

UNCLASSIFIED

AD 267 379

*Reproduced
by the*

**ARMED SERVICES TECHNICAL INFORMATION AGENCY
ARLINGTON HALL STATION
ARLINGTON 12, VIRGINIA**



UNCLASSIFIED

NOTICE: When government or other drawings, specifications or other data are used for any purpose other than in connection with a definitely related government procurement operation, the U. S. Government thereby incurs no responsibility, nor any obligation whatsoever; and the fact that the Government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use or sell any patented invention that may in any way be related thereto.

CATALOG
AS AD NO

9
H

267 379

Technical Report No. 148
Copy 17 of 35

VOLUME II

SILVER SPRING, MARYLAND
SANTA MONICA, CALIFORNIA
ATLANTA, GEORGIA

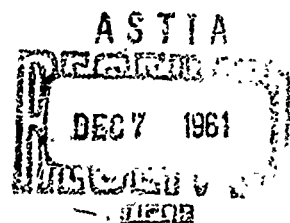
PROJECT PHAROS
ANNUAL COMPREHENSIVE REPORT ON
SENSITIVITY ANALYSIS OF OPERATIONAL
EFFECTIVENESS: CW/BW WEAPONS SYSTEMS

by

Roger C. Eyer, Reynold Greenstone,
Thomas L. Nowberry, and Cophus T. Patch

31 August 1961

Prepared for Director, U.S. Army Chemical Corps
Operations Research Group Army Chemical Center, Maryland
under Contract No. DA 18-108-CML-6554
Order No. CPl-902



6150

OPERATIONS RESEARCH, *Incorporated*
SILVER SPRING, MARYLAND

PROJECT PHAROS
ANNUAL COMPREHENSIVE REPORT ON
SENSITIVITY ANALYSIS OF OPERATIONAL
EFFECTIVENESS: CW/BW WEAPONS SYSTEMS

by

**Roger C. Eyer, Reynold Greenstone,
Thomas L. Newberry, and Cephus T. Patch**

31 August 1961

**Prepared for Director, U.S. Army Chemical Corps
Operations Research Group Army Chemical Center, Maryland
under Contract No. DA 18-108-CML-6554
Order No. CPI-902**

TABLE OF CONTENTS

	Page
I. INTRODUCTION	1
LIST OF FIGURES	iii
II. FLOW CHARTS FOR COMPUTER SIMULATION OF ARTILLERY-FIRED GB SHELLS	3
1. MASTER FLOW CHART	4
2. TARGETING AND DELIVERY	5
3. FRAGMENTATION	6
4. SPLASH	7
5. TIME DEPENDENT VAPOR	8
6. BREATHING PARAMETER SUBROUTINE	9
7. DOSE SUBROUTINE	10
III. DESCRIPTION OF THE FLOW CHARTS	11
NOTATION FOR FLOW DIAGRAM	12
Coordinate Notation; Targeting and Delivery Notation; Fragmentation Notation; Splash Notation; Breathing Parameter Notation; Dosage Notation	
DESCRIPTION OF FLOW CHARTS—ACC-2, 20 JUNE 1961	16
Master Flow Chart; Targeting and Delivery; Fragmentation; Splash; Time Dependent Vapor; Breathing Parameters; Breathhold Rules; Dose Routine	
IV. COMPUTER PROGRAM	45
THE ALGOL VERSION OF THE COMPUTER SIMULATION	47
THE FORTRAN VERSION OF THE COMPUTER SIMULATION	67

LIST OF FIGURES

		Page
1.	Computation of the Impact Coordinates	21
2.	Fragmentation Pattern.	23
3.	Splash Pattern Showing K_{MAX} Ellipse and Circumscribed Rectangle	29
4.	Relative Coordinates in the Wind-Coordinate System	32
5.	Breathing Pattern When t_{-1} Falls in Inhale Portion of Second Cycle (B13, B14 Situation)	38

I. INTRODUCTION

1.1 This volume of the comprehensive report presents the specific computer simulation that has been devised and tested during the contract year.

1.2 In Section II the conventional computer flow charts are presented. In Section III the symbols used in the flow charts are defined and a description of the flow charts is given. This description takes the reader step-by-step through the simulation explaining the purpose of each computer operation.

1.3 In Section IV the literal computer program for the simulation is presented. In order to achieve generality in application, two versions of the program, one in ALGOL and the other in FORTRAN, have been prepared and are presented one after the other.

II. FLOW CHARTS FOR COMPUTER SIMULATION OF ARTILLERY-FIRED GB SHELLS

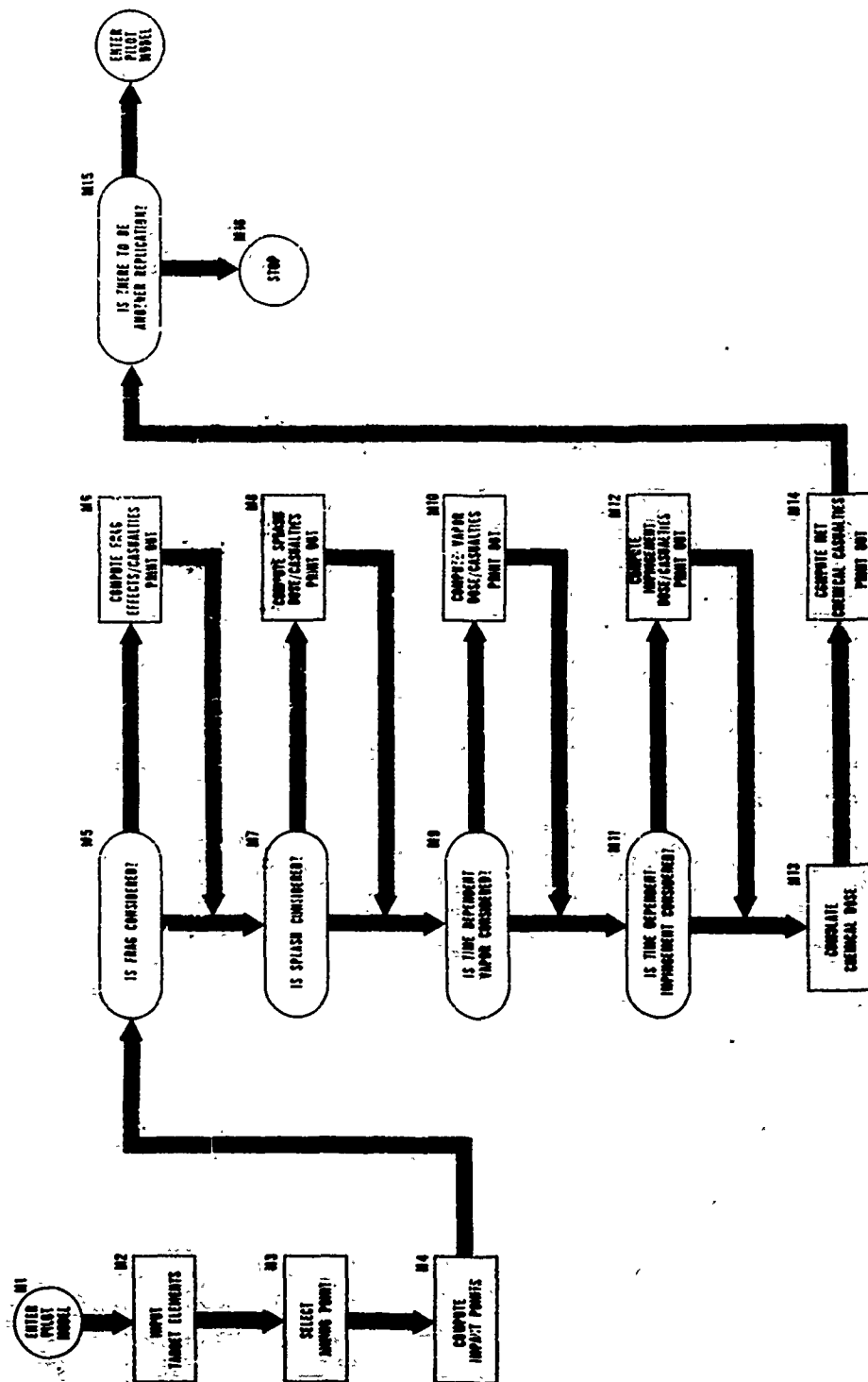
2.1 This section presents the seven flow charts which provide a conventional representation of the computer simulation.

2.2 With the advent of the new computer languages such as ALGOL and FORTRAN, certain simplifications have become available in computer programing which tend to cause minor differences between the flow charts and the programs based on them. For example, the FOR instruction in ALGOL supplants the conventional loop program illustrated in the flow charts. The instruction: FOR N = (1, 1, 12); BEGIN... takes the place of the loop: SET N = 1; PERFORM OPERATION ON VARIABLE N; Is N = 12?; IF N = 12, EXIT; IF N $\neq$ 12, ADD 1 TO N, PERFORM OPERATION ON VARIABLE N....

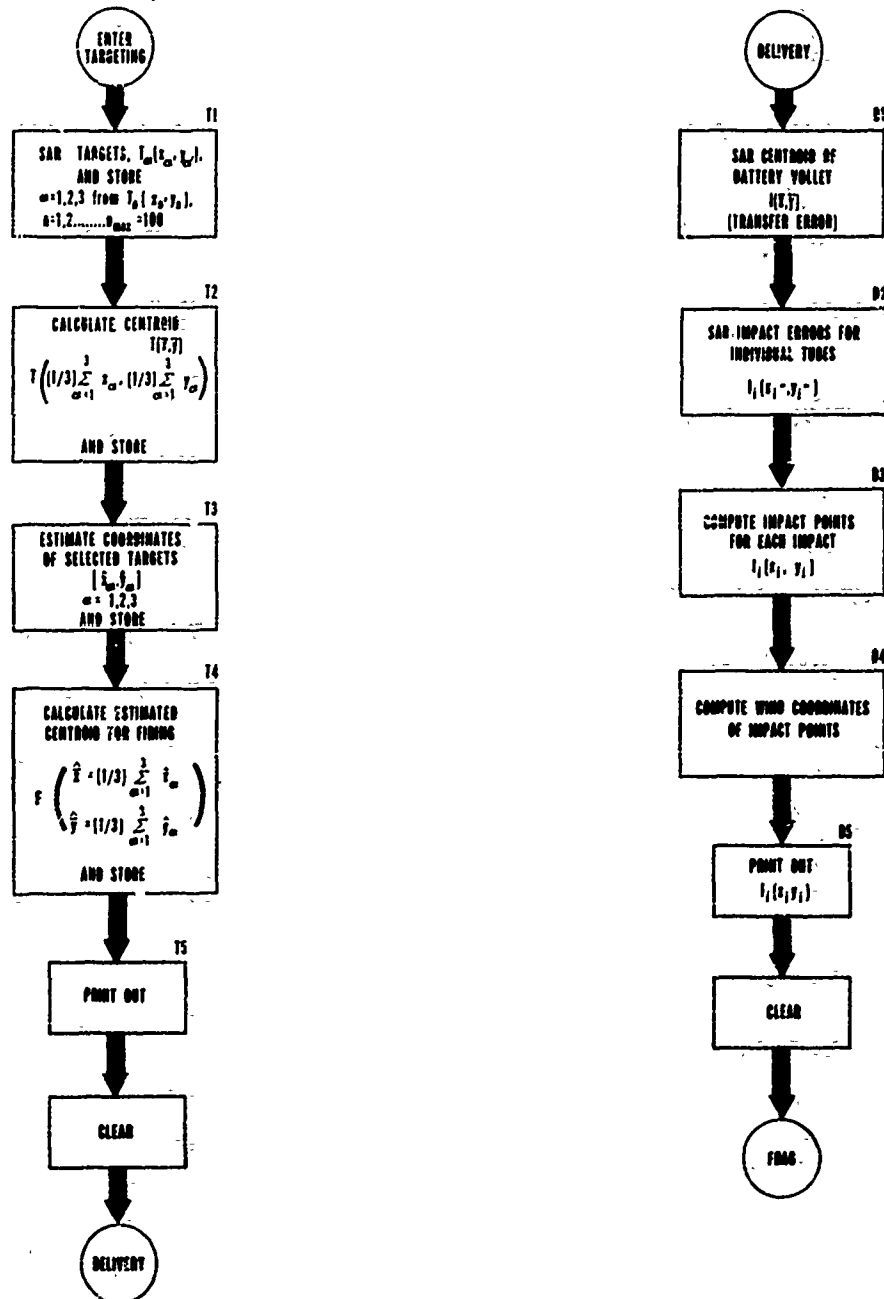
2.3 When several loops are contained one inside the other, the flow charts tend to clarify the order of operations which can become obscure when embodied in the computer programs.

2.4 Flow Chart 1 gives an over-all view of the simulation. The succeeding charts present: (a) targeting and delivery of the GB shells, (b) determination of the fragmentation effect, (c) determination of the splash effect, (d) determination of the GB vapor effect. (This last appears on three charts. On the first chart of the series a geometric determination of whether a target is in the path of the GB vapor cloud is made. On the second chart the determination of breathing parameters, response delay time, breathing phase at impact time, masking time, breathhold time, and gasp time is made. On the third chart the inhaled vapor dose is calculated.)

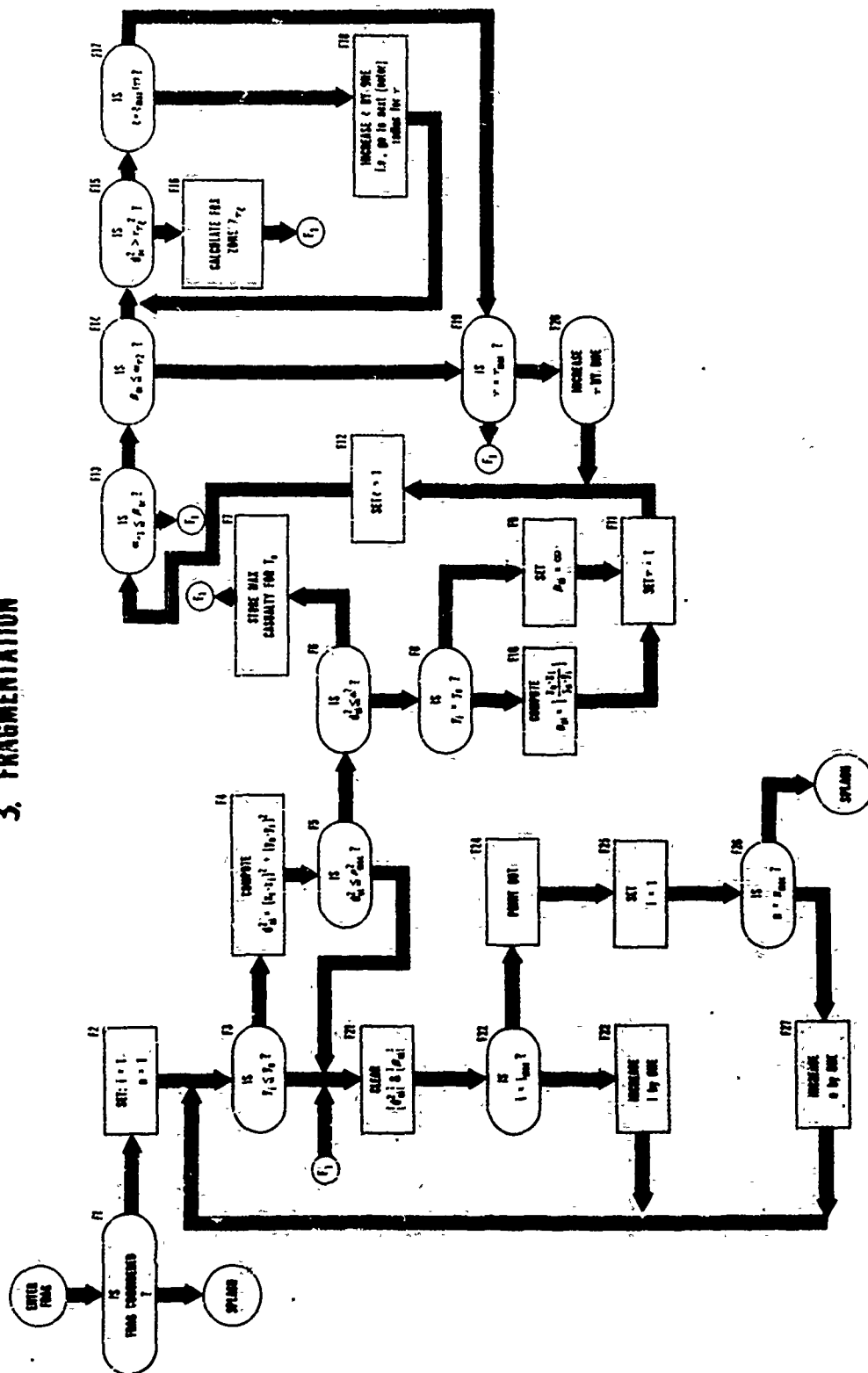
1. MASTER FLOW CHART



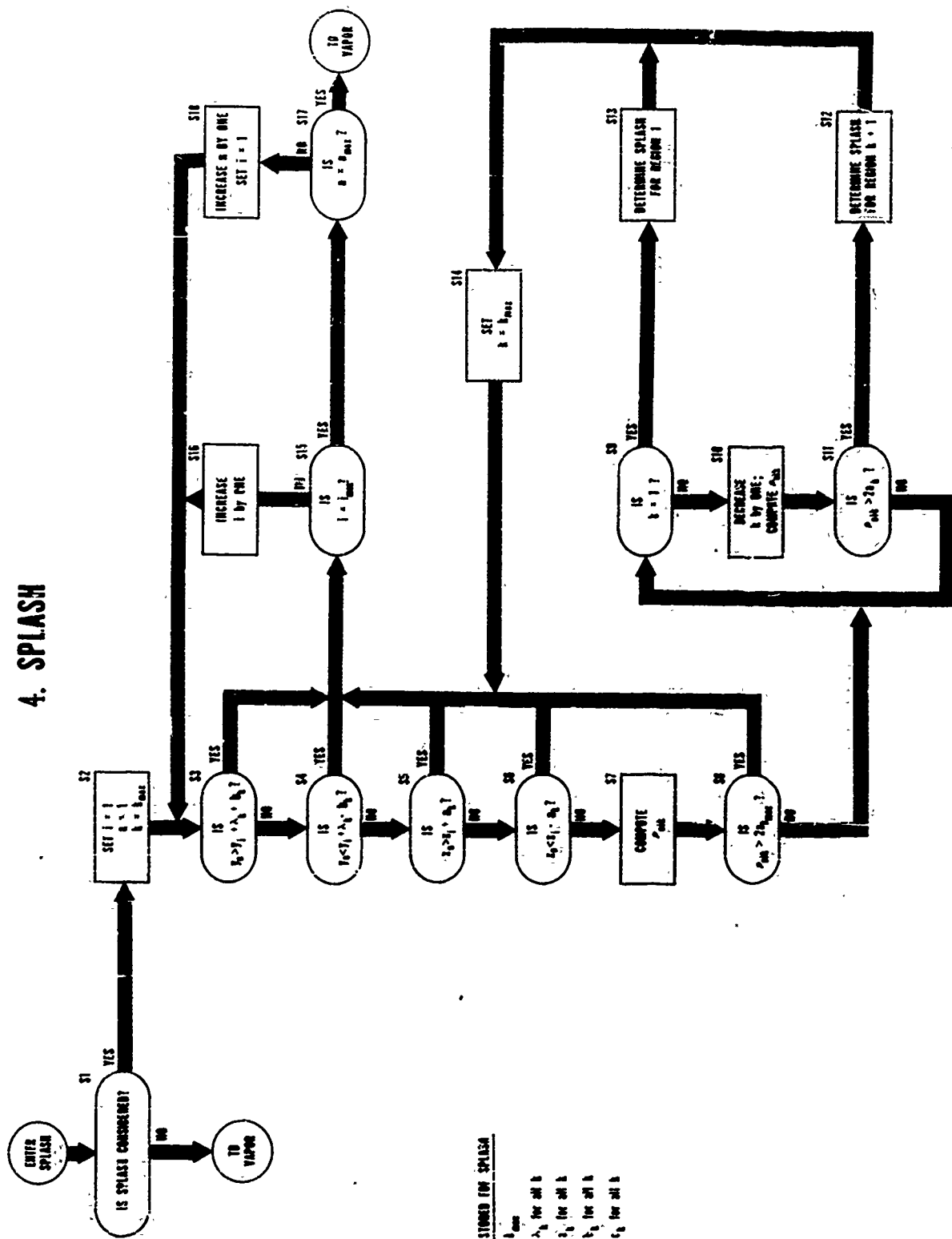
2. TARGETING AND DELIVERY



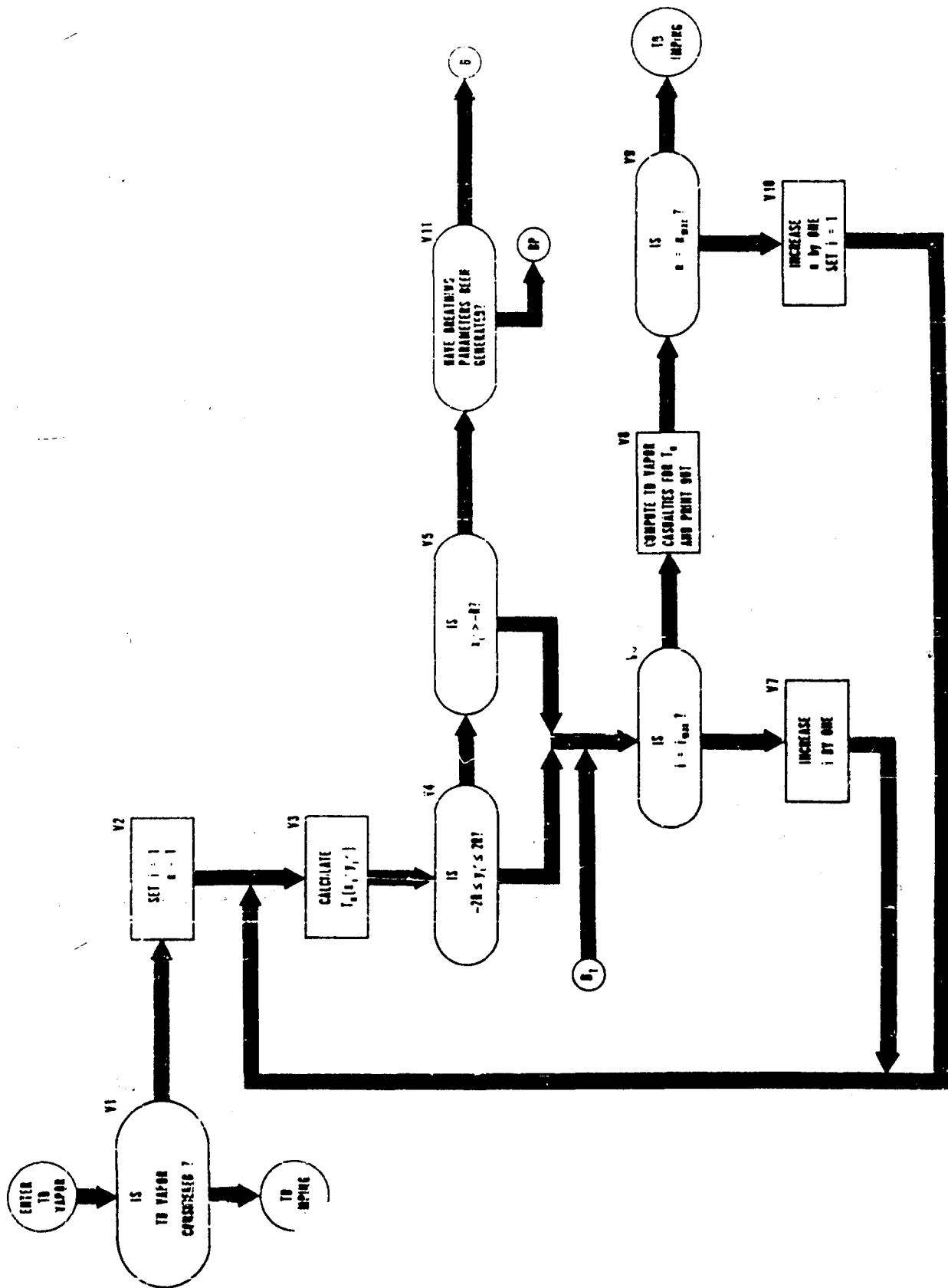
3. FRAGMENTATION



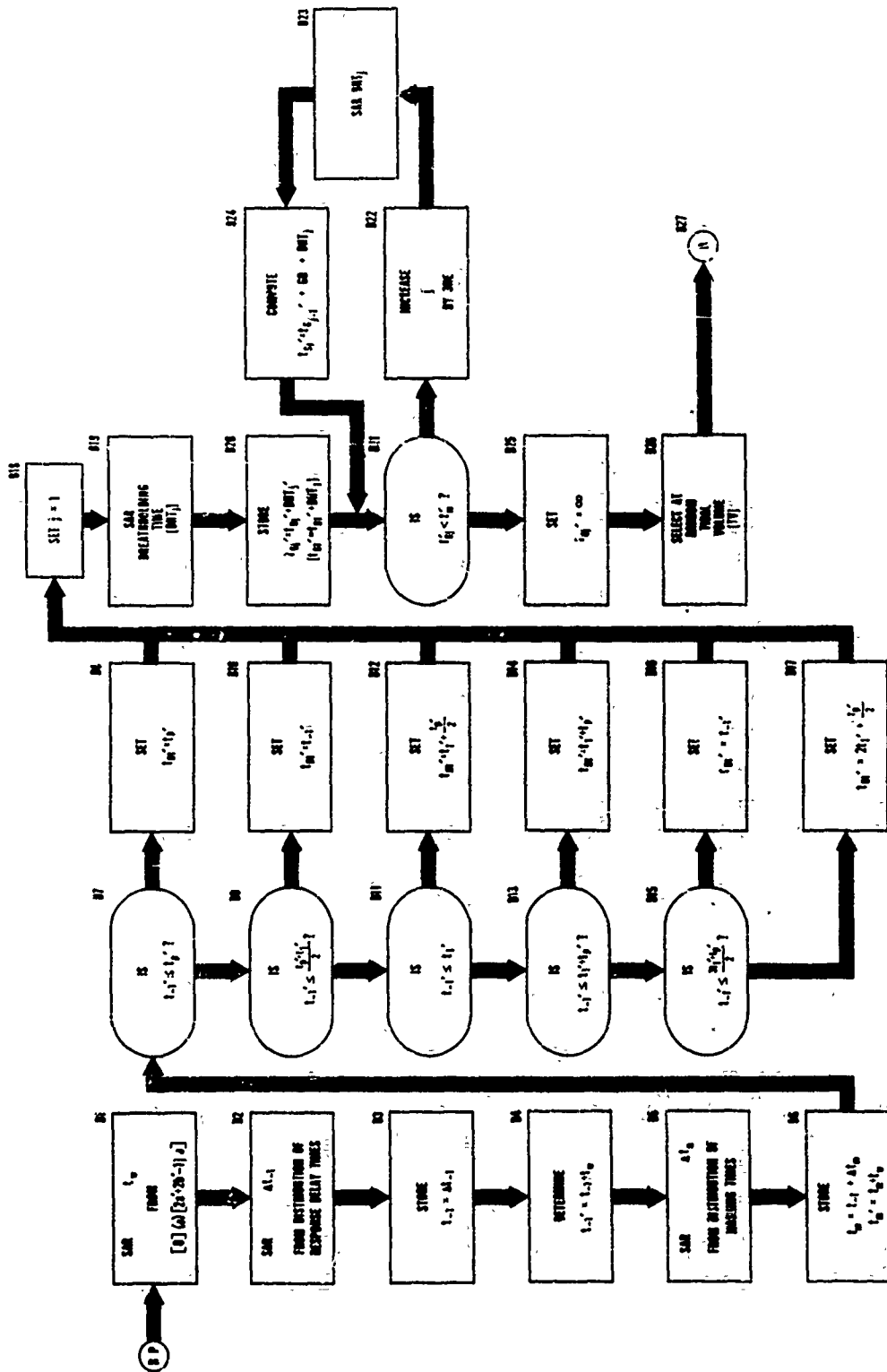
4. SPLASH



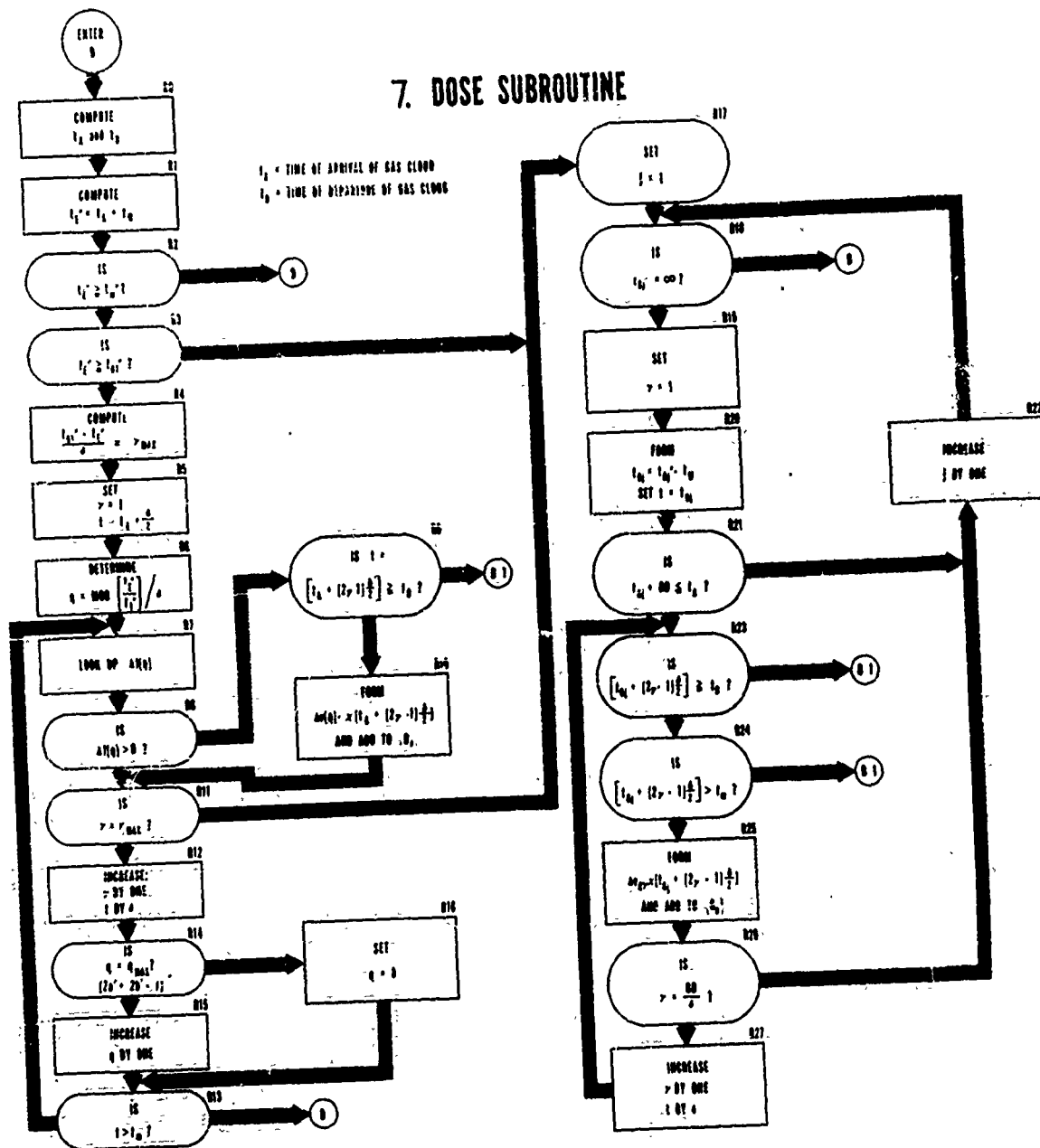
5. TIME DEPENDENT VAPOR



C. BREATHING PARAMETER SUBROUTINE



7. DOSE SUBROUTINE



III. DESCRIPTION OF THE FLOW CHARTS

3.1 This section presents a list of symbols used in the flow charts and a step-by-step description of each operation. Each explanation is keyed to a corresponding block on the flow charts by the use of block numbers.

3.2 Although Flow Chart 1 (the Master Flow Chart) provides for determination of time dependent impingement (downwind impingement), there is no flow chart or program at present to carry out this determination. This feature of the model has not yet been developed, but is expected to be added early in the coming contract year. The modular nature of the model makes this addition quite simple.

3.3 Since this model is, by design, a Monte Carlo simulation, it is of interest to summarize the variables whose values are determined by sampling from probability distributions. In the current form of the simulation, the following probability distributions are used.

- a. Target element location error (assumed symmetrical in x and y ^{1/}).
- b. Battery-volley centroid x-coordinate error.
- c. Battery-volley centroid y-coordinate error.

^{1/} The assumed symmetry in x and y makes the detection process correspond to aerial photographic reconnaissance. If, for example, artillery observer reconnaissance were used, there would be different distributions for x and for y.

- d. Individual-piece x-coordinate error.
- e. Individual-piece y-coordinate error.
- f. Response delay time.
- g. Masking time.
- h. Duration of breathhold.
- i. Tidal volume.

3.4 With further development it will be possible to sample values of additional quantities such as source strength, stability parameter, and wind speed.

3.5 In addition to the sampling of probability distributions just referred to, there are currently three other random determinations:

- a. At the start there is a random selection of three target elements from the array of target elements (see Block T-1).
- b. In the fragmentation routine a target may be determined to be in a zone of potential fragmentation casualty. When this is the case a random number is compared with the probability of casualty for the particular zone. If the random number is less than or equal to the probability the target is a casualty (see Block B-16).
- c. In the breathing parameter subroutine the target's phase in the breathing cycle is selected at random.

3.6 Currently, the chemical dose-response relation is not used in the program which has been written. The final output of a computer run is the total chemical dose received by each target element. It is a relatively simple matter to extend the program so that a Monte Carlo determination of degree of casualty can be made following the determination of the chemical dose.

NOTATION FOR FLOW DIAGRAM

3.7 The notation utilized in the flow diagram is defined under the following classifications:

- 1. Coordinate Notation
- 2. Targeting and Delivery Notation

3. Fragmentation Notation
4. Splash Notation
5. Breathing Parameter Notation
6. Dosage Notation

Coordinate Notation

(x_i, y_i)	Location of impact of the i^{th} shell
(x_n, y_n)	Location of the n^{th} target
(x_o, y_o)	Location of one of the targets observed
(x_o', y_o')	Estimated location of one of the targets observed
$(\bar{x}', \bar{y}')$	Estimated centroid of target observed; location at which battery fires
$(\bar{x}, \bar{y})$	Location of impact centroid
(x'', y'')	Errors in firing of individual guns
(x_i^w, y_i^w)	Location of impact of i^{th} shell in wind coordinates
(x_n^w, y_n^w)	Location of n^{th} target in wind coordinates
(x', y')	Distance downwind and crosswind from impact to target.

Targeting and Delivery Notation

i	Impact index; $i = 1, 2, \dots, i_{\text{MAX}}$
L	Location of impact of shell fired from left-most gun if there were no individual piece firing error.
n	Target index; $n = 1, \dots, n_{\text{MAX}}$
r'	The largest integer obtained by dividing the number 1000 by n_{MAX} .
ζ	The distance between aiming points of each gun.
θ	The direction of the wind in degrees measured counterclockwise from the x-axis.

Fragmentation Notation

d_{ni}	Distance of the n^{th} target from the i^{th} impact.
β_{ni}	Cotangent of the angle between the x-axis and the line between the impact and the target.

Δ	A small distance from the impact in which a maximum casualty occurs.
ρ_{MAX}	Maximum range of fragmentation.
τ	Sector in which fragmentation occurs; $\tau = 1, 2, \dots, \tau_{MAX}$
ξ	Denotes varying radii as measured from the impact points; $\xi = 1, 2, \dots, \xi_{MAX}(\tau)$
$\sigma_{\tau 1}$	Cotangent of the angle between side 1 of the τ^{th} sector and the x-axis.
$\sigma_{\tau 2}$	Cotangent of the angle between side 2 of the τ^{th} sector and the x-axis.
$r_{\tau \xi}$	The radii for the τ^{th} sector.
$Z_{\tau \xi}$	The zone formed by the τ^{th} sector and the radii, $r_{\tau \xi}$ and $r_{\tau, \xi-1}$

Splash Notation

k	Index for denoting ellipses of different sizes; $k = 1, 2, \dots, k_{MAX}$.
a_k	Distance from the center of the k^{th} ellipse to the edge of the ellipse along the major axis.
b_k	Distance from the center of the k^{th} ellipse to the edge of the ellipse along the minor axis.
c_k	Distance from the center of the k^{th} ellipse to each focus.
λ_k	Distance from the center of the k^{th} ellipse to the impact point.
ρ_{nik}	Sum of the distances of the target from each of the two foci for the k^{th} ellipse.

Breathing Parameter Notation

j	Gasp index; $j = 1, 2, \dots, j_{MAX}$.
BHT_j	Time for j^{th} breathhold plus exhale time.
GD	Gasp Duration (duration of inhale portion of gasp).
q	Indexes an interval of width δ in the breathing cycle; $q = 1, 2, \dots, t_1' / \delta$

R	Burst radius of shell.
δ	Increment of time in breathing cycle used for numerical integration.
t_A	Time of arrival of gas cloud relative to time of impact.
t_D	Time of departure of gas cloud relative to time of impact.
Δt_m	Masking time (in increments of δ).
t_m	Time at which masking is completed relative to time of impact.
t_w	The time relative to the beginning of a man's breathing cycle at impact time.
t_{-1}	Response delay time (increments of δ).
t	Time relative to time of impact.
t_E'	Time when gas cloud arrives relative to beginning of the man's breathing cycle.
t_{Gj}'	Time at which the j^{th} "gasp" begins relative to beginning of the man's breathing cycle.
t_{H1}'	Time at which breathholding first begins relative to beginning of breathing cycle.
t_m'	Time at which masking is completed relative to beginning of the man's breathing cycle.
t_p'	The time relative to the beginning of the man's cycle when the peak of the breathing cycle is obtained.
t_1'	The length of the breathing cycle.
t_{-1}'	Time relative to beginning of the breathing cycle when the man can respond to the impact.
TV	Tidal volume of air intake.
a, b	Integers; the duration of inhale is $2a\delta$; the duration of exhale is $2b\delta$; $t_p' = 2a\delta$; $t_1' - t_p' = 2b\delta$.

Dosage Notation

B_n	Location for accumulating steady breathing dose.
-------	--

G_n	Location for accumulating gasp dose.
q	Intervals in breathing cycle; $q = 1, 2, \dots, t_1'/\delta$.
$\bar{u}$	Wind speed.
α	Stability parameter.
γ	Index used for numerical integration.
$\Delta f(q)$	Fraction of TV inhaled during the q^{th} interval.
$\Delta v(q)$	Volume of air intake during the q^{th} interval.
$x(t, x'y')$	Concentration of cloud at time t when target is x' and y' from impact.

DESCRIPTION OF FLOW CHARTS—ACC-2, 20 JUNE 1961

Master Flow Chart

BLOCK M-1. Enter Pilot Model

The input elements—the constants and probability distributions necessary for the battle-field model considered—are entered into the computer.

BLOCK M-2. Input Target Elements

The (x, y) coordinates of the targets are entered into the computer. The number of target elements varies from 3 to 100.

BLOCK M-3. Select Aiming Point

The point at which the guns are to be fired is determined as a consequence of randomly selecting three of the actual target elements (errors associated with target acquisition are considered).

BLOCK M-4. Compute Impact Points

The coordinates of each of the impact points are determined. Two sets of coordinate systems are used throughout the simulation:

1. a regular coordinate system and
2. a wind coordinate system.

BLOCK M-5. Is Frag Considered?

This block makes consideration of fragmentation effects optional.

BLOCK M-6. Compute Frag Effects/Casualties

This block determines the fragmentation effects.

BLOCK M-7. Is Splash Considered?

This block makes consideration of splash effects optional.

BLOCK M-8. Compute Splash Dose/Casualties

This block determines the splash effects.

BLOCK M-9. Is Time Dependent Vapor Considered?

This block makes consideration of time dependent vapor effects optional.

BLOCK M-10. Compute Vapor Dose/Casualties

This block determines the results of the time dependent vapor.

BLOCK M-11. Is Time Dependent Impingement Considered?

This block makes consideration of time dependent impingement effects optional.

BLOCK M-12. Compute Impingement Dose/Casualties

This block determines the results of the time dependent impingement.

BLOCK M-13. Cumulate Chemical Dose

This block determines the total chemical dosage obtained from the splash effect, the time-dependent vapor effect, and the time-dependent impingement effect.

BLOCK M-14. Compute Net Chemical Casualties

This block determines the total chemical casualties as a result of the chemical dose obtained in Block M-13.

BLOCK M-15.

Is There to Be Another Replication?

This block tells the computer the number of times to perform the simulation. If there is to be another replication, the sequence of events that began in Block M-1 is repeated.

BLOCK M-16.

Stop

When all replications for the specified input elements are completed, the computer stops.

The blocks as generally described in the Master Flow Chart will be described in more detail in the following pages.

Targeting and Delivery

BLOCK T-1.

Select at Random Three Targets

From the n_{MAX} target elements entered into the machine, three target elements are selected at random. The random process for selecting these three targets is as follows:

1. The largest integer, r' , that can be obtained by dividing the number 1000 by n_{MAX} is determined.
2. Using the interval width of r' , n_{MAX} intervals are constructed.
3. There is an interval which corresponds to each of the target elements, $1, 2, \dots, n_{MAX}$. (Any remainder, $1000 - (r')(n_{MAX})$, as a consequence of dividing n_{MAX} into 1000, is not a usable random number.)
4. The first target element is obtained by determining the correspondence between the interval in which the first random number falls and the input target elements. If the random number selected is greater than $(r')(n_{MAX})$, another random number is chosen.
5. After the first target is selected, there are two intervals which are not usable, the interval corresponding to the first target selected and the interval greater than $(r')(n_{MAX})$.
6. This process is repeated until three different target elements have been selected.

BLOCK T-2.

Calculate the Centroid

The centroid of the three targets selected at random is calculated by the

following equations:

$$\bar{x} = \sum_{\alpha=1}^3 x_{\alpha}/3 \qquad \bar{y} = \sum_{\alpha=1}^3 y_{\alpha}/3$$

BLOCK T-3. Estimate Coordinates of Randomly Selected Targets

The x and y errors involved in the estimates of the three target elements are determined from a probability distribution of errors. In the current version of the model the variance of the x errors is identical with the variance of the y errors. These variances can be different. The estimated coordinates of the three randomly selected targets are obtained by adding these x and y errors to the actual x and y coordinates of the three randomly selected target elements, respectively.

BLOCK T-4. Calculate Estimated Centroid

The estimated centroid is the point at which the center of the volley is aimed. This point is obtained from the estimated coordinates obtained in Block T-3 by the following two equations:

$$\bar{x} = \sum_{\alpha=1}^3 x_{\alpha}'/3 \qquad \bar{y} = \sum_{\alpha=1}^3 y_{\alpha}'/3$$

BLOCK T-5. Print Targeting Information

The following targeting information is printed:

1. The coordinates of each of the randomly selected target elements (determined in Block T-1).
2. The actual centroid of the three randomly selected elements (determined in Block T-2).
3. The coordinates of the three estimated target elements (determined in Block T-3).
4. The estimated centroid for firing (determined in Block T-4).

BLOCK D-1. Select Impact Centroid

The impact centroid is determined by adding the x and y errors in battery

firing to the coordinates of the estimated centroid (calculated in Block T-4). The x and y errors are obtained from a probability distribution of errors in the impact centroid from the estimated centroid. The variance of the x errors is not equal to the variance of the y errors. This block takes into account the artillery battery delivery error.

BLOCK D-2. Select Impact Errors for Each Impact

The x and y errors for each impact point about the point at which the gun was fired are obtained from a probability distribution of individual piece firing errors. An x and y error is selected for each of the i_{MAX} impact points.

BLOCK D-3. Computer Impact Coordinates in Regular (x, y) System

Figure 1 illustrates the computation of the impact coordinates. The x-coordinate of the impact point of the shell fired from the left-most gun would be located a distance of $(i_{MAX}-1)(\zeta/2) = L$ from the x-coordinate of the impact centroid (determined in Block D-1) if there were no individual piece firing error. The remaining x-coordinates of the points at which the shells would strike if there were no individual piece firing errors are

$$L + \zeta, L + 2\zeta, \dots, L + (i_{MAX}-1)\zeta.$$

By adding the x-errors as obtained in Block D-2 to $L + \zeta, L + 2\zeta, \dots, L + (i_{MAX}-1)\zeta$, the actual x-coordinates of the individual impact points are determined. The actual y-coordinates for each of the individual impact points are determined by adding the y-errors, as obtained in Block D-2, to the y-coordinates of the impact centroid, as obtained in Block D-1.

BLOCK D-4. Compute Wind Coordinates of Impact Points

The impact coordinates are determined when the regular x and y axes are rotated by the number of degrees corresponding to the wind direction. The following equations are used:

$$x_1^w = x_1 \cos \theta + y_1 \sin \theta$$

$$y_1^w = y_1 \cos \theta - x_1 \sin \theta$$

where x_1 and y_1 are the impact coordinates in the regular coordinate system (where the x axis is parallel to FEBA and the y axis is perpendicular to FEBA), x_1^w and y_1^w are the coordinates in the wind coordinate

6 Tubes in Battery ($i_{MAX} = 6$)

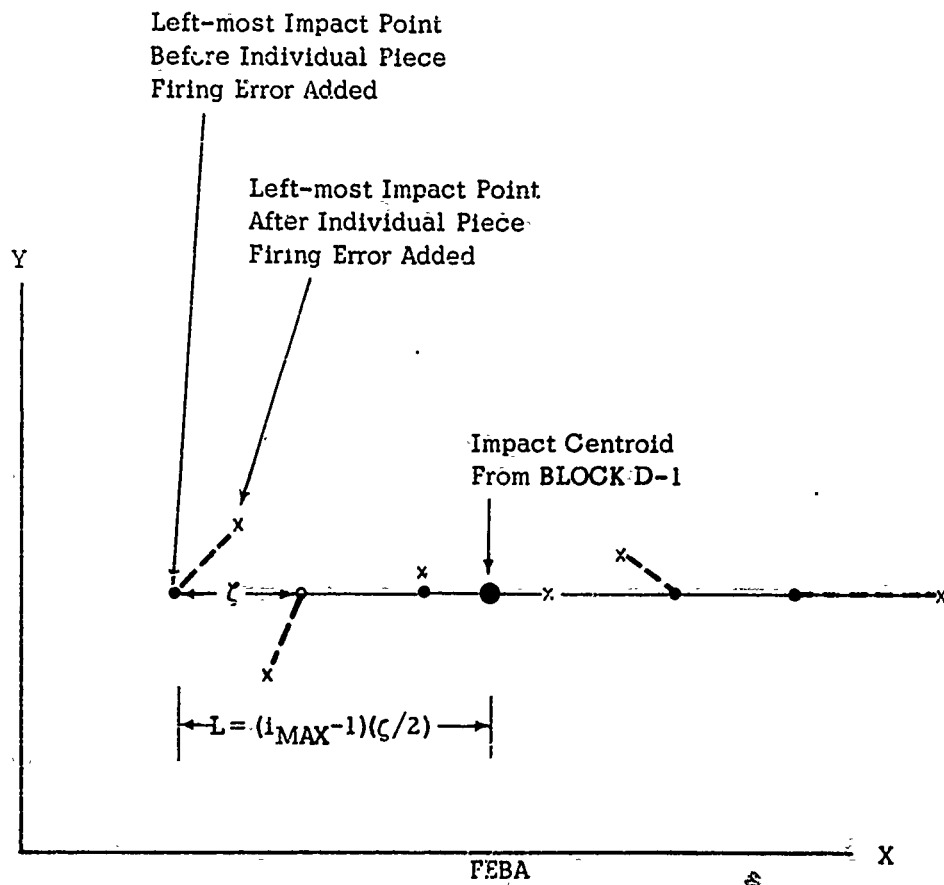


FIGURE 1. COMPUTATION OF THE IMPACT COORDINATES

system, and θ is the wind direction in degrees with respect to the x axis (the x^W axis is parallel to wind direction).

BLOCK D-5. Print Delivery Information

The impact centroid and the impact points for each of the shells are printed both in the regular coordinate system and also in the wind coordinate system.

Fragmentation

BLOCK F-1. Is Frag Considered?

This block makes consideration of fragmentation effects optional.

BLOCK F-2. Set $i = 1$ and $n = 1$

In this block i and n are set equal to one.

BLOCK F-3. Is $y_i \leq y_n$?

See Figure 2. The coordinates of the impact point are denoted by (x_i, y_i) , and the coordinates of the target are denoted by (x_n, y_n) . Whenever the y coordinate of the impact point, y_i , is greater than the coordinate of the target, y_n , the target is not hit.

BLOCK F-4. Compute Distance of Target from Impact Point

For economy in computer operation, the squares of distances are computed and compared, rather than the actual distances. The distance squared of the target from the impact point is computed by the following equation:

$$d_{ni}^2 = (x_n - x_i)^2 + (y_n - y_i)^2$$

BLOCK F-5. Is $d_{ni}^2 \leq \rho_{MAX}^2$?

There is a distance, ρ_{MAX} , from the impact point beyond which fragmentation effects are negligible. If fragmentation effects are negligible, the computer returns to Block F-21. If the target is within ρ_{MAX} , the severity

Distance From Target to
Impact is d_{n1}

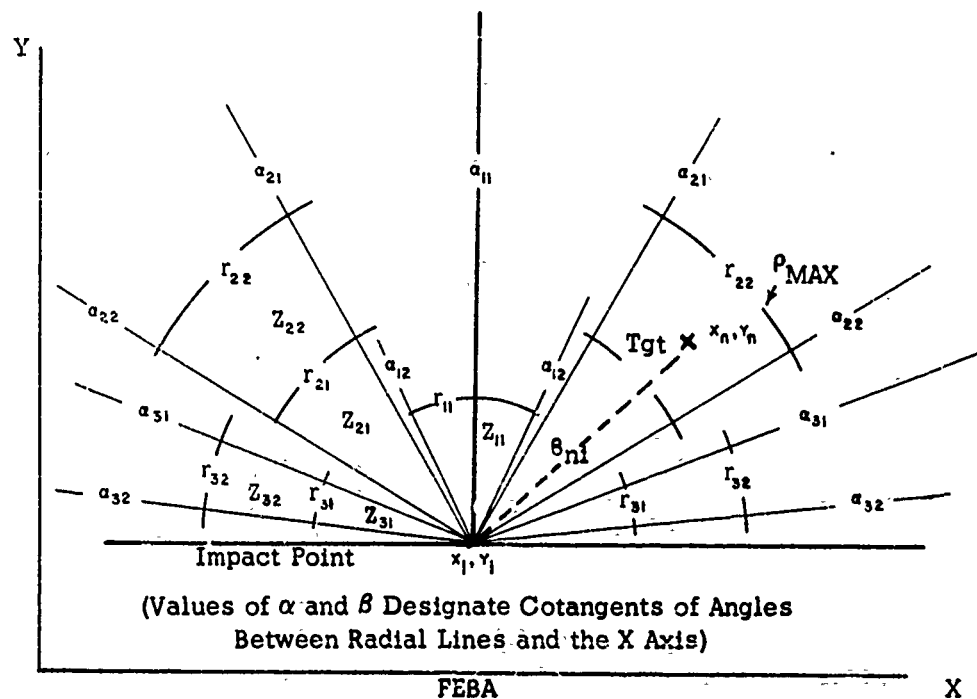


FIGURE 2. FRAGMENTATION PATTERN

of fragmentation effects is computed in the remaining blocks.

BLOCK F-6. Is $d_{ni}^2 \leq \Delta^2$?

It is possible that the distance between the target and the impact point is so small, Δ , that a direct hit is obtained. If a direct hit is obtained, the occurrence of such a hit is stored.

BLOCK F-7. Store Maximum Casualty

If a direct hit has occurred, maximum casualty is stored.

BLOCK F-8. Is $y_i = y_n$?

In this block, and also Block F-9, and F-10, attention is focused on determining the cotangent, β_{ni} , of the angle between the x axis and the line connecting the impact point and the target. If the impact point and the target are the same distance from the x-axis, $y_i = y_n$, β_{ni} is infinite.

BLOCK F-9. Set $\beta_{ni} = \infty$

Since the computer is not equipped to handle a number as large as infinity, special handling must take place. The location which contains β_{ni} is set equal to 9.9×10^{40} to indicate that β_{ni} is equal to infinity.

BLOCK F-10. Compute β_{ni}

Since this cotangent is less than infinity, the computation for β_{ni} is obtained by using the following equation:

$$\beta_{ni} = \left| \frac{x_n - x_i}{y_n - y_i} \right|$$

BLOCK F-11. Set $\tau = 1$

τ denotes the sector in which fragmentation effects occur. (See Figure 2.) With $\tau = 1$, a determination will be made to see if the target is in the first sector. In general, there are τ_{MAX} sectors ($\tau = 1, 2, \dots, \tau_{MAX}$).

BLOCK F-12. Set $\xi = 1$

The Greek letter ξ is used as a subscript to denote varying radii as measured from the impact point. In general, ξ is equal to $1, 2, \dots, \xi_{\text{MAX}(\tau)}$. The radii for the τ^{th} sector are denoted by $r_{\tau 1}, r_{\tau 2}, \dots, r_{\tau \xi_{\text{MAX}(\tau)}}$.

BLOCK F-13. Is $\alpha_{\tau 1} \leq \beta_{ni}$?

As shown in Figure 2, $\alpha_{\tau 1}$ is the cotangent of the angle between side one of the τ^{th} sector and the x axis. If $\alpha_{\tau 1}$ is not less than or equal to β_{ni} , the target is not in the τ^{th} sector. In this case the machine returns to F-21 and the next impact is considered. If $\alpha_{\tau 1}$ is less than or equal to β_{ni} , a check is made in Block F-14 against side two of the τ^{th} sector.

BLOCK F-14. Is $\beta_{ni} \leq \alpha_{\tau 2}$?

The notation $\alpha_{\tau 2}$ is used to denote the cotangent of the angle between side two of the τ^{th} sector and the x axis. If β_{ni} is equal to or less than $\alpha_{\tau 2}$, the target is in the τ^{th} sector.

BLOCK F-15. Is $d_{ni}^2 > r_{\tau \xi}^2$?

Since ξ is initially set to one, a no answer means that the target lies within radius one of sector τ . This region is denoted zone $Z_{\tau 1}$. If the answer is yes, ξ is increased until the zone is determined.

BLOCK F-16. Calculate Fragmentation Effects for Zone $Z_{\tau \xi}$

Corresponding to each zone, $Z_{\tau \xi}$, there is a probability that a fragmentation casualty occurs. A casualty occurs if a random number is equal to or less than the probability of a casualty. The number of casualties for each target element is accumulated over all impacts.

BLOCK F-17. Is ξ equal to $\xi_{\text{MAX}(\tau)}$?

If ξ is equal to $\xi_{\text{MAX}(\tau)}$, the target element is in an area free from fragmentation. The sequence of operations in F-21 follows and the effects of the next impact are considered. If ξ is not equal to $\xi_{\text{MAX}(\tau)}$, then the target element may be within a radius corresponding to a greater value of ξ and the next larger radius is considered.

BLOCK F-18. Increase ξ by one

For the same sector of τ , ξ is increased to the next radius, $\xi + 1$. The sequence of operations in Block F-15 follows.

BLOCK F-19. Is $\tau = \tau_{MAX}$?

If τ is equal to τ_{MAX} , then the target element is outside the last fragmentation sector and fragmentation effects are negligible. If τ is not equal to τ_{MAX} (that is, τ is less than τ_{MAX}), the next τ sector is considered.

BLOCK F-20. Increase τ by one

Since τ is less than τ_{MAX} , not all sectors have been considered. Therefore, the next sector is considered by increasing τ by one. After increasing τ , the sequence of operations in Block F-12 follows.

BLOCK F-21. Clear d_{ni}^2 and β_{ni}

Since the next impact is to be considered, certain storage locations must be cleared.

BLOCK F-22. Is $i = i_{MAX}$?

If i is equal to i_{MAX} , all impact points for this target element have occurred and a print-out follows. If i is not equal to i_{MAX} (that is, i is less than i_{MAX}), further impact points are to be considered.

BLOCK F-23. Increase i by one

The next impact is considered by increasing i by one.

BLOCK F-24. Print Out

The results of the impacts on the target elements are printed out. Specifically, the number of times in which the target element is a casualty is printed out.

BLOCK F-25. Set $i = 1$

The effects of each of the i_{MAX} impacts on the next target element are

considered, again beginning with the first impact.

BLOCK F-26. Is $n = n_{MAX}$?

If n is equal to n_{MAX} , the effects of all impacts on each of the target elements have been evaluated and printed; therefore, the computer enters the routine for evaluating the effects of splash. If n is not equal to n_{MAX} (that is, n is less than n_{MAX}), the effects on all of the target elements have not been evaluated.

BLOCK F-27. Increase n by One

The value of n is increased by one so that the fragmentation effects on the next target element may be evaluated.

Splash

BLOCK S-1. Is Splash Considered?

This block makes consideration of splash effects optional.

BLOCK S-2. Set $i = 1, n = 1, k = k_{MAX}$

This block initializes values for the three parameters i, n , and k . The letter k , is used as an index for denoting ellipses of different sizes; as k increases, the size of the ellipse increases.

Note to BLOCKS S-3, S-4, S-5, and -6:

In Blocks S-3, S-4, S-5, and S-6, the index k has only the value k_{MAX} so that only the largest splash region is under consideration. The largest splash ellipse, k_{MAX} , is approximated by a circumscribing rectangle. If the target element lies inside this rectangle, further tests starting with BLOCK S-7 are employed. These tests determine the annular region between ellipses in which the target element lies.

BLOCK S-3. Is $y_n > y_i + \lambda_k + b_k$?

This block determines whether the target element is too far downrange from the point of impact.

BLOCK S-4. Is $y_n < y_i + \lambda_k - b_k$?

This block determines whether the target element is too far uprange from the point of impact.

BLOCK S-5. Is $x_n > x_i + a_k$?

This block determines whether the target element is too far to the right of the point of impact.

BLOCK S-6. Is $x_n < x_i - a_k$?

This block determines whether the target element is too far to the left of the point of impact.

BLOCK S-7. Compute ρ_{nik}

Refer to Figure 3. The symbol ρ_{nik} is used to denote the sum of the distances of the target from each of the two foci of the k^{th} ellipse and c_k is used to denote the distance from the center of the k^{th} ellipse to each focus. The formula for ρ_{nik} is as follows:

$$\rho_{nik} = \sqrt{(x_n - x_i)^2 + (y_n - y_i + c_k - \lambda_k)^2} + \sqrt{(x_n - x_i)^2 + (y_n - y_i - c_k - \lambda_k)^2}.$$

BLOCK S-8. Is $\rho_{nik} > 2a_{k\text{MAX}}$?

If $\rho_{nik} > 2a_{k\text{MAX}}$, then the target element is not within the $k_{\text{MAX}}^{\text{th}}$ ellipse and the next impact point will be considered. If ρ_{nik} is less than $2a_{k\text{MAX}}$, then the target element is within the $k_{\text{MAX}}^{\text{th}}$ ellipse. The subsequent blocks determine the lowest numbered ellipse which includes the target element.

BLOCK S-9. Is $k = 1$?

If k is equal to one, the target element is within the first ellipse or region one. Proceed to Block S-13. If k is not equal to one, proceed to Block S-10.



.....

BLOCK S-10.

Decrease k by One and Compute ρ_{nik}

The value of ρ_{nik} for the next smaller ellipse is computed.

BLOCK S-11.

Is $\rho_{nik} > 2a_k$?

If ρ_{nik} is greater than $2a_k$, then the target element is not within the k^{th} ellipse, but is within the $k + 1^{\text{th}}$ ellipse. An inquiry is made about the target being in the k^{th} ellipse only if the target is within the $k + 1^{\text{th}}$ ellipse. If ρ_{nik} is less than $2a_k$, then the target element is within the k^{th} ellipse. Therefore, the target element is in one of the following regions: $k, k - 1, \dots, 2, 1$.

BLOCK S-12.

Determine Splash for Region $k + 1$

Since it has been ascertained that the target element is within the $k + 1^{\text{th}}$ region, the corresponding amount of splash is determined.

BLOCK S-13.

Determine Splash for Region 1

As a result of the logic in Block S-9, it is determined that the target element is within region one and the corresponding quantity of splash is determined.

BLOCK S-14.

Set $k = k_{\text{MAX}}$

Since the location of the target element with respect to the splash regions has been determined for this impact, the value of k is reset to k_{MAX} for use in determining the location of the target element with respect to the next impact.

BLOCK S-15.

Is $i = i_{\text{MAX}}$?

If i is equal to i_{MAX} , all impacts have been considered with reference to a specific target element. If i is not equal to i_{MAX} (that is, i is less than i_{MAX}), then the next impact is to be considered.

BLOCK S-16.

Increase i by One

The next impact is considered by increasing i by one.

BLOCK S-17. Is $n = n_{MAX}$?

If n is equal to n_{MAX} , all target elements have been evaluated for all impacts and the computer will proceed to the evaluation of time-dependent vapor. If n is not equal to n_{MAX} (that is, n is less than n_{MAX}), the next target element is considered.

BLOCK S-18. Increase n by one, set $i = 1$

In order to ascertain the effects of the splash upon additional target elements for each of the impacts, n is increased by one and i is set equal to one.

Time Dependent Vapor

BLOCK V-1. Is Time Dependent Vapor Considered?

This block makes consideration of time dependent vapor effects optional.

BLOCK V-2. Set $i = 1$; $n = 1$

This block initializes values for the parameters i and n .

BLOCK V-3. Calculate x' and y' distances from impact to target

(See Figure 4.) This block calculates the x' distance and y' distance (wind coordinates) between the target and the impact. The equations used for this calculation are as follows:

$$x' = x_n^W - x_1^W \quad \text{and} \quad y' = y_n^W - y_1^W$$

BLOCK V-4. Is $-2R \leq y' \leq 2R$?

This block determines whether or not the target is within permissible bounds on the y^W axis. R is the burst radius of the cloud. If $-2R \leq y' \leq 2R$, then the target may be subject to the vapor cloud effects. If y' is less than $-2R$ or if y' is greater than $+2R$, the vapor cloud has no effect upon the target.

BLOCK V-5. Is $x' > -R$?

If x' is greater than $-R$, then the vapor cloud may affect the target. If x' is not greater than $-R$ (that is, x' is less than $-R$), the vapor cloud has no effect upon the target.

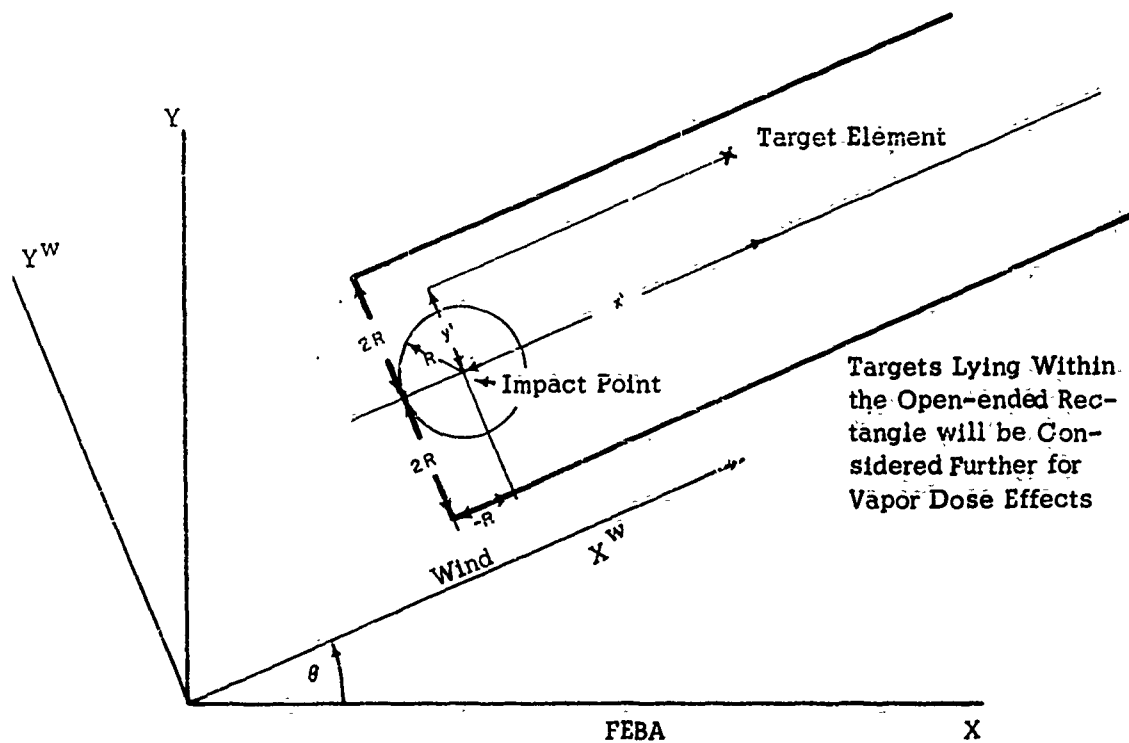


FIGURE 4. RELATIVE COORDINATES IN THE WIND-COORDINATE SYSTEM

BLOCK V-6. Is $i = i_{MAX}$?

If i is equal to i_{MAX} , the effects of the vapor cloud for each of the impacts have been considered upon a specific target. If i is not equal to i_{MAX} (that is, i is less than i_{MAX}), then the effects of additional impacts must be considered for the target under consideration.

BLOCK V-7. Increase i by One

The value of i is increased by one, so that the vapor effects of the next impact upon the target element may be evaluated.

BLOCK V-8. Compute Time Dependent Vapor Casualties for a Target Element and Print Out

The effects of the various dosages of vapor, which are obtained by breathing and gasping, are summarized for each man and printed out. Also, there is a print-out for each man which gives the number of times, if any, in which he received a casualty-producing dose.

BLOCK V-9. Is $n = n_{MAX}$?

If n is equal to n_{MAX} , the effects of each of the vapor clouds for all impacts upon each of the target elements have been evaluated and pertinent information printed out; therefore, the computer enters the routine for evaluating the effects of the time dependent impingement. If n is not equal to n_{MAX} (that is, n is less than n_{MAX}), the next target element is considered.

BLOCK V-10. Increase n by One; set $i = 1$

The value of n is increased by one, so that the vapor cloud effects upon the next target element will be evaluated for each of the impacts.

BLOCK V-11. Have Breathing Parameters Been Generated?

If the breathing parameters have been generated, the computer enters the dose subroutine. If the breathing parameters have not been generated, the computer enters the breathing parameter subroutine.

Breathing Parameters

BLOCK B-1. Select at Random t_w .

The random variable, t_w , is the time relative to the beginning of a man's breathing cycle at impact time. The variable t_w is expressed as an integral number of intervals of width δ . The inhale time is $2a\delta$; the exhale time is $2b\delta$; a and b are integers.

BLOCK F-2. Select at Random Response Delay Time

This is the time which the target element takes before reacting to the impact. This delay time is expressed as an integral number of intervals of width δ .

BLOCK B-3. Store Response Delay Time

The value of the response delay time, t_{-1} , is stored. Note: In the determination of the breathing parameters, it is convenient to have two time scales. One time scale is measured from impact time. The other scale is measured from the beginning of the breathing cycle of a particular target element. This time scale is denoted by time symbols containing a prime (that is, t'). A time scale in terms of the impact is shifted t_w time units from the other time scale.

BLOCK B-4. Determine t_{-1}'

Symbol t_{-1}' is used to denote the time relative to the beginning of the breathing cycle when the target element begins to respond to the impact. This time value is obtained as follows:

$$t_{-1}' = t_{-1} + t_w.$$

BLOCK B-5. Select at Random Masking Times

The time required for each target element to mask, Δt_m , is determined from a distribution of masking times, which are expressed as an integral number of intervals of width δ .

BLOCK B-6. Compute and Store Masking Times

The times at which masking occurs is determined for both time scales from

the following equations:

$$t_m = t_{-1} + \Delta t_m \text{ and}$$

$$t_m' = t_m + t_w.$$

Breathhold Rules

There are five blocks which are used to determine the time at which the target element begins the breathhold procedure. See Figure 5. The rules which are used in establishing the time for breathhold are as follows:

- (1) If at the end of the response delay time, a man is in the inhale portion of the cycle, he will continue inhaling until he reaches the cycle peak, t_p' .
- (2) If at the end of the response delay time, the man is in the exhale portion and has exhaled less than 50% of the volume of air, he will begin the breathhold immediately.
- (3) If at the end of the response delay time, the man is in the exhale portion of the cycle and has exhaled more than 50% of the volume of air, he will continue exhaling and inhale 50% of the normal volume before breathhold begins.

BLOCK B-7.

$$\text{Is } t_{-1}' \leq t_p'?$$

If t_{-1}' is equal or less than t_p' then the man was in the inhale portion of the cycle. Otherwise, the man was not in the inhale portion of the first cycle.

BLOCK B-8.

$$\text{Set } t_{H1}' = t_p'$$

In accordance with rule one, as stated above, the man begins breathhold at the peak of the cycle, if he is in the inhale portion at the end of the response delay time.

BLOCK B-9.

$$\text{Is } t_{-1}' \leq \frac{t_p' + t_1'}{2} ?$$

If t_{-1}' is equal to or less than $\frac{t_p' + t_1'}{2}$, then the man was in the exhale portion of the cycle and less than 50% of all the volume of intake air had

been exhaled. If t_{-1}' is not equal to or less than $\frac{t_p' + t_1'}{2}$, the man is not in the first half of the exhale portion of the cycle in which the impact occurred.

BLOCK B-10.

$$\text{Set } t_{H1}' = t_{-1}'$$

In accordance with the breathhold rules, when the man is in the exhale portion of the cycle and less than 50% volume has been exhaled, he begins breathhold immediately.

BLOCK B-11.

$$\text{Is } t_{-1}' \leq t_1'?$$

If t_{-1}' is equal to or less than t_1' , then the man was in the exhale portion of the cycle and more than 50% of volume had been exhaled. If t_{-1}' is not equal to or less than t_1' , then the man was not in the breathing cycle in which the impact occurred.

BLOCK B-12.

$$\text{Set } t_{H1}' = t_1' + \frac{t_p'}{2}$$

In accordance with the third breathhold rule, when the man is in the exhale portion of the cycle and more than 50% of his intake volume has been exhaled, the time at which he begins breathhold occurs after he has completed the exhale and half of the next inhale.

BLOCK B-13.

$$\text{Is } t_{-1}' \leq t_1' + t_p'?$$

If t_{-1}' is equal to or less than $t_1' + t_p'$, then the man is in the inhale portion of the second cycle. If t_{-1}' is not equal to or less than t_p' , then the man is not in the inhale portion of the second cycle.

BLOCK B-14.

$$\text{Set } t_{H1}' = t_1' + t_p'$$

In accordance with the first breathhold rule, the time at which breathhold begins is at the peak of the inhale cycle.

BLOCK B-15.

$$\text{Is } t_{-1}' \leq \frac{3t_1' + t_p'}{2}?$$

If t_{-1}' is equal to or less than $\frac{3t_1' + t_p'}{2}$, the man was in the exhale portion

of the second cycle and less than 50% of volume had been exhaled. If t_{-1}' is not equal to or less than $\frac{3t_1' + t_p'}{2}$, then the man is not in the first half of the exhale portion of the second cycle since the impact occurred.

BLOCK B-16. Set $t_{H1}' = t_{-1}'$

In accordance with rule three, when the man is in the exhale portion of the cycle and less than 50% of volume has been exhaled, the breathhold is begun immediately.

BLOCK B-17. Set $t_{H1}' = 2t_1' + \frac{t_p'}{2}$

In the logic utilized in the program, it is implicit that the man will be in either the first or second breathing cycle when he begins to respond to the impact. Consequently, with a negative response to Block B-15, the man is in the exhale portion of the second cycle and more than 50% of the volume of air has been exhaled. Therefore, in accordance with the third breathhold rule, he will complete the exhale procedure and inhale 50% of the normal inhale volume prior to holding his breath.

BLOCK B-18. Set $j = 1$

The following blocks are concerned with determining the times at which gasps following the first breathhold time occur.

BLOCK B-19. Select at Random a Breathholding Time

From a probability distribution of breathholding times a value denoted as BHT_j is obtained. This variable, BHT_j , is expressed as an integral number of intervals of width δ .

BLOCK B-20. Store t_{Gj}'

The time at which, for example, the first gasp occurs, is equal to the time of the first breathhold plus the breathhold time. (See Figure 5.) Note that the breathhold time includes the time for exhaling.

BLOCK B-21. Is $t_{Gj}' < t_m'$?

This block inquires whether or not the j^{th} gasp occurred prior to the time

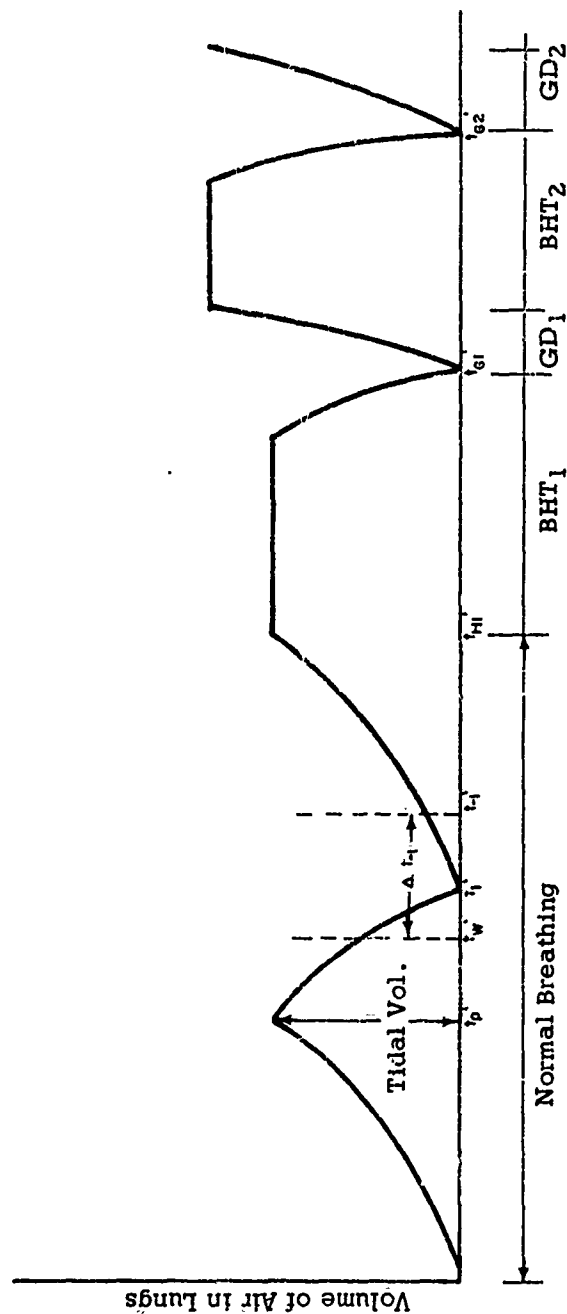


FIGURE 5. BREATHING PATTERN WHEN t_{i1} FALLS IN INHALE PORTION OF SECOND CYCLE (B13, B14 SITUATION)

at which masking was completed. If t_{Gj}' is less than t_m' , then the man has not masked himself at the beginning of the j^{th} gasp. If t_{Gj}' is not less than t_m' , masking has occurred prior to the beginning of the j^{th} gasp; consequently, the j^{th} gasp does not produce any vapor dose.

BLOCK B-22. Increase j by One

The test of Block B-21 is applied to the $(j + 1)^{\text{th}}$ gasp.

BLOCK B-23. Select at Random BHT_j

From a probability distribution of breathhold times, a value for the next breathhold time will be selected.

BLOCK B-24. Compute t_{Gj}'

All values of t_{Gj}' for j equal to 2, 3, ..., are obtained by adding to the time at which the previous gasp began, the gasp duration, GD, and the next breathholding time.

BLOCK B-25. Set $t_{Gj}' = \infty$

When it is determined that the masking time occurred prior to the gasping time a large number, 9.9×10^{40} , is placed in storage to represent infinity.

BLOCK B-26. Select at Random TV

The volume of air intake is selected at random from a probability distribution of tidal volumes. Note: The following breathing parameters have been generated for a specific target element:

t_{H1}'
 $t_{G1}', t_{G2}', \dots, \infty$
 TV
 t_m'

BLOCK B-27. Exit to Dose Subroutine

After the breathing parameters have been determined, the computer begins the dose subroutine.

Dose Routine

BLOCK R-0.

Compute t_A and t_D

This block computes the times of arrival and departure of the gas cloud for a particular target from the following equations:

$$t_A = \frac{x' - 2R}{\bar{u}} \quad \text{and} \quad t_D = \frac{x' + 2R}{\bar{u}}$$

If the formula leads to negative t_A , t_A is set equal to zero. Both t_A and t_D are expressed as an integral number of intervals of width δ .

BLOCK R-1.

Compute $t_E' = t_A + t_W$

This block computes the time of arrival of the gas cloud with respect to the beginning of the man's breathing cycle.

BLOCK R-2.

Is $t_E' \geq t_m'$?

If the time at which the gas cloud arrives is greater than the time at which the man puts on his gas mask, then the man receives no dose. The effects of the next impact will then be determined with respect to this same man.

BLOCK R-3.

Is $t_E' \geq t_{H1}'$?

If the time at which the gas cloud arrives is equal to or greater than the time of first breathhold, then the machine will determine the gasp dose. However, if the time at which the gas cloud arrives is less than the time of first breathhold the steady-breathing dose will be determined.

BLOCK R-4.

Compute $\frac{t_{H1}' - t_E'}{\delta} = \gamma \text{ MAX}$

This block determines the number of increments which will be used in the numerical integration for ascertaining the quantity of vapor inhaled prior to the beginning of the first breathhold.

BLOCK R-5.

Set $\gamma = 1$

$$t = t_A + \frac{\delta}{2}$$

The index, γ , is used to denote the number of the next numerical integration to be computed. The time, t , is set at the midpoint of the first interval after the gas cloud arrives.

BLOCK R-6. Determine $q = \text{MOD} \left[\frac{t_E'}{t_1'} \right] / \delta$

The breathing cycle increment number for the target, corresponding to the time at which the gas cloud arrives, is determined.

BLOCK R-7. Look-up $\Delta f(q)$

The fraction of TV inhaled in the q^{th} increment of the breathing cycle is determined from a table placed in the machine.

BLOCK R-8. Is $\Delta f(q) > 0$?

If the fraction of TV inhaled is greater than zero, the incremental volume inhaled is obtained as follows:

$$\Delta v(q) = TV [\Delta f(q)]$$

If the quantity of vapor inhaled is equal to zero, this computation is avoided.

BLOCK R-9. Is $t = t_A + (2\gamma - 1) \frac{\delta}{2} \geq t_D$?

If the mid-point of the interval is greater than the time of departure of the gas cloud, then no vapor is inhaled since the gas cloud has passed by. If the mid-point of the interval is less than the time at which the gas cloud departs, then the man may breathe while the gas cloud is around him.

BLOCK R-10. Form $[\Delta v(q)] [X(t, x', y')]$ and add to B_n

If the gas cloud has not passed by, then the quantity of vapor inhaled is computed as a product of the quantity of air inhaled times the concentration of the vapor. The concentration of the vapor is a function of the time since the impact and the x and y coordinates in terms of the wind-coordinate system. This product is then added to the previous quantity of steady-breathing dosage for the man which is stored in B_n .

BLOCK R-11.

Is $\gamma = \gamma_{MAX}$?

If γ is equal to γ_{MAX} , all of the intervals in which the man could have breathed the vapor prior to beginning the breathhold routine have been considered. The machine will next determine the quantity of vapor inhaled during the gasping routine. If γ is not equal to γ_{MAX} , (that is, γ is less than γ_{MAX}), then the subsequent interval will be considered.

BLOCK R-12.

Increase γ by one; (t by δ)

γ and t are incremented by one and δ , respectively, for use in determining the concentration of gas in the next interval.

BLOCK R-13.

Is $t > t_m$?

If t is greater than t_m , then the man is masked prior to the mid-point of the subsequent interval and, as a result, he will receive no vapor. The machine then will determine the effects of the next impact upon this man.

BLOCK R-14.

Is $q = q_{MAX}$? ($q_{MAX} = \frac{t_1}{\delta}$)

If q is equal to q_{MAX} , then the dose for all intervals in this breathing cycle has been computed for this man. If q is less than q_{MAX} , then the man is still in the same breathing cycle.

BLOCK R-15.

Increase q by One

The index q is increased by one so that the fraction of the tidal volume inhaled during the subsequent interval can be obtained as in Block R-7.

BLOCK R-16.

Set $q = 1$

The index q is set equal to one so that the next breathing cycle for the man can be properly initiated.

BLOCK R-17.

Set $j = 1$

The index for the gasp time is set equal to one.

BLOCK R-18.

Is $t_{Gj} = \infty$?

If t_{Gj}' is equal to ∞ , this implies that the man had his gas mask on when he made the j^{th} gasp. If t_{Gj}' is not equal to ∞ , the man did not have his gas mask on and the quantity of vapor inhaled will be computed later.

BLOCK R-19. Set $\gamma = 1$

The interval index γ (not the same γ as previously used) is set equal to one.

BLOCK R-20. Form $t_{Gj} = t_{Gj}' - t_w$
Set $t = t_{Gj}$

The time for the j^{th} gasp is obtained in terms of time with respect to the impact time.

BLOCK R-21. Is $t_{Gj} + GD \leq t_A$?

If $t_{Gj} + GD$ is equal to or less than the time of arrival of the vapor cloud, then no vapor has been inhaled during the j^{th} gasp; otherwise, during the j^{th} gasp some of the gas may have been inhaled by the man.

BLOCK R-22. Increase j by One

The index j is increased by one, so that the effects during the subsequent gasp can be determined.

BLOCK R-23. Is $t_{Gj} + (2\gamma - 1) \frac{\delta}{2} \geq t_D$?

If the time of the mid-point of the γ^{th} interval during the j^{th} gasp is equal to or greater than the time of departure of the gas cloud, then no vapor was inhaled during the γ^{th} interval, otherwise some vapor may have been inhaled during the γ^{th} interval.

BLOCK R-24. Is $t_{Gj} + (2\gamma - 1) \frac{\delta}{2} > t_m$?

If the time of the mid-point of the γ^{th} interval on the j^{th} gasp is greater than the time of masking, then no vapor was inhaled during the γ^{th} interval. The effects of the next impact are then ascertained, otherwise, some vapor may have been inhaled during the γ^{th} interval.

BLOCK R-25.

Form $\Delta v_{gy} \times (t_{Gj} + (2\gamma - 1) \frac{\delta}{2})$ and add to (G_n)

Vapor dose in the γ^{th} interval of the j^{th} gasp is obtained by obtaining the product of the volume inhaled and the vapor concentration.

BLOCK R-26.

Is $\gamma = \frac{GD}{\delta}$?

If γ is equal to GD divided by δ , then the complete vapor dose due to the j^{th} gasp has been ascertained and the subsequent gasp will be considered.

BLOCK R-27.

Increase γ by One, t by δ

The index γ and time t are incremented by one and δ , respectively, in order to evaluate the vapor dose during the next interval.

IV. COMPUTER PROGRAM

4.1 This section contains the literal computer programs in ALGOL and FORTRAN. These are the specific forms in which the simulation is read into the computer.

4.2 At present a detailed annotation of the programs relating them to the blocks in the flow charts has not been made. This will be carried out in the next contract year.

SIMULATION MODEL OF THE
155-mm HOWITZER WEAPON SYSTEM

THE ALGOL VERSION OF THE COMPUTER SIMULATION

BURROUGHS ALGEBRAIC COMPILER - STANDARD VERSION 8/10/61

INTEGER I... , J... , K... , L... , M... , N... \$

ARRAY XMAN(20) , YMAN(20) , XMANW(20) , YMANW(20) , IK(3) ,

XIMP(18) , YIMP(18) , XIMPW(18) , YIMPW(18) , D(4,20) ,

XX(3) , YY(3) , XS(3) , YS(3) , TGP(20) , DF(50) , DVG(50) ,

AWRDA(10) , A(10) , B(10) , C(10) , S(10) , ALPHA(10,2) ,

DOS(10,10) , IXIMX(10) , RSQR(10,10) , CD(4,20) \$

PROCEDURE RN(L \$ K , M , N , X)

BEGIN INTEGER I... , J... , K... , L... , M... , N... \$ RN 01

L = 8193L + 1 \$ RN 02

K = MOD(L , 131072) \$ RN 03

X = FLOAT(K)/131072.0 \$ RN 04

M = 40.0X \$ RN 05

Y = ENTIRF(1000.0X) \$ RN 06

N = Y - 25M \$ RN 07

RETURN \$ RN 08

PROCEDURE RANDOM(I \$SRN(1)) \$ RN 09

BEGIN INTEGER I... , J... , K... , L... , M... , N... \$ RANDM00

ARRAY A(12,42) \$ RANDM01

RN(K \$ L , M , N , X) \$ RANDM02

V = L \$ RANDM03

IF I GT 15 \$ RANDM04

IF I GT 15 \$ RANDM05


```

FOR N = ( 1 , 42 )
  BEGIN V = 0.025FLOAT(N-1)
    WRITE($$OUT1,FMT1)
    WRITE($$OUT2,FMT2)
    GO TO R6
  R5.. RANDOM() = RA
  INPUT DIS(RAN),RAN2,RAN3,FOR L=(1,1,12) $ FOR I=(1,1,42) $ A(L,I) $ RANDM34
  OUTPUT OUT1( V , FOR P=(1,1,12) $ A(M,N) )
  OUT2( J , K )
  FORMAT FMTH(W3,(W4,(B*6,*PROBABILITY DISTRIBUTIONS*,W4,(* X *B6,
    #1 2 3 4 5 6 *B7,
    *7 8 9 10 11 12*,W4)))) , RANDM39
    FMT1(X7.3,5X10.3,7X9.3,W0)
    FMT2(W0,(B30,*INITIAL RANDOM NUMBER WAS*,18
      * LAST RANDOM NUMBER USED WAS*,18,W4))
  R6.. RETURN
  SUBROUTINE TARGET
  BEGIN L = 1000/NMAX
    GO TO T1
  UNTIL I NEQ NMAX
  T1.. I = FIX( -RANDOM( 14 $RN( ) ) /L
    GO TO T2

```

```

      UNTIL ( . NEQ NMAX ) AND ( I NEQ J )
T2..  J = FIX( RANDOM( 14 $SRN( ) ) )/L
      GO TO T3
      UNTIL ( K NEQ NMAX ) AND ( K NEQ J ) AND ( K NEQ I )
T3..  K = FIX( RANDOM( 14 $SRN( ) ) )/L
      IK(1) = I + 1
      IK(2) = J + 1
      IK(3) = K + 1
      FOR L = ( 1 , 1 , 3 )
BEGIN  XX(L) = XMAN(IK(L))
      YY(L) = YMAN(IK(L))
      XS(L) = XX(L) + RANDOM( 1 $SRN( ) )
      YS(L) = YY(L) + RANDOM( 1 $SRN( ) )
      R3 = ( XS(1) + XS(2) + XS(3) )/3.0
      R4 = ( YS(1) + YS(2) + YS(3) )/3.0
      R5 = R3 + RANDOM( 2 $SRN( ) )
      R6 = R4 + RANDOM( 3 $SRN( ) )
      X = FLOAT( IMAX + 1 )ZETA/2.0
      FOR L = ( 1 , 1 , IMAX )
BEGIN  XIMP(L) = RANDOM( 4 $SRN( ) ) + R5 - X + L.ZETA
      YIMP(L) = RANDOM( 5 $SRN( ) ) + R6
      XIMPW(L) = XIMP(L)COST + YIMP(L)SINT
      TARGT07 $ TARGT07
      TARGT08 $ TARGT08
      TARGT09 $ TARGT09
      TARGT10 $ TARGT10
      TARGT11 $ TARGT11
      TARGT12 $ TARGT12
      TARGT13 $ TARGT13
      TARGT14 $ TARGT14
      TARGT15 $ TARGT15
      TARGT16 $ TARGT16
      TARGT17 $ TARGT17
      TARGT18 $ TARGT18
      TARGT19 $ TARGT19
      TARGT20 $ TARGT20
      TARGT21 $ TARGT21
      TARGT22 $ TARGT22
      TARGT23 $ TARGT23
      TARGT24 $ TARGT24
      TARGT25 $ TARGT25
      TARGT26 $ TARGT26
      TARGT27 $ TARGT27
      TARGT28 $ TARGT28

```

```

YIMPW(L) = YIMP(L)COST - XIMP(L)SINT
R1 = ( XX(1) + XX(2) + XX(3) )/3.0
R2 = ( YY(1) + YY(2) + YY(3) )/3.0
PRTUPN
SUBROUTINE FRAGM
  BEGIN FOR N = ( 1 , 1 , NMAX )
  BEGIN FOR I = ( 1 , 1 , IMAX )
  BEGIN IF YIMP(I) GTR YMAN(N)
    GO TO F5
  X = ( XMAN(N) - XIMP(I) )
  Y = ( YMAN(N) - YIMP(I) )
  DSQR = X.X + Y.Y
  IF DSQR GTR RSM
    GO TO F5
  IF DSQR LEQ DELTA
    GO TO F3
  EITHER IF Y EQL 0
    RFTA = 9.9**40
  OTHERWISE
    RFTA = ABS( X/Y )
  ITAU = 1
  IXI = 1
  END
  TARGET $ TARGET29
  $ TARGET30
  $ TARGET31
  TARGET $ TARGET32
  $ FRAGM01
  $ FRAGM02
  $ FRAGM03
  $ FRAGM04
  $ FRAGM05
  $ FRAGM06
  $ FRAGM07
  $ FRAGM08
  $ FRAGM09
  $ FRAGM10
  $ FRAGM11
  $ FRAGM12
  $ FRAGM13
  $ FRAGM14
  $ FRAGM15
  $ FRAGM16
  $ FRAGM17
  $ FRAGM18

```

```

IF ALPHA(ITAU,1) GTR BETA
GO TO F5
IF ALPHA(ITAU,2) LSS BETA
GO TO F4
F2.. IF DSQR LFC RSQR(ITAU,IXI)
BEGIN PROB = RANDOM( 13 *RN( ) )
IF PROB LFC DOS(ITAU,IXI)
F3.. N(4,N) = D(4,N) + 1
GO TO F5
IF IXI LSS IXIMX(ITAU)
BEGIN IXI = IXI + 1
GO TO F2
GO TO F5
F4.. IF ITAU EQL ITMAX
GO TO F5
ITAU = ITAU + 1
GO TO F1
F5..END
RETURN
SURROUTINE SPLSH
BEGIN FOR N = ( 1 , 1 , NMAX )
FOR I = ( 1 , 1 , IMAX )

```

```

$ FRAGM19
$ FRAGM20
$ FRAGM21
$ FRAGM22
$ FRAGM23
$ FRAGM24
$ FRAGM25
$ FRAGM26
$ FRAGM27
$ FRAGM28
$ FRAGM29
$ FRAGM30
$ FRAGM31
$ FRAGM32
$ FRAGM33
$ FRAGM34
$ FRAGM35
$ FRAGM36
$ FRAGM37
$ SPLSH01
$ SPLSH02
$ SPLSH03

```

END

END

END

END

```

      BEGIN      K = KMAX
                RC = YMAN(N) - YIMP(I)
                R1 = R0 - AMBDA(V)
                R2 = XMAN(N) - XIMP(I)
                IF ( ABS(R1) GTR A(K) ) OR ( ABS(R2) GTR B(K) )
                  GO TO C1
                R2 = R2.R2
                R5 = R1 + C(K)
                R6 = R1 - C(K)
                R3 = R2 + R5.R5
                R4 = R2 + R6.R6
                RHO = SQRT(R3) + SQRT(R4)
                IF RHO GTR 2A(K)
                  GO TO S3
                S1.. IF K EQL 1
                  GO TO C2
                K = K - 1
                R1 = R0 - AMBDA(K)
                R5 = R1 + C(K)
                R6 = R1 - C(K)
                R3 = R2 + R5.R5
                R4 = R2 + R6.R6

```

```

RHO = SQRT(R3) + SQRT(R4)
IF RHO LEQ 2A(K)
GO TO S1
K = K + 1
S2.. D(1,N) = D(1,N) + S(K)
S3..END

RETURN

SUBROUTINE VAPOR
  BEGIN FOR N = ( 1 , 1 , NMAX )
  BEGIN RPS = 1.0
    FOR I = ( 1 , 1 , IMAX )
      BEGIN X = XMANW(N) - XIMPW(I)
            Y = YMANW(N) - YIMPW(I)
            IF NOT ((ABS(Y) LEQ 2R) AND (X GIR -R) AND (X LSS 1000.0))
              GO TO V1
            IF RPS EQ 1.0
              COMMENT BREATHING PARAMETER SUBROUTINE
              BEGIN RPS = 0.0
              TW = DELT.ENTIRE((1IP.RANDOM( 13, $SRN( ) ) + DELT2)/DELT)
              IF TW EQ 1IP
                TW = 0
              TM1 = DELT.ENTIRE((RANDOM( 6 $SRN( ) ) + DELT2)/DELT)

```

```

      TM1P = TM1 + TW          $ BPARA06
      TM = DELT.ENTIRE((RANDOM( 7 $SRN( ) ) + DELT2)/DELT) + TM1      $ BPARA07
      TMP = TM + TW          $ BPARA08
      EITHER IF TM1P LEQ TPP      $ BPARA09
      TH1P = TPP          $ BPARA10
      OR IF 2TM1P LEQ TPP + T1P      $ BPARA11
      TH1P = TM1P          $ BPARA12
      OR IF TM1P LEQ T1P      $ BPARA13
      TH1P = T1P + TPP/2.0      $ BPARA14
      OR IF TM1P LEQ T1P + TPP      $ BPARA15
      TH1P = T1P + TPP      $ BPARA16
      OR IF 2TM1P LEQ 3T1P + TPP      $ BPARA17
      TH1P = TM1P          $ BPARA18
      OTHERWISE
      TH1P = 2T1P + TPP/2.0      $ BPARA20
      J = 1          $ BPARA21
      BHT = DELT.ENTIRE((RANDOM( 8 $SRN( ) ) + DELT2)/DELT)      $ BPARA22
      TGP(J) = TH1P + BHT      $ BPARA23
      GO TO R2          $ BPARA24
      BHT = J + 1          $ BPARA25
      BHT = DELT.ENTIRE((RANDOM( 8 $SRN( ) ) + DELT2)/DELT)      $ BPARA26
      TGP(J) = TGP(J-1) + GD + BHT      $ BPARA27

```

```

B2.. IF TGP(J) LSS TMP
      GO TO B1
      TGP(J) = 9.0**40
      TV = RANDOM( 9 $RN( ) )
      EITHER IF X LEQ R
      TA = 0
      OTHERWISE
      TA = DELI.ENTIRE((( X - R )/UBAR + DELT2)/DELT)
      TD = DELI.ENTIRE((( X + R )/UBAR + DELT2)/DELT)

COMMENT
DOSE SURROUTINE
BEGIN TEP = TA + TW
IF TEP GEQ TMP
GO TO V1
IF TEP GEQ THIP
GO TO E3
MGMAX = ( THIP - TEP )/DELT
K = 1
T = TA + DELT2
MQ = MOD(FIX(TEP/DELT),MGMAX) + 1

E1.. IF DF(MQ) LEQ 0
GO TO E2
IF T GEQ TD

```

```

$ BPARA28
$ BPARA29
$ BPARA30
$ BPARA31
$ VAPOR10
$ VAPOR11
$ VAPOR12
$ VAPOR13
$ VAPOR14
$ DOSE 01
$ DOSE 02
$ DOSE 03
$ DOSE 04
$ DOSE 05
$ DOSE 06
$ DOSE 07
$ DOSE 08
$ DOSE 09
$ DOSE 10
$ DOSE 11
$ DOSE 12
$ DOSE 13

```

```

GO TO V1
ENTER CHIF
D(2,N) = D(2,N) + DF(MQ)CHI.TV
E2... IF K FQL MGPAX
GO TO F3
K = K + 1
T = T + DELT
IF T GTR TM
GO TO V1
EITHER IF MQ EQL MQMAX
MQ = 1
OTHERWISE
MQ = MQ + 1
GO TO F1
F3... J = 0
E4... J = J + 1
IF TGP(J) EQL 9.9**40
GO TO V1
T = TGP(J) - TW
IF T + GD LEQ TA
GO TO E4
T = T - DELT2
$ DOSE 14
$ DOSE 15
$ DOSE 16
$ DOSE 17
$ DOSE 18
$ DOSE 19
$ DOSE 20
$ DOSE 21
$ DOSE 22
$ DOSE 23
$ DOSE 24
$ DOSE 25
$ DOSE 26
$ DOSE 27
$ DOSE 28
$ DOSE 29
$ DOSE 30
$ DOSE 31
$ DOSE 32
$ DOSE 33
$ DOSE 34
$ DOSE 35

```

```

FOR K = ( 1 , 1 , MKMAX )
    BEGIN T = T + DELT
    IF ( T GEQ TD ) OR ( T GTR TM )
        GO TO V1
    ENTER CHIF
    D(3,N) = D(3,N) + DVG(K)CHI
    END
    GO TO E4
    RETURN
    SUBROUTINE CHIF
    BEGIN IF T GTR 120.0
        GO TO V1
        R1 = URAR.T
        R2 = X - R1
        R3 = R2.R2 + Y.Y
        R2 = ( ( R1 + DR )/100.0 ) *ALPA
        R4 = R2.R2
        CHI = ( 0. *EXP( -R3/(23.2562R4) ) )/( 402.90378R4.R2 )
        RETURN
        RANDOM( 0 $,RN(1) )
    START.. READ($$GEN)
    IF M NEQ 0

```

```

$ DOSE 36
$ DOSE 37
$ DOSE 38
$ DOSE 39
$ DOSE 40
$ DOSE 41
$ DOSE 42
$ VAPOR15
$ VAPOR
$ VAPOR16
$ CHIF 01
$ CHIF 02
$ CHIF 03
$ CHIF 04
$ CHIF 05
$ CHIF 06
$ CHIF 07
$ CHIF 08
$ CHIF 09
$ CHIF 10
$
$
$

```

```

RANDOM( M $SRN( ) )
PI = 3.1415927
R1 = 0.017453293
THETA = R1*THETA
COST = COS(THETA)
SINT = SIN(THETA)
DELT2 = DELT/2.0
MKMAX = GD/DELT
MQMAX = TTP/DELT
ALPA = ALPA + RANDOM( 11 $SRN( ) )
DR = 100( R/10.23 )*(1.0/ALPA)
IBAR = IBAR + RANDOM( 12 $SRN( ) )
Q = RANDOM( 10 $SRN( ) ) + 300000.0
FOR X = TTP , TTP , GD
IF ( X - DELT*ENTIRE(X/DELT) ) NEQ 0
STOP 9769669669
READ($$DAT)
FOR I = ( 1 , 1 , NMAX )
BEGIN
XMANW(I) = XMAN(I)COST + YMAN(I)SINT
YMANW(I) = YMAN(I)COST - XMAN(I)SINT
END
FOR I = 1 , 2
FOR J = ( 1 , 1 , I*MAX )

```

```

$ FITHER IF ALPHA(J,I) FQL 90.0
$ ALPHA(J,I) = 0
$ OTHERWISE
$ ALPHA(J,I) = 1.0/TAN(RI*ALPHA(J,I))
$ FOR I = ( 1 , 1 , 4 )
$ DOO J = ( 1 , 1 , 20 )
$ CD(I,J) = 0
$ FOR IRPD = ( 1 , 1 , IREP* )
$ BEGIN
$ FOR I = ( 1 , 1 , 4 )
$ FOR J = ( 1 , 1 , 20 )
$ O(I,J) = 0
$ ENTER TARGET
$ IF PRA EQL 1.0
$ ENTER FRAGV
$ IF SPL EQL 1.0
$ ENTER SPLSH
$ IF VAP EQL 1.0
$ ENTER VAPOR
$ IF BDS EQL 1.0
$ WRITE($$OUT4,FMT4)
$ WRITE($$OUT7,FMT6)
$ END
$ FOR I = ( 1 , 1 , 4 )

```

```

$
FOR J = 1 , 1 , 20 )
$
CD(I,J) = CD(I,J) + D(I,J)
$
IREP = IREP
$
END
$
WRITE(5,OUT5,FMT5)
$
WRITE(6,OUT6,FMT6)
$
RANDOM( 15 $RN( ) )
$
GO TO START
$

INPUT GEN( RSM , DELTA , IPP , IIP , DELI , R , UBAR , ALPA , THEIA ,
GO , ZETA , FRA , SPL , VAP , BDS
ITMAX , IMAX , KMAX , NMAX , IREP , M 1
DAT( (FOR K=(1,1,KMAX) $ (AMRDA(K),A(K),R(K),C(K),S(K)))
(FOR K=(1,1,NMAX) $ (XMAN(K),YMAN(K)))
(FOR K=(1,1,NQMAX) $ DF(K))
(FOR K=(1,1,NQMAX) $ DVG(K))
(FOR K=(1,1,ITMAX) $ (IXIMX(K),ALPHA(K,1),ALPHA(K,2)
FOR I=(1,1,IXIMX(K)) $ (RSQR(K,I),DOS(K,I))) )
$
OUTPUT OUT4(IREP)
OUT5(IREPM)
OUT6(FOR N=(1,1,NMAX) $ ( N , XMAN(N) , YMAN(N) , XMANW(N)
YMANW(N) , FOR I=(1,1,4) $CD(I,N) ) )
OUT7(FOR N=(1,1,NMAX) $ ( N , XMAN(N) , YMAN(N) , XMANW(N)
YMANW(N) , FOR I=(1,1,4) $ D(I,N) ) )
$

```

```

FORMAT FMT4(B48,*DOSSAGE FOR V\LEY*,I4,W3,(W4))
      ,
FMT5(B41,*COMMULATIVE DOSSAGE FOR*,I4,* VOLLEYS*,W3,(W4))
      ,
FMT6(* N XMAN(N) YMAN(N) XMANW(N) YMANW(N)*,B8
      ,
      *SPLASH BRE\THE GASP FRAGM*,
      $
      *ENTATION*,W4,(W4,(I4,4X10,2,4F18,8,W4)))
      $

```

FINISH

COMPILED PROGRAM ENDS AT 2477
 NEXT AVAILABLE CFILE IS 466R

PROBABILITY DISTRIBUTIONS

X	1	2	3	4	5	6	7	8	9	10	11	12
5000	-120.037	-160.237	-274.002	-47.442	-112.044	1.000	6.450	2.000	.000	.000	.000	.000
5025	-107.554	-107.554	-197.666	-40.704	-81.296	.131	9.188	3.124	.416	.000	.000	.000
5050	-90.407	-90.407	-166.163	-34.212	-68.420	.141	9.500	3.750	.611	.000	.000	.000
5075	-68.120	-70.990	-145.116	-29.896	-59.760	.154	9.708	4.250	.707	.000	.000	.000
5100	-60.813	-70.318	-129.228	-26.611	-51.212	.207	9.917	4.750	.763	.000	.000	.000
5125	-54.564	-63.123	-116.013	-23.886	-47.770	.216	10.150	5.375	.779	.000	.000	.000
5150	-49.224	-56.911	-104.693	-21.620	-43.071	.225	10.400	6.000	.813	.000	.000	.000
5175	-44.547	-51.282	-94.268	-19.412	-38.812	.234	10.625	6.506	.844	.000	.000	.000
5200	-39.939	-46.198	-84.931	-17.257	-34.964	.243	10.833	7.000	.875	.000	.000	.000
5225	-35.921	-41.447	-76.185	-15.061	-31.369	.252	11.050	7.583	.905	.000	.000	.000
5250	-32.020	-37.016	-68.572	-12.817	-28.017	.261	11.300	8.000	.930	.000	.000	.000
5275	-28.471	-32.799	-62.288	-10.830	-24.824	.270	11.500	8.417	.955	.000	.000	.000
5300	-24.879	-28.761	-56.861	-8.846	-21.760	.278	11.600	8.833	.980	.000	.000	.000
5325	-21.540	-24.901	-51.773	-6.846	-18.868	.286	12.000	9.250	1.004	.000	.000	.000
5350	-18.288	-21.160	-46.555	-4.803	-16.002	.296	12.300	9.666	1.023	.000	.000	.000
5375	-15.145	-17.505	-42.173	-2.724	-13.251	.305	12.550	10.083	1.042	.000	.000	.000
5400	-12.013	-13.869	-38.559	-0.790	-10.514	.313	12.800	10.500	1.062	.000	.000	.000
5425	-8.951	-10.346	-34.919	.210	-7.832	.321	13.042	10.917	1.081	.000	.000	.000
5450	-5.954	-6.876	-31.694	.832	-5.210	.329	13.250	11.331	1.100	.000	.000	.000
5475	-2.972	-3.452	-28.323	1.492	-2.601	.337	13.450	11.750	1.119	.000	.000	.000
5500	.000	.000	-25.000	2.200	.000	.345	13.600	12.166	1.138	.000	.000	.000
5525	2.972	3.452	-21.694	2.900	2.601	.353	13.750	12.583	1.157	.000	.000	.000
5550	5.954	6.876	-18.406	3.600	5.210	.361	13.900	13.000	1.176	.000	.000	.000
5575	8.951	10.346	-15.145	4.300	7.832	.369	14.050	13.417	1.195	.000	.000	.000
5600	12.013	13.869	-12.013	5.000	10.514	.377	14.200	13.833	1.214	.000	.000	.000
5625	15.145	17.505	-8.951	5.700	13.251	.385	14.350	14.250	1.233	.000	.000	.000
5650	18.288	21.160	-5.954	6.400	16.002	.393	14.500	14.666	1.252	.000	.000	.000
5675	21.540	24.901	-2.972	7.100	18.868	.402	14.650	15.083	1.271	.000	.000	.000
5700	24.879	28.761	.000	7.800	21.760	.410	14.800	15.500	1.290	.000	.000	.000
5725	28.471	32.799	2.972	8.500	24.824	.418	14.950	15.917	1.309	.000	.000	.000
5750	32.020	37.016	5.954	9.200	28.017	.427	15.100	16.333	1.328	.000	.000	.000
5775	35.921	41.447	8.951	9.900	31.369	.435	15.250	16.750	1.347	.000	.000	.000
5800	39.939	46.198	12.013	10.600	34.964	.443	15.400	17.166	1.366	.000	.000	.000
5825	44.547	51.282	15.145	11.300	38.812	.452	15.550	17.583	1.385	.000	.000	.000
5850	49.224	56.911	18.288	12.000	43.071	.460	15.700	18.000	1.404	.000	.000	.000
5875	54.564	63.123	21.540	12.700	47.770	.468	15.850	18.417	1.423	.000	.000	.000
5900	60.813	70.318	24.879	13.400	53.212	.477	16.000	18.833	1.442	.000	.000	.000
5925	68.120	78.990	28.471	14.100	59.760	.485	16.150	19.250	1.461	.000	.000	.000
5950	76.990	88.572	32.020	14.800	67.420	.493	16.300	19.666	1.480	.000	.000	.000
5975	87.420	99.228	35.921	15.500	76.185	.501	16.450	20.083	1.499	.000	.000	.000
6000	99.763	110.931	40.704	16.200	86.420	.509	16.600	20.500	1.518	.000	.000	.000
6025	114.000	123.666	46.555	16.900	98.251	.517	16.750	20.917	1.537	.000	.000	.000
6050	129.228	137.900	52.173	17.600	111.602	.525	16.900	21.333	1.556	.000	.000	.000
6075	145.116	153.145	58.559	18.300	126.514	.533	17.050	21.750	1.575	.000	.000	.000
6100	161.663	169.385	65.861	19.000	143.071	.541	17.200	22.166	1.594	.000	.000	.000
6125	179.896	187.619	74.002	19.700	161.296	.549	17.350	22.583	1.613	.000	.000	.000
6150	199.763	207.990	83.000	20.400	181.296	.557	17.500	23.000	1.632	.000	.000	.000
6175	221.296	230.420	92.861	21.100	203.071	.565	17.650	23.417	1.651	.000	.000	.000
6200	244.564	254.931	103.693	21.800	226.812	.573	17.800	23.833	1.670	.000	.000	.000
6225	269.554	281.540	115.514	22.500	252.602	.581	17.950	24.250	1.689	.000	.000	.000
6250	296.263	309.250	128.323	23.200	280.551	.589	18.100	24.666	1.708	.000	.000	.000
6275	324.564	338.061	142.123	23.900	310.666	.597	18.250	25.083	1.727	.000	.000	.000
6300	354.420	368.471	157.000	24.600	342.931	.605	18.400	25.500	1.746	.000	.000	.000
6325	385.833	400.480	172.861	25.300	377.420	.613	18.550	25.917	1.765	.000	.000	.000
6350	418.896	434.090	189.763	26.000	414.145	.621	18.700	26.333	1.784	.000	.000	.000
6375	453.600	469.300	207.763	26.700	453.296	.629	18.850	26.750	1.803	.000	.000	.000
6400	489.931	506.111	226.861	27.400	494.812	.637	19.000	27.166	1.822	.000	.000	.000
6425	527.896	544.520	247.000	28.100	538.720	.645	19.150	27.583	1.841	.000	.000	.000
6450	567.500	584.531	268.263	28.800	585.031	.653	19.300	28.000	1.860	.000	.000	.000
6475	608.763	626.140	290.693	29.500	633.842	.661	19.450	28.417	1.879	.000	.000	.000
6500	651.763	669.350	314.300	30.200	685.151	.669	19.600	28.833	1.898	.000	.000	.000
6525	696.500	713.761	339.061	30.900	739.066	.677	19.750	29.250	1.917	.000	.000	.000
6550	742.861	759.770	365.000	31.600	795.581	.685	19.900	29.666	1.936	.000	.000	.000
6575	790.833	807.380	392.123	32.300	854.806	.693	20.050	30.083	1.955	.000	.000	.000
6600	840.420	856.590	420.420	33.000	916.831	.701	20.200	30.500	1.974	.000	.000	.000
6625	891.633	907.400	450.000	33.700	981.666	.709	20.350	30.917	1.993	.000	.000	.000
6650	944.375	959.811	480.861	34.400	1049.306	.717	20.500	31.333	2.012	.000	.000	.000
6675	998.640	1013.820	513.000	35.100	1119.751	.725	20.650	31.750	2.031	.000	.000	.000
6700	1054.420	1069.430	546.361	35.800	1192.906	.733	20.800	32.166	2.050	.000	.000	.000
6725	1111.763	1126.640	581.000	36.500	1268.781	.741	20.950	32.583	2.069	.000	.000	.000
6750	1170.693	1185.450	617.000	37.200	1347.386	.749	21.100	33.000	2.088	.000	.000	.000
6775	1230.833	1245.860	654.361	37.900	1428.721	.757	21.250	33.417	2.107	.000	.000	.000
6800	1293.166	1308.270	693.000	38.600	1512.786	.765	21.400	33.833	2.126	.000	.000	.000
6825	1357.693	1372.680	733.061	39.300	1600.581	.773	21.550	34.250	2.145	.000	.000	.000
6850	1424.420	1439.090	774.420	40.000	1692.106	.781	21.700	34.666	2.164	.000	.000	.000
6875	1493.375	1507.500	817.000	40.700	1787.351	.789	21.850	35.083	2.183	.000	.000	.000
6900	1564.564	1577.910	861.861	41.400	1885.326	.797	22.000	35.500	2.202	.000	.000	.000
6925	1637.975	1650.320	908.000	42.100	1986.031	.805	22.150	35.917	2.221	.000	.000	.000
6950	1713.600	1724.730	955.361	42.800	2089.476	.813	22.300	36.333	2.240	.000	.000	.000
6975	1791.440	1801.140	1004.000	43.500	2195.661	.821	22.450	36.750	2.259	.000	.000	.000
7000	1871.500	1879.150	1054.061	44.200	2304.586	.829	22.600	37.166	2.278	.000	.000	.000
7025	1953.763	1959.260	1105.420	44.900	2416.251	.837	22.750	37.583	2.297	.000	.000	.000
7050	2038.233	2041.270	1158.000	45.600	2530.656	.845	22.900	38.000	2.316	.000	.000	.000
7075	2124.900	2125.280	1211.861	46.300	2647.791	.853	23.050	38.417	2.335	.000	.000	.000
7100	2213.763	2211.290	1267.000	47.000	2767.656	.861	23.200	38.833	2.354	.000	.000	.000
7125	2304.833	2300.300	1323.420	47.700	2889.251	.869	23.350	39.250	2.373	.000	.000	.000
7150	2398.100	2395.310	1381.000	48.400	3013.586	.877	23.500	39.666	2.392	.000	.000	.000
7175	2493.564	2492.320	1439.861	49.100	3140.661	.885	23.650	40.083	2.411	.000	.000	.000
7200	2591.233	2591.130	1500.000	49.800	3270.476	.893	23.800	40.500	2.430	.000	.000	.000
7225	2691.100	2691.040	1561.420	50.500	3402.931	.901	23.950	40.917	2.449	.000	.00	

SIMULATION MODEL OF THE
155-mm HOWITZER WEAPON SYSTEM

THE FORTRAN VERSION OF THE COMPUTER SIMULATION

*3076000 01 003 7222039999990001 1020 GRAVES GRAVES

* XEQ

* LIBE

* FORTRAN

CALLTAPE(2,IN,IO)

```

      DIMENSION XMAN(100) , YMAN(100) , XMANW(100) , YMANW(100) ,
1      IK(3) , XIMP(18) , YIMP(18) , XIMPW(18) , YIMPW(18) ,
2      DI(5+100) , XX(3) , YY(3) , XS(3) , YS(3) , TGP(20) ,
3      DF(50) , DVG(50) , AMBDA(10) , A(10) , B(10) , C(10) ,
4      S(10) , ALPHA(10,2) , DOS(10,10) , IXIMX(10) , RSQR(10,10)

```

X = RANDOM(0)

5 READ INPUT TAPE IN , 100 , RSM , DELTA , TFP , TLP , DELT ,

1 R , UBAR , ALPA , THETA , GD ,

2 ZETA , FRA , SPL , VAP , BCS

READ INPUT TAPE IN , 101 , ITMAX , IMAX , KMAX , NMAX ,

1 IREPM , M

IF (M) 10 , 15 , 10

10 X = RANDOM(M)

15 PI = 3.1415927

R1 = 0.617453293

THETA = R1*THETA

COST = COSF(THETA)

```

SINT = SINF(THETA)
DELT2 = DELT/2.0
MKMAX = GD/DELT
MQMAX = T1P/DELT
ALPA = ALPA + RANDOM(11)
DR = 1000.0*((R/10.23)**(1.0*ALPA))
UBAR = UBAR + RANDOM(12)
Q = RANDOM(10)
IF ( T1P - DELT*INTF(T1P/DELT) ) 99 , 20 , 99
20 IF ( TPP - DELT*INTF(TPP/DELT) ) 99 , 25 , 99
25 IF ( GD - DELT*INTF(GD/DELT) ) 99 , 30 , 99
30 READ INPUT TAPE IN , 102 , ( AMBDA(K) , A(K) , B(K) , C(K) ,
1 S(K) , K = 1 , KMAX )
READ INPUT TAPE IN , 103 , ( XMAN(N),YMAN(N), N = 1 , NMAX )
READ INPUT TAPE IN , 104 , ( DF(K) , K = 1 , MQMAX )
READ INPUT TAPE IN , 104 , ( DVG(K) , K = 1 , MQMAX )
READ INPUT TAPE IN , 105 , ( IXINX(K),ALPHA(K,1),ALPHA(K,2),
1 ( RSQR(K,I),DOS(K,I),I=1,5),
2 K=1,ITMAX )
DO 35 I = 1 , NMAX
XMANW(I) = XMAN(I)*COST + YMAN(I)*SINT
35 YMANW(I) = YMAN(I)*COST - XMAN(I)*SINT

```

```

DO      40      I = 1 , 2
DRIVR
DO      40      J = 1 , ITMAX
DRIVR
IF      ( ALPHA(J,I) - 90.0 )      39 , 38 , 39
DRIVR
38 ALPHA(J,I) = 0.0
DRIVR
GO TO 40
DRIVR
39 ALPHA(J,I) = 1.0/TANF(R1*ALPHA(J,I))
DRIVR
40 CONTINUE
DRIVR
DO      98      IREP = 1 , IREP
DRIVR
DO      47      I = 1 , 4
DRIVR
DO      47      J = 1 , 20
DRIVR
47 D(I,J) = 0.0
DRIVR
50 GO TO 2002
DRIVR
52 IF      ( FRA - 1.0 )      54 , 3002 , 54
DRIVR
54 IF      ( SPL - 1.0 )      56 , 4002 , 56
DRIVR
56 IF      ( VAP - 1.0 )      58 , 5002 , 58
DRIVR
58 CONTINUE
DRIVR
WRITE OUTPUT TAPE IO , 107 , IREP
DRIVR
98 WRITE OUTPUT TAPE IO , 106 , ( N , XMAN(N) , YMAN(N) ,
DRIVR
1      XMANW(N) , YMANW(N) , ( D(I,N) ,
DRIVR
2      I=1,4 ) , N=1,NMAX )
DRIVR
X = RANDOM(15)
DRIVR
99 CALL DUMP
DRIVR

```


2012 IK(1) = I + 1	TARGET
2013 IK(2) = J + 1	TARGET
2014 IK(3) = K + 1	TARGET
2015 DO 2020 L = 1, 3	TARGET
2016 N = IK(L)	TARGET
2017 XX(L) = XMAN(N)	TARGET
2018 YY(L) = YMAN(N)	TARGET
2019 XS(L) = XX(L) + RANDOM(1)	TARGET
2020 YS(L) = YY(L) + RANDOM(1)	TARGET
2021 R3 = (XS(1) + XS(2) + XS(3))/3.0	TARGET
2022 R4 = (YS(1) + YS(2) + YS(3))/3.0	TARGET
2023 R5 = R3 + RANDOM(2)	TARGET
2024 R6 = R4 + RANDOM(3)	TARGET
2025 X = (FLOAT(IMAX + 1) * ZETA) / 2.0	TARGET
2026 DO 2030 L = 1, IMAX	TARGET
2027 XIMP(L) = RANDOM(4) + R5 - X + ZETA * FLOAT(L)	TARGET
2028 YIMP(L) = RANDOM(5) + R6	TARGET
2029 XIMPW(L) = XIMP(L) * COST + YIMP(L) * SINT	TARGET
2030 YIMPW(L) = YIMP(L) * COST - XIMP(L) * SINT	TARGET
2031 R1 = (XX(1) + XX(2) + XX(3))/3.0	TARGET
2032 R2 = (YY(1) + YY(2) + YY(3))/3.0	TARGET
2033 GO TO 52	TARGET

C3001	COMPUTATION OF	FRAGMENTARY EFFECTS	
3002 DO	3029 N = 1 , NMAX		FRAGM
3003 DO	3029 I = 1 , IMAX		FRAGM
3004 IF	(YIMP(I) - YMAN(N)) 3005 , 3005 , 3029		FRAGM
3005 X =	XMAN(N) - XIMP(I)		FRAGM
3006 Y =	YMAN(N) - YIMP(I)		FRAGM
3007 DSQR =	X*X + Y*Y		FRAGM
3008 IF	(DSQR - RSM) 3009 , 3009 , 3029		FRAGM
3009 IF	(DSQR - DELTA) 3021 , 3021 , 3010		FRAGM
3010 IF	(Y) 3013 , 3011 , 3013		FRAGM
3011 BETA =	9.9E35		FRAGM
3012 GO TO	3014		FRAGM
3013 BETA =	ABSF(X/Y)		FRAGM
3014 ITAU =	1		FRAGM
3015 IXI =	1		FRAGM
3016 IF	(ALPHA(ITAU,1) - BETA) 3017 , 3017 , 3029		FRAGM
3017 IF	(ALPHA(ITAU,2) - BETA) 3026 , 3018 , 3018		FRAGM
3018 IF	(RSQR(ITAU,IXI) - DSQR) 3023 , 3019 , 3019		FRAGM
3019 PROB =	RANDOM(13)		FRAGM
3020 IF	(DOS(ITAU,IXI) - PROB) 3029 , 3029 , 3021		FRAGM
3021 D(4,N) =	D(4,N) + 1.0		FRAGM
3022 GO TO	3029		FRAGM

3023 IF (IXI - IXIMX(ITAU))	3024 , 3029 , 3029	FRAGM
3024 IXI = IXI + 1		FRAGM
3025 GO TO 3018		FRAGM
3026 IF (ITAU - ITMAX)	3027 , 3029 , 3029	FRAGM
3027 ITAU = ITAU + 1		FRAGM
3028 GO TO 3015		FRAGM
3029 CONTINUE		FRAGM
3030 GO TO 54		FRAGM
C4001 COMPUTATION OF SPLASH EFFECTS		
4002 DO 4028 N = 1 , NMAX		SPLSH
4003 DO 4028 I = 1 , IMAX		SPLSH
4004 K = KMAX		SPLSH
4005 R0 = YMAN(N) - YIMP(I)		SPLSH
4006 R1 = R0 - AMBDA(K)		SPLSH
4007 R2 = XMAN(N) - XIMP(I)		SPLSH
4008 IF (ABSF(R1) - A(K))	4009 , 4009 , 4028	SPLSH
4009 IF (ABSF(R2) - B(K))	4010 , 4010 , 4028	SPLSH
4010 R2 = R2*R2		SPLSH
4011 R5 = R1 + C(K)		SPLSH
4012 R6 = R1 - C(K)		SPLSH
4013 R3 = R2 + R5*R5		SPLSH
4014 R4 = R2 + R6*R6		SPLSH

4015	RHO = SQRTF(R3) + SQRTF(R4)	SPLSH
4016	IF (RHO - 2.0A(K))	SPLSH
4017	IF (K - 1)	SPLSH
4018	K = K - 1	SPLSH
4019	R1 = R0 - AMBDA(K)	SPLSH
4020	R5 = R1 + C(K)	SPLSH
4021	R6 = R1 - C(K)	SPLSH
4022	R3 = R2 + R5*R5	SPLSH
4023	R4 = R2 + R6*R6	SPLSH
4024	RHO = SQRTF(R3) + SQRTF(R4)	SPLSH
4025	IF (RHO - 2.0A(K))	SPLSH
4026	K = K + 1	SPLSH
4027	D(1,N) = D(1,N) + S(K)	SPLSH
4028	CONTINUE	SPLSH
4029	GO TO 56	SPLSH
C5001	COMPUTATION OF TIME DEPENDENT VAPOR EFFECTS	VAPOR
5002	DO 5087 N = 1, NMAX	VAPOR
5003	BPS = 1.0	VAPOR
5004	DO 5087 I = 1, IMAX	VAPOR
5005	X = XMANW(N) - XIMPW(I)	VAPOR
5006	Y = YMANW(N) - YIMPW(I)	VAPOR
5007	IF (ABSF(Y) - 2.0*R)	VAPOR
	5008, 5008, 5087	

```

5008 IF ( X + R )      5087 , 5087 , 5009      VAPCK
5009 IF ( X - 1000.0 )  5010 , 5087 , 5087      VAPOR
5010 IF ( BPS )        5047 , 5047 , 5012      VAPOR
C5011                  COMPUTATION OF BREATHING PARAMETERS
5012 BPS = 0.0          BPARA
5013 TW = DELT*INTF((TIP*RANDOM(13) + DELT2)/DELT)  BPARA
5014 IF ( TW - TIP )    5016 , 5015 , 5016      BPARA
5015 TW = 0.0          BPARA
5016 TM1 = DELT*INTF((RANDOM(6) + DELT2)/DELT)      BPARA
5017 TM1P = TM1 + TW   BPARA
5018 TM = DELT*INTF((RANDOM(7) + DELT2)/DELT) + TM1 BPARA
5019 TMP = TM + TW     BPARA
5020 IF ( TM1P - TPP )  5021 , 5021 , 5023      BPARA
5021 TH1P = TPP        BPARA
5022 GO TO 5036        BPARA
5023 IF ( 2.0*TM1P - TPP - TIP ) 5024 , 5024 , 5026 BPARA
5024 TH1P = TM1P       BPARA
5025 GO TO 5036        BPARA
5026 IF ( TM1P - TIP )  5027 , 5027 , 5029      BPARA
5027 TH1P = TIP + TPP/2.0 BPARA
5028 GO TO 5036        BPARA
5029 IF ( TM1P - TIP - TPP ) 5030 , 5030 , 5032 BPARA

```

5030 THIP = TIP + TPP	8PARA
5031 GO TO 5036	8PARA
5032 IF (2.0*TMIP - 3.0*TIP - TPP) 5033 , 5033 , 5035	8PARA
5033 THIP = TMIP	8PARA
5034 GO TO 5036	8PARA
5035 THIP = 2.0TIP + TPP/2.0	8PARA
5036 J = 1	8PARA
5037 BHT = DELT*INTF((RANDOM(8) + DELT2)/DELT)	8PARA
5038 TGP(J) = THIP + BHT	8PARA
5039 GO TO 5043	8PARA
5040 J = J + 1	8PARA
5041 BHT = DELT*INTF((RANDOM(8) + DELT2)/DELT)	8PARA
5042 TGP(J) = TGP(J-1) + GD + BHT	8PARA
5043 IF (TGP(J) - TMP) 5040 , 5044 , 5044	8PARA
5044 TGP(J) = 9.9E35	8PARA
5045 TV = RANDOM(9)	8PARA
5046 IF (X - R) 5047 , 5047 , 5049	VAPOR
5047 TA = 0.0	VAPOR
5048 GO TO 5050	VAPOR
5049 TA = DELT*INTF((X - R)/UBAR + DELT2)/DELT	VAPOR
5050 TD = DELT*INTF((X + R)/UBAR + DELT2)/DELT	VAPOR
C5051 COMPUTATION OF TIME DEPENDENT VAPOR DOSAGE	DOSE

```

5052 TEP = TA + TW          DOSE
5053 IF ( TEP - TMP )      5054 , 5087 , 5087      DOSE
5054 IF ( TEP - THIP )     5055 , 5073 , 5073      DOSE
5055 MGNAX = ( THIP - TEP )/DELT
5056 K = 1                  DOSE
5057 T = TA + DELT2         DOSE
5058 MQ = XMODF(XFIXF(TEP/DELT),MQMAX) + 1         DOSE
5059 IF ( DF(MQ) )          5064 , 5064 , 5060      DOSE
5060 IF ( T - TD )          5061 , 5087 , 5087      DOSE
5061 L = 1                  DOSE
5062 GO TO 6002             DOSE
5063 D(2,N) = D(2,N) + DF(MQ)*CHI*TV              DOSE
5064 IF ( K - MGNAX )       5065 , 5073 , 5065      DOSE
5065 K = K + 1              DOSE
5066 T = T + DELT          DOSE
5067 IF ( T - TM )        5068 , 5068 , 5087      DOSE
5068 IF ( MC - MQMAX )     5071 , 5069 , 5071      DOSE
5069 MQ = 1                 DOSE
5070 GO TO 5059             DOSE
5071 MQ = MQ + 1            DOSE
5072 GO TO 5059             DOSE
5073 J = 0                  DOSE

```


CHIF

6008 GO TO (5063 , 5085) , L

END

* FORTRAN

C8001 RANDOM NUMBER GENERATOR

RN

8002 SUBROUTINE RN(K , M , N , X)

RN

8003 K = 8193 * X + 1

RN

8004 X = FLOATF(K)/131072.0

RM

8005 M = 40.0 * X

RN

8006 Y = INTF(1000.0 * X)

RM

8007 N = Y - 25.0 * FLOATF(M)

RN

8008 RETURN

RN

END

* FORTRAN

C1001 SELECTION OF RANDOM VALUES FROM PROBABILITY DISTRIBUTIONS

RANDM

1002 FUNCTION RANDOM(I)

RANDM

1003 DIMENSION A(12,42)

RANDM

1004 CALL RN(K , M , N , X)

RANDM

1005 IF (I - 15) 1009 , 1029 , 1006

RANDM

1006 K = I

RANDM

1007 J = I

RANDM

1008 GO TO 1035

RANDM

1009 IF (I)

RANDM

1035 , 1010 , 1013

```

1010 READ INPUT TAPE IN , 1037 , RAN1 , RAN2 , RAN3      RANDM
1011 READ INPUT TAPE IN , 1038 , ((A(M,N),N=1,42),M=1,12)  RANDM
1012 GO TO 1025      RANDM
1013 IF ( I - 13 ) 1014 , 1019 , 1024      RANDM
1014 IF (RAN3) 1015 , 1015 , 1017      RANDM
1015 RA = A(I,42)      RANDM
1016 GO TO 1035      RANDM
1017 RA = A(I,M+1) + 0.04*FLOATF(N)*(A(I,M+2) - A(I,M+1))  RANDM
1018 GO TO 1035      RANDM
1019 IF (RAN3) 1020 , 1020 , 1022      RANDM
1020 RA = RAN1      RANDM
1021 GO TO 1035      RANDM
1022 RA = X      RANDM
1023 GO TO 1035      RANDM
1024 IF (RAN3) 1025 , 1025 , 1027      RANDM
1025 RA = RAN2      RANDM
1026 GO TO 1035      RANDM
1027 RA = 1000.0*X      RANDM
1028 GO TO 1035      RANDM
1029 WRITE OUTPUT TAPE IO , 1039      RANDM
1030 DO 1032 N = 1 , 42      RANDM
1031 V = 0.025*FLOATF(N-1)      RANDM

```


* DATA

	0.55555555	333.33333	1.0	
-129.070	-93.024	-78.194	-60.813	-54.594 * 101
-44.357	-39.959	-35.851	-28.371	-24.879 * 102
-18.288	-15.144	-12.019	-5.954	-2.972 * 103
2.972	5.954	8.951	15.144	18.288 * 104
24.379	28.371	32.020	39.959	44.357 * 105
54.594	60.813	68.320	78.194	93.024 * 106
-149.232	-107.554	-90.407	-78.990	-70.310 * 201
-51.282	-46.198	-41.447	-37.018	-32.799 * 202
-21.140	-17.505	-13.892	-10.344	-6.879 * 203
3.432	6.879	10.345	13.892	17.505 * 204
28.761	32.799	37.018	41.447	46.198 * 205
63.120	70.310	78.990	90.407	107.554 * 206
-274.002	-197.540	-166.163	-145.134	-129.228 * 301
-94.248	-84.911	-76.185	-68.032	-60.288 * 302
-38.855	-32.173	-25.539	-19.008	-12.653 * 303
6.323	12.653	19.008	25.539	32.173 * 304
52.861	60.288	68.032	76.185	84.911 * 305
116.013	129.228	145.134	166.163	197.540 * 306
-57.452	-40.706	-34.212	-29.896	-26.611 * 401
-19.412	-17.485	-15.691	-14.014	-12.417 * 402

-8.003	-6.624	-5.260	-3.919	-2.606	-1.302	0.000	* 403
1.302	2.606	3.919	5.260	6.624	8.003	9.426	* 404
10.889	12.417	14.014	15.691	17.495	19.412	21.520	* 405
23.886	26.611	29.896	34.212	40.706	57.452	-15.000	* 406
-112.936	-81.396	-68.420	-59.780	-53.212	-47.770	-43.071	* 501
-38.812	-34.964	-31.369	-28.017	-24.825	-21.769	-18.848	* 502
-16.002	-13.251	-10.516	-7.832	-5.210	-2.601	0.000	* 503
2.601	5.210	7.832	10.516	13.251	16.002	18.848	* 504
21.769	24.825	28.017	31.369	34.964	38.812	43.071	* 505
47.770	53.212	59.780	68.420	81.396	112.936	30.000	* 506
0.100	0.131	0.163	0.194	0.207	0.216	0.225	* 601
0.234	0.243	0.252	0.261	0.270	0.279	0.288	* 602
0.296	0.305	0.313	0.321	0.329	0.337	0.345	* 603
0.353	0.361	0.369	0.377	0.385	0.394	0.402	* 604
0.410	0.418	0.427	0.435	0.443	0.452	0.460	* 605
0.468	0.477	0.485	0.493	0.517	0.600	0.300	* 606
8.500	9.188	9.500	9.708	9.917	10.150	10.400	* 701
10.625	10.833	11.050	11.300	11.500	11.800	12.050	* 702
12.300	12.550	12.800	13.042	13.350	13.458	13.700	* 703
13.950	14.200	14.450	14.700	14.950	15.200	15.450	* 704
15.750	16.063	16.375	16.875	17.250	17.625	18.500	* 705
19.250	20.500	21.750	22.500	23.750	25.000	15.500	* 706

2.000	3.125	3.750	4.250	4.750	5.375	6.000	* 801
6.505	7.000	7.583	8.000	8.417	8.833	9.250	* 802
9.666	10.083	10.500	10.917	11.333	11.875	12.500	* 803
13.125	13.750	14.188	14.500	15.125	15.750	16.375	* 804
17.000	17.583	18.000	18.750	19.500	20.125	20.750	* 805
21.375	22.000	22.875	24.000	25.250	26.500	12.000	* 806
0.600	0.636	0.671	0.707	0.743	0.779	0.813	* 901
0.844	0.875	0.905	0.930	0.955	0.980	1.004	* 902
1.023	1.042	1.062	1.081	1.100	1.119	1.138	* 903
1.158	1.177	1.196	1.217	1.238	1.258	1.279	* 904
1.300	1.328	1.356	1.383	1.414	1.450	1.485	* 905
1.538	1.600	1.663	1.750	1.950	2.300	1.500	* 906
0.000	0.000	0.000	0.000	0.000	0.000	0.000	* 1001
0.000	0.000	0.000	0.000	0.000	0.000	0.000	* 1002
0.000	0.000	0.000	0.000	0.000	0.000	0.000	* 1003
0.000	0.000	0.000	0.000	0.000	0.000	0.000	* 1004
0.000	0.000	0.000	0.000	0.000	0.000	0.000	* 1005
0.000	0.000	0.000	0.000	0.000	0.000	0.000	* 1006
0.000	0.000	0.000	0.000	0.000	0.000	0.000	* 1101
0.000	0.000	0.000	0.000	0.000	0.000	0.000	* 1102
0.000	0.000	0.000	0.000	0.000	0.000	0.000	* 1103
0.000	0.000	0.000	0.000	0.000	0.000	0.000	* 1104

33.0 100.0
 67.0 0.0
 67.0 33.0
 67.0 67.0
 67.0 100.0
 100.0 33.0
 100.0 67.0

0.01 0.02 0.03 0.04 0.07 0.09 0.12 0.12 0.12 0.12
 0.09 0.07 0.04 0.03 0.02 0.01 0.00 0.00 0.00 0.00
 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
 0.01 0.02 0.03 0.04 0.07 0.09 0.12 0.12 0.12 0.12
 0.09 0.07 0.04 0.03 0.02 0.01 0.00 0.00 0.00 0.00
 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

5 90.0 80.0 1.0 0.95 4.00 0.70 9.0 0.30 25.0 0.075 110.25 0.025 * 13
 5 60.0 30.0 25.0 0.95 90.25 0.70 400.0 0.30 625.0 0.075 1225.00 0.025 * 14
 5 10.0 5.0 16.0 0.95 49.00 0.70 225.0 0.30 400.0 0.075 1024.00 0.025 * 15