAO | (1) |
| | RAD | (1) |
| | RAS | (1) |
| | RASG | (1) |
| | ROAMA (ROAEP-1) | (4) |
| | EMIH | (5) |
| | EMIIT | (5) |
| | EMIF | (5) |
| | EMID | (5) |
| | RASH | (1) |
| | RASS | (1) |
| | RAU | (1) |
| | RAUA | (1) |
| | RAUE | (1) |
| | RAUM | (1) |
| | RAUO | (1) |
| | RAWC | (1) |
| | RAWI | (1) |
| | RAWI | (1) |
| | RAYA | (1) |
| | RAI | (1) |

DISTRIBUTION LIST (Continued)

- | | | |
|-----|---|---------------------------------|
| 6. | GEEIA
Griffiss AFB
Rome, New York
Attn: ROZMA
ROZMC
ROZME
ROZMN
ROZMCAT | (1)
(1)
(1)
(1)
(1) |
| 7. | RAOL (Maj. Shields)
Griffiss AFB
Rome, New York | (1) |
| 8. | RAOL (S/L Tanner)
Griffiss AFB
Rome, New York | (2) |
| 9. | AFSC
Andrews AFB
Washington 25, D. C.
Attn: SCSE
SCLDE
SCFRE | (1)
(1)
(1) |
| 10. | Redstone Scientific Information Center
U. S. Army Missile Command
Redstone Arsenal, Ala.
Attn: Chief, Document Section | (2) |
| 11. | Bureau of Naval Weapons
Main Navy Building
Washington 25, D. C.
Attn: Tech. Lib. DL1-3 | (2) |
| 12. | Chief, Bureau of Ships
Code 454 F
Main Navy Building
Washington 25, D. C. | (1) |
| 13. | Central Intelligence Agency
2430 E Street NW
Washington 25, D. C.
Attn: OCR Mail Room | (1) |
| 14. | U. S. Army Material Command
Harry Diamond Laboratories
Washington 25, D. C.
Attn: AMXDO-TIB | (1) |
| 15. | Scientific and Technical Information Facility
P. O. Box 5700
Bethesda, Maryland
Attn: NASA Representative (S-AK/DL) | (2) |

DISTRIBUTION LIST (Continued)

16. Director (1)
National Security Agency
Ft. George G. Meade, Maryland
Attn: C3/TDL
17. Commander (1)
Naval Missile Center
Tech Library (Code No. 3022)
Pt. Mugu, California
18. Commanding Officer and Director (1)
U. S. Navy Electronic Lab (Lib)
San Diego 52, California
92152
19. Office of Chief of Naval Operations (Op-723 E) (1)
Rm. 1602
Building T-3, Navy Department
Washington 25, D. C.
20. Office of Naval Research (1)
Chief Scientist (Code 427)
Washington 25, D. C.
21. The Rand Corp. (Tech Lib) (1)
1700 Main Street
Santa Monica, California
22. Hq TAC (DORQ-S) (1)
Langley AFB
Virginia
23. Hq TAC (OA) (1)
Langley AFB
Virginia
24. Commander, TAC Comm Region (1)
Langley AFB
Virginia
25. USAFSS (ECD) (1)
San Antonio
Texas
26. Commanding General (2)
US Army Electronic Proving Ground
Attn: Tech Library
Fort Huachuca, Arizona
27. Commanding Officer (1)
US Army Electronics R and D Lab
Attn: SELRA/ADT
Ft. Monmouth, New Jersey

DISTRIBUTION LIST (Continued)

- | | | |
|-----|--|-------------------|
| 28. | Institute of Technology Library
MCLI-LIB, Bldg 125 Area "B"
Wright-Patterson AFB
Ohio | (1) |
| 29. | NAFEC Library
Bldg. 3
Atlantic City, New Jersey | (1) |
| 30. | Commandant
US Army War College (Lib)
Carlisle Barracks, Pa. | (1) |
| 31. | Commanding Officer
US Army Electronic Research Unit
P O Box 205
Mountain View, California | (1) |
| 32. | Electronic Defense Labs
Attn: Library
P O Box 205
Mountain View, California | (1) |
| 33. | U S Naval Avionics Facility (Library)
Indianapolis 18
Indiana | (1) |
| 34. | Commander
US Naval Air Dev Cen (NADC Lib)
Johnsville, Pa. | (1) |
| 35. | Director
US Naval Research Lab (Code 2027)
Washington 25, D. C. | (2) |
| 36. | Hq USAF
Attn: AFRSTE
AFRDPC
AFGOA
Washington 25, D. C. | (1)
(2)
(1) |
| 37. | National Aeronautics and Space Admin
Langley Research Center
Langley Station
Hampton, Virginia
Attn: Librarian | (3) |
| 38. | RTD (RTH)
Bolling AFB 25 D. C. | (1) |
| 39. | Federal Aviation Agency
Information Retrieval Br Hq-630
Washington 25, D. C. | (1) |

DISTRIBUTION LIST (Continued)

- 40. RTD (RTHR/ Capt. K. O. Malkemes (2)
Bolling AFB, D. C.
20332
- 41. OSD (DDR and E/ Mr. Willie Moore (2)
Washington D C
20330
- 42. Hq USAF (AFRST/ Lt. Col. A. H. Grinsted, Jr.) (1)
Washington D C
20330
- 43. Advanced Research Projects Agency (1)
Command and Control Research
(Dr. J. C. R. Licklider)
Washington, D. C.
- 44. National Security Agency (1)
Engineering Research Div, Code R42
(Mr. O. H. Bartlett, Jr.)
Fort Meade, Maryland

