THE ENRICHMENT OF SMOLER'S MODEL OF LAND COMBAT. (U)

SEP 80  G N MILLS

END
THESIS

THE ENRICHMENT OF SIMLER'S MODEL OF LAND COMBAT

by

Glenn M. Mills

Sept 1980

Thesis Advisor: J. G. Taylor

Approved for public release; distribution unlimited
The Enrichment of Smoler's Model of Land Combat

Glenn M. Mills

Naval Postgraduate School
Monterey, California 93940

Naval Postgraduate School
Monterey, California 93940

Approved for public release; distribution unlimited.

Lanchester-type ground combat model.

This thesis provides the student of combat modelling with a computer program for a relatively simple combat model that can be used in a classroom environment for study and analysis. The model is an aggregated, force-on-force ground-combat model that uses...
Lanchester's aimed-fire equations for casualty assessment. The original version of the model was developed in 1979 in a previous thesis, and shortfalls in the original model have been overcome, along with the addition of several enrichments to provide added user flexibility. A user's manual is provided to facilitate user access to the model from a permanent disk in the W. R. Church Computer Center.
The Enrichment of Smoler's
Model of Land Combat

by

Glenn M. Mills
Captain, United States Army
B.S., North Carolina State University, 1971

Submitted in partial fulfillment of the requirements for the degree of

MASTERS OF SCIENCE IN OPERATIONS RESEARCH

from the

NAVAL POSTGRADUATE SCHOOL
September 1980

Author

Approved by

Thesis Advisor

Second Reader

Chairman, Department of Operations Research

Dean of Information and Policy Sciences
ABSTRACT

This thesis provides the student of combat modelling with a computer program for a relatively simple combat model that can be used in a classroom environment for study and analysis. The model is an aggregated, force-on-force ground-combat model that uses Lanchester's aimed-fire equations for casualty assessment. The original version of the model was developed in 1979 in a previous thesis, and shortfalls in the original model have been overcome, along with the addition of several enrichments to provide added user flexibility. A user's manual is provided to facilitate user access to the model from a permanent disk in the W. R. Church Computer Center.
## TABLE OF CONTENTS

I. INTRODUCTION ................................................. 6

II. THE ORIGINAL MODEL ........................................... 9
   A. GENERAL DESCRIPTION ....................................... 9
   B. ATTRITION PROCESS ........................................ 9
   C. UNIT LOCATION AND MOVEMENT .............................. 11
   D. BATTLE TERMINATION ....................................... 12
   E. DETECTION AND FIRE ALLOCATION .......................... 13

III. MODEL CHANGES ................................................ 15
   A. GENERAL ...................................................... 15
   B. CONDITIONALLY-DETERMINISTIC ATTRITION ................. 15
   C. UNIT LOCATION AND MOVEMENT .............................. 20
      1. Straight Line Routes .................................... 21
      2. User Determined Routes ................................... 21
   D. BATTLE TERMINATION ....................................... 23
   E. ALTERNATE DEFENSIVE POSITIONS ........................... 25

IV. FUTURE ENHANCEMENT ........................................... 27
   A. HETEROGENEOUS FORCES ..................................... 27
   B. AMMUNITION EXPENDITURE ................................... 28
   C. ARTILLERY .................................................... 29

APPENDIX A: USER'S MANUAL ....................................... 31
APPENDIX B: PROGRAM LISTING ..................................... 49
APPENDIX C: DEFINITION OF VARIABLES IN COMPUTER PROGRAM 77
BIBLIOGRAPHY ....................................................... 81
INITIAL DISTRIBUTION LIST ....................................... 82
I. INTRODUCTION

As the art of combat modelling becomes more advanced, combat modelers are continually building more and more complicated models. To the beginning modeler, the ability to understand how these models operate is difficult, if not impossible. As a student of combat modelling, I have sensed a need for a relatively simple model that could be easily studied in a classroom environment. It was with this in mind that this project was initiated.

At present, there seems to be no simple combat model available that demonstrates the basics of model building to the beginning student. In 1979, J. Smoler (a student at the Naval Postgraduate School) attempted to build such a model for his thesis research (Smoler, 1979). His model was a deterministic, force-on-force computer model that used Lanchester's aimed-fire equations for casualty assessment. The general scheme of his model is shown in Figure 1.

Although Smoler's model was a bold attempt at a simplistic combat model, it did have some problem areas that warranted investigation. Some of these problems were first discovered during a class project in a combat models class (OA 4655) at the Naval Postgraduate School. It is the purpose of this thesis to study Smoler's model and to undertake an enrichment program that will make the model
START

INITIALIZATION

TIME = 0

TIME = TIME + 10 sec

MOVEMENT

DETECTION

FIRE-ALLOCATION

ATTRITION

NO

END OF BATTLE

YES

END

FIGURE 1. GENERAL SCHEME
more realistic and flexible, while maintaining its transparency and simplicity.

Once enhanced, the model would then be made available for use as an instructional tool for combat modelling classes. This will be accomplished through the development of a user's manual to facilitate the use of the model by students by placing the model on a permanent disk in the W. R. Church Computer Center where it will be easily accessed by any desired user.

The remainder of this paper will discuss, in detail, the problems found with Smoler's model and the methods used to solve these problems. It will also outline some new features that have been incorporated into the model to allow user flexibility. Finally, the User's Manual (Appendix A) will provide all the required information to enable even the novice user to utilize the model.
II. THE ORIGINAL MODEL

A. GENERAL DESCRIPTION

Smoler's original model of land combat is a deterministic model that plays combat between two homogeneous forces, a blue force and a red force. The blue force is comprised of three subunits in a static defense, with each subunit armed with three TOW antitank missile systems. The red force is composed of three subunits of three tanks each, attacking on pre-planned routes. The battle takes place on the 10 x 10 Km Fulda Box that has been developed and used in the STAR simulation model (Wallace, 1978). Since the major thrust of this paper is to alter the original model to a form that will be easily used for classroom instruction, a brief discussion of the major components of the original model will be presented, including problem areas that have motivated model changes.

B. ATTRITION PROCESS

The attrition process in the original model utilizes Lanchester "aimed-fire" equations with variable attrition coefficients. The Lanchester equations used are fairly simple and will be discussed later. However, the calculation of the attrition coefficients is of more immediate interest. Smoler used the Bonder-Farrell model to calculate the coefficients, $A_{ij}$, the rate at which one firer...
of unit i kills unit j targets. These $A_{ij}$'s are computed according to:

$$A_{ij} = \frac{1}{E(T_{ij})}$$

where $E(T_{ij})$ is the expected time for one firer of unit i to kill one target of unit j. The $E(T_{ij})$ is computed using the Bonder-Farrell formula:

$$E(T_{ij}) = t_a + t_l + t_h + \frac{(t_h + t_f)}{P(K|H)} + \frac{(t_m + t_f)}{P(h|m)} \times \frac{(1-P(h|h))/P(K|H) + P(h|h) - P)}$$

where

- $t_a =$ time to acquire a target
- $t_l =$ time to fire first round following acquisition
- $t_h =$ time to fire following a hit
- $t_m =$ time to fire following a miss
- $t_f =$ time of flight of a round
- $P =$ probability of a first round hit
- $P(h|h) =$ probability of a hit following a hit
- $P(h|m) =$ probability of a hit following a miss
- $P(K|H) =$ probability of a kill given a hit

This formula holds for the conditions that the hit probability of any round depends only on the result of the previous round and no accumulated damage is considered.

Smoler also assumed that for the TOW weapon system, $P(K|H) = 1.0$ and $P(h|m) = P(h|h) = P$, thus reducing the
equation to:

\[ E(T_{ij}) = t_a + t_l + t_f + (t_m + t_f)(1-P)/P \]

Smoler also assumed that for tanks, \( P(K|H) = 1.0 \) and \( t_f = 0 \), reducing the formula to:

\[ E(T_{ij}) = t_a + t_l + t_m(1-P)/P(h|m) \]

Utilizing these equations to calculate the \( A_{ij} \)'s, the attrition during each time step was computed using the Euler-Cauchy differencing equations to approximate Lanchester's force-on-force attrition differential equations.

C. UNIT LOCATION AND MOVEMENT

Red and blue unit locations were handled in two different ways in the original model. The blue locations were left as user inputs, while the red locations had been pre-determined by the model builder and could not be altered by the user. This allowed flexibility of defensive positions, but because of the method of determining movement routes for the attackers, this flexibility was limited.

The method utilized for route determination was, at best, unrealistic. For each original red location, a straight west to east route was calculated. Each route was divided into 40 meter intervals, since each red unit was assumed to move that distance during each 10 second time interval. This approximated an average rate of movement of 9 m.p.h. This method of route selection, although easy to implement, has several significant shortcomings. It does not allow the attacking units to utilize terrain...
features during movement. Also, it permits attackers to move over terrain that is, in real life, impassible.

D. BATTLE TERMINATION

In any combat model, adequate battle termination rules must be considered. Smoler utilized two criteria to terminate the battle. The first of these was annihilation of one of the two forces. This criteria is reasonable in an expected value model like this and was adequately handled. This is not the case for the second termination rule.

The second rule for termination is that the distance between red and blue forces becomes too small. The use of this criteria for a stopping rule is fine, provided it is implemented properly. In the original model, a geographic center was calculated for each force. It was these center of mass points that were compared to determine if the units were too close. This method can lead to problems, such as those that were encountered during some initial trial runs. In one battle, the red units, during their advance, had eliminated one entire blue unit and had suffered the loss of one of their own units. Since the red units are allowed to advance as much as 150 meters apart and blue units were set up so they were approximately 1300 meters apart, there was considerable distance between units on the battlefield. Since the two units had been destroyed, the center of mass computation allowed the red
units to move completely through the blue defense, while both sides still had forces remaining to fight with. This reflects a problem with the center of mass computation as a method for determining distances between units.

E. DETECTION AND FIRE ALLOCATION

Both detection and fire allocation processes are handled well in the original model and no changes have been made. A detailed description of these methods is contained in Smoler's Thesis (Smoler, 1979). However, a brief and general discussion of both processes will be included here.

The detection phenomena is modeled in two ways. First, a non-firing detection can occur as a result of an observer's random search within his designated section of responsibility. A 30° field of vision for an observer is assumed, and the probability that an observer is looking in the direction which enables him to detect a target is computed by integrating the Limicon Function over limits that are ± 15° from the primary direction the observer is looking. The Limicon Function, \( f(\theta) \), is the following probability density function:

\[
f(\theta) = A + B \cos(\theta)
\]

where

\[
D = \text{assigned sector width}/2
\]

\[
A = -B \cos(D)
\]

\[
B = 1 / 2(\sin(D) - D \cos(D))
\]

\( \theta = \text{primary direction observer is looking} \)
Also, A and B are chosen such that
\[ \int_{-D}^{D} f(\theta) d\theta = 1 \]

The second method of detection is a firing detection. This phenomena occurs when the following happens. If a firer's location is within \( \pm 15^\circ \) of an observer's primary direction of observation when he is firing, he is assumed to be detected and is added to the observer's target list. This models the detection of a firer by locating a firing signature of a weapon.

The fire allocation process is also modeled well by Smoler. Since each firing unit is not restricted to firing at one target, a fraction of each firing unit is allocated to each target on the firer's target list. This fraction is determined as a function of range to each target and of predetermined priorities of fire that are set so each unit's fire priority goes to those targets to his immediate front. Targets to the firer's flanks are then allocated a smaller percent of the available firepower. Again, the detection and fire allocation processes have been handled well and no changes to these routines will be made.
III. MODEL CHANGES

A. GENERAL

As the previous section outlines, there were several major problems discovered in Smoler's original model. In order to make the model more flexible and classroom useable, several major changes were found to be necessary. This chapter will provide a detailed description of these changes, which include conditionally-deterministic attrition, unit locations and movement, battle termination and movement of defenders.

B. CONDITIONALLY-DETERMINISTIC ATTRITION

In order to introduce a stochastic process into the attrition computation for this model, several options were available. These include the use of a Markov-process to determine casualties or the use of random attrition-rate coefficients. After considering both options, and in keeping with the transparent nature of this model, it was decided that stochastic attrition coefficients should be utilized, because the attrition-rate coefficient is a random quantity measuring a unit's fighting ability and can be realized before any given battle. The following procedure was used to develop a method for random attrition-rate coefficients.

The attrition-rate coefficient, $A_{ij}$, is a measure of the rate a firer of unit $i$ attrits a target of unit $j$. 
This also can be interpreted as a measure of the fighting ability of an element of type i. It is intuitively obvious that this is a variable quantity that can be affected by many different factors, including weather, espirit-de-corps, previous engagements and leadership, to mention only a few. In a homogeneous model, one can attempt to capture these random effects by developing a distribution of initial fighting unit capabilities, i.e., initial $A_{ij}$'s for each unit. Since no data is readily available that captures this phenomena, an attempt was made to fit a distribution to an intuitive feeling as to how this fighting ability varies from one unit to the next prior to a given engagement. It was the author's intuitive feeling that the $A_{ij}$'s should be distributed between .3 and .8 with the majority of the units being rated between .5 and .6.

Utilizing the above intuition, an attempt was made to "fit" a distribution to these assumed fighting levels. Graphically, the distribution would look like Figure 2.

![Figure 2. Distribution of Initial $A_{ij}$'s](image)
Initially, an attempt was made to fit a Beta distribution to this curve. The fit was fairly accurate in the middle range, but was not satisfactory in the tails. The same was true for a Normal approximation. It was finally determined that the best fit was a straight quadratic fit, where a Uniform(0,1) input variable could be used and the output would yield the desired random number. The quadratic fit that was used is:

\[ A_{ij}^0 = -2U^2 + 2U + .3 \]

Once a distribution for the attrition-rate coefficients had been derived, the implementation into the model was accomplished. Since it was assumed that the fighting ability of each unit is a random quantity prior to a given battle, it was only necessary to obtain a realization of the random variable for each unit prior to the initiation of the battle. This realization, \( A_{ij}^0 \), is determined by using a random Uniform(0,1) number and the above formula. A new attrition-rate coefficient for each unit is then computed during each time step using the equation:

\[
A_{ij} = \begin{cases} 
A_{ij}^0 (1-r/e)^2 & \text{for } 0 < r < e \\
0 & \text{for } r \leq e 
\end{cases}
\]

where

- \( r_e \) = maximum effective range of a firer's weapon
- \( r \) = current range between firer and target
- \( A_{ij}^0 \) = realization of unit's fighting ability
This equation is utilized because it varies with range, but it also is a function of $A_{ij}^0$, thus creating a different attrition-rate curve for each unit, depending on that unit's fighting ability prior to the battle. Graphically, as an example, the attrition-rate coefficient curve for an $A_{ij}^0$ of .6 and a maximum effective range, $r_e$, of 3000 meters would look as in Figure 3.

![Figure 3. Attrition-rate coefficient curve for $A_{ij}^0 = .6$ and $r_e = 3000$ m.](image)

Once the above method for determining attrition-rate coefficients had been selected, it was coded and included as a user option in the program. Once the code had been implemented, several runs were made utilizing different random number seeds to compare battle results using this method for attrition-rate coefficient determination. These runs were then compared with a run using the original deterministic method. The results are summarized below.

First, the model was run using the deterministic attrition method. For purposes of comparison, all locations,
movement rates, force levels and other input variables were held constant for all runs. Combatant attrition using the deterministic process as a function of time is shown in Figure 4.

![Figure 4. Attrition using deterministic attrition](image)

In this battle, termination occurred as a result of forces being too close together, with force levels at termination reflected in Figure 4. This was then compared to the runs that utilized the new attrition module. The results of four of these runs are in Figure 5. The $A_{ij}^0$'s are the realizations of the random variable denoting a unit's initial fighting capability prior to the battle. The three numbers under the red and blue headings are the realizations for each subunit in the battle on which casualty assessments are computed.

These results show, in fact, that a randomly selected $A_{ij}^0$ does have an effect on the final outcome of the battle. This module now provides the user the option of selecting a conditionally-deterministic process that still uses Lanchester's equations as the basis for the attrition computation.
<table>
<thead>
<tr>
<th>$A_{ij}$</th>
<th>Force Level</th>
<th>Blue</th>
<th>Red</th>
<th>Time</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Blue</td>
<td>Red</td>
<td></td>
<td></td>
</tr>
<tr>
<td>.33 .77</td>
<td>.46 .79 .49</td>
<td>1.7</td>
<td>1.5</td>
<td>780</td>
<td>Too close</td>
</tr>
<tr>
<td>.39 .67</td>
<td>.58 .63 .78</td>
<td>4.3</td>
<td>0</td>
<td>330</td>
<td>Blue wins</td>
</tr>
<tr>
<td>.78 .67</td>
<td>.80 .74 .68</td>
<td>3.2</td>
<td>0</td>
<td>320</td>
<td>Blue wins</td>
</tr>
<tr>
<td>.32 .43</td>
<td>.68 .51 .77</td>
<td>1.2</td>
<td>3.8</td>
<td>640</td>
<td>Too close</td>
</tr>
<tr>
<td>Deterministic</td>
<td>1.1</td>
<td>2.8</td>
<td>460</td>
<td>Too close</td>
<td></td>
</tr>
</tbody>
</table>

Figure 5. Comparison of Stochastic and Deterministic Battles

C. UNIT LOCATION AND MOVEMENT

As was pointed out in an earlier section, the unit locations and movement logic that Smoler used tended to be rigid and inflexible. The changes that have been made to rectify this and allow more flexibility for the user will be discussed here.

First, the user is now responsible for inputing all combatant locations, both attackers and defenders. The format for these inputs will be specified in Appendix A, the User's Manual. Additionally, the user is given the option of selecting routes for the red advance or allowing the model to calculate straight west to east routes as in the past. To add more flexibility, the user will also be required to input the rate of advance (vehicle speed) for the attacking forces. The inputed rate of advance is used in both methods of route calculation, thus controlling the red attack.
1. **Straight Line Routes**

The option to use straight line routes from west to east is handled in the same manner as before. The route is broken down into discrete distance intervals from the initial red location straight to the east. The only difference is that the length of each interval is determined by the vehicle speed that the user inputs. The route for each red unit consists of 125 equal length intervals.

2. **User Determined Routes**

The option to allow the user to select routes for the red advance has been added to enable the user to make use of available terrain and to add realism to the model. The method for calculating these routes is straightforward and is discussed next.

The user is required to input the original location of each red subunit and from one to ten nodes he wishes each attacking subunit to move through. This information, along with vehicle speed, is used to calculate route intervals that move the attacking unit through each of the designated nodes. A complete route would look like the one depicted in Figure 6. The method used to compute the routes is as follows.

The straight line ground distance between the first two adjacent nodes, DIST, is calculated as shown in Figure 7. The angle between the desired direction of movement and straight west to east movement, \( \alpha \), is then calculated. Utilizing these quantities and the distance desired to
move during each time step, DST, the distance to be moved in the x and y direction, XLI and YLI, is now computed as shown in Figure 7. These distances are then added to the coordinates of the previous interval endpoint, point C in the figure, to determine the coordinates of the next interval endpoint, point D. This same distance is again added to compute the coordinates of the next endpoint, point E. This process is continued until the distance from the last endpoint computed to the next node is less than DST. This process is repeated between the next two nodes until the entire route is completed. In order to insure all intervals are of equal length, the computation of the first interval between any two nodes must be considered separately, by taking into account the distance left over from the last computation between the previous two nodes. To do this, the first interval takes the remaining distance, e, and adds it to an interval length of DST-e for the first interval between any two nodes. This insures that each interval along the route is of length DST, which is the required length.

D. BATTLE TERMINATION

As previously stated, the battle is terminated either by annihilation of one of the two forces or by forces getting too close. The annihilation criteria has not been changed. However, to insure the distance criteria is effective, two changes have been made.

First, the distance between units is no longer calculated
NODE COORDINATES USE

A XIC(N-1),YIC(N-1) (N-1)st Interval Endpoint
B XLOC(I,J) User Inputed Node
C XIC(N),YIC(N) Nth Interval Endpoint
D XIC(N+1),YIC(N+1) (N+1)st Interval Endpoint
E XIC(N+2),YIC(N+2) (N+2)nd Interval Endpoint
F XLOC(I,J+1) User Inputed Node

\[ \text{DIST} = \sqrt{X^2 + Y^2} \]
\[ a = \tan^{-1}(Y/X), \text{ where } Y = |YL| \]
\[ e = \text{distance less than DST at end of calculation} \]
\[ \text{of intervals between adjacent nodes.} \]
\[ YLN = \text{DIST} \times \sin(a) \quad XLN = \text{DIST} \times \cos(a) \]

Figure 7. Route Computation
as a center of mass distance. Instead, a distance is computed between each attacking subunit on which casualties are being computed that is still alive and each defending subunit that is still in the battle. If any of these distances becomes too small, the battle is considered to have moved to a "close-in" combat (hand-to-hand) that this program does not currently model. For this reason, the battle is then terminated. However, to insure the attackers do not pass by the flanks of remaining defenders and remain outside termination distance, a check of x coordinates for each subunit is also made. If any attacking subunit's x coordinate places him past the location of the forward most blue defensive subunit still in the battle, the battle is also terminated.

The criteria for being too close has been left as a user determination. It is one of the inputs that is described in the User's Manual, Appendix A. This allows for flexibility of breakpoint distances which also lends itself to the study of optimum breaking distances for various weapon systems on the battlefield.

E. ALTERNATE DEFENSIVE POSITIONS

The last addition to the model, added to increase the flexibility and realism of the battle, is the option of having the defending units move to alternate positions if the attackers close to within a user specified range. The move is handled in a simple and transparent manner.

If the user selects this option, the defender will move
when the attacking forces close within the breakpoint distance. When this occurs, each defensive unit that is still alive will move to an alternate position the user has selected. The duration of the move is also a user input. He simply specifies the number of 10 second time steps he wants to allow the blue forces to move. At the completion of the move, during which the red forces continue to attack, the battle will continue until one of the termination criteria discussed above has been met.
IV. FUTURE ENHANCEMENT

Although the above changes have been implemented and a more useable and flexible model has been created, there are several areas that could still be considered in future work. These include the introduction of heterogeneous forces, explicit computation of ammunition expenditure and artillery (indirect fire). Possible methods for employing these ideas are discussed below.

A. HETEROGENEOUS FORCES

The current model involves combat between homogeneous forces only. In other words, each opposing force is comprised entirely of one weapon system type. Added flexibility could be attained by allowing heterogeneous force structures on both sides. This would enable the user to investigate the effect of different force mixes on battle outcome.

The introduction of different weapon systems within a single unit would require extensive changes to the attrition process. Although Lanchester equations for aimed-fire could still be utilized, casualty assessment against each weapon system type by each opposing weapon system type would have to be calculated. The total attrition of any particular unit would then be the sum of the damage assessed to each weapon system of that unit. Additionally, the force level for each weapon system would be required
as an input as well as a separate set of hit and kill
probabilities for each weapon system type against each
type of target.

Introduction of this feature would create a more comp-
licated model, adding realism but detracting from the current
simple and transparent form. Since the purpose of the
current effort has been to maintain this simplicity, this
option has not been included, but could be considered for
future work.

B. AMMUNITION EXPENDITURE

The combat process is a complicated and intricate
process. One of the hardest, and not many times attempted,
areas to model is the logistic area. However, in the
model currently being studied, the problem of ammunition
expenditure could be modeled.

Since the model is an expected value type model, it
would not be effective to model the amount of ammunition
expended by simply counting bullets as they are fired.
Actually, in this type of model it would not be possible
to count each round, because of the aggregated nature of
each unit. The only way to model ammunition usage is to
model it in the same way as attrition. This simply means
that each unit would have to be given a starting level of
ammunition on hand, a basic load, and each time the unit
fired, the expected amount of ammunition expended would
be subtracted from what is on hand. The amount expended
would necessarily be a function of the size of the unit.
firing, the number of targets fired at and the rate this unit is firing. This can be directly correlated to the attrition-rate coefficient for the firing unit. When ammunition on hand reaches the zero level, the unit would have to be removed from the battle permanently or for some specified time period, to simulate resupply.

C. ARTILLERY

In any armed conflict, there are more ways to inflict casualties than just direct fire. Many types of indirect fire are utilized on the battlefield, including artillery, close air support and naval gunfire. To include this type of play in this model, primarily artillery, would involve a major effort with large program additions.

To model artillery in an expected value model that already uses Lanchester equations for direct fire would suggest the use of Lanchester's area-fire equations. These equations are no more complicated to handle than the aimed-fire equations that are already in use. However, other model considerations would have to be investigated. These would include location of artillery units, whether to use forward observers for target location or only use pre-planned strikes or both and fire allocation procedures to be used. Also, a significant amount of data would need to be collected concerning weapon types to be used, effective ranges, killing radii and accuracy data.
To model artillery would be a significant addition to the model. However, this too would effect the simplicity and transparency the model currently possesses. This addition has also been left for future consideration.
APPENDIX A

User's Manual

I. INTRODUCTION

Smoler's model of land combat is a force-on-force combat model that utilizes Lanchester's aimed fire equations for casualty assessment. The battle is simulated on a 10 x 10 KM piece of terrain representing an area east of Fulda, West Germany. A portion of the terrain map is inclosed as Figure 1. It is a computer model that is coded in FORTRAN and is available for use in the W. R. Church Computer Center.

The purpose of this manual is to familiarize the user with the model and to provide the required Job Control Language (JCL) and user inputs to run the model. A sample listing of program output will also be provided to familiarize the user with the expected program output.
II. PROGRAM STRUCTURE

Smoler's enhanced model is a computerized model, coded in FORTRAN, containing a main program and nine subroutines. To assist the user in understanding how the model operates, a brief description of the function of each of these major parts of the program will be included.

A. MAIN PROGRAM

The main program has several important functions. All of the input and output functions, except the line-of-sight data, are contained in the main program. Additionally, the main program is used to structure all of the other functions during each 10 second time step. Attrition, detection, movement and fire allocation are also handled in the main program. The nine subroutines provide needed input numbers for the above calculations. The general flow of the program is depicted in the flow chart shown at Figure 2.

B. SUBROUTINES

There are nine subroutines included in the model. Each of these subroutines performs a distinct function, each of which will be discussed below.

1. Subroutine SETUP

This subroutine is used to read in the terrain data for the Fulda Map. This terrain data will be used when computing line-of-sight between targets and observers,
INITIALIZATION:
Read input data (terrain, initial force levels, initial positions, routes data, tactical parameters, etc.).
Initialize variables and array values.

Time = 0

Time = Time + 10 sec

Update locations of attacking units

Check line-of-sight between any two opponent units

NO

YES

LOS exists

fire conditions are satisfied

NO

Update the accumulated detection probabilities

Allocate fire

Compute attrition

Print results for the current time step

NO

End of battle

YES

Allocate targets

Print final results

END

FIGURE 2. FLOW CHART
as well as providing a grid system for unit locations and movement.

2. **Subroutine LOS**

   This module, developed by Professor Jim Hartman of the Naval Postgraduate School, computes a percent of a target visible to a particular observer, given the coordinates (location) of both. This visible fraction is used in the detection and attrition modules in the model.

3. **Subroutine KCVER**

   This subroutine is used by subroutine LCS in determining what portion of a particular target is covered by the terrain between the target and observer. This number is used in the detection and attrition modules.

4. **Subroutine ETK**

   This module computes the expected time for a given firer to kill a given target. The calculation is a function of range, time of flight for a round and hit and kill probabilities for the firing weapon system. It is a number that is used in the computation of the deterministic attrition coefficients, \( A_{ij} \).

5. **Subroutine STOCH**

   This is used when a user has selected a stochastic attrition option to compute the attrition coefficients during each time step. The calculation is a function of the original stochastically determined \( A_{ij} \) as well as a function of range.
7. **Subroutine LAIBDA**

This subroutine is used in conjunction with the LOS routine to compute the detection rate of a target by an observer given the percent of target visible to the observer.

8. **Subroutine ROUTE**

This is used to compute the route of each attacking red unit when the user has selected the option of inputing attacker routes. It calculates the coordinates of each interval endpoint along the route, making each interval length (distance moved during a 10 second time step) the same. The interval length is determined by the speed the user has selected and inputed for the current battle.

9. **Subroutine ELEV**

A subroutine that is used to calculate the terrain elevation for a given set of X, Y coordinates. This function is used in conjunction with the LOS subroutine in computing line-of-sight between observer and target.
III. AVAILABLE OPTIONS

The enhanced version of this model has been written to allow for maximum user flexibility while maintaining the simple and transparent nature of the model. To allow this flexibility, there have been several user options incorporated into the model. Each of these options, including user responsibilities, will be discussed here, with the required input data for each being outlined in the next section.

A. STOCHASTIC VS DETERMINISTIC ATTRITION:

The user is required to specify whether he wants to use stochastic or deterministic attrition calculations. Both methods utilize Lanchester's aimed-fire equations, the difference being the method of calculating the attrition-rate coefficients. The deterministic procedure uses the Bonder-Farrell method of calculating the $A_{ij}$, while the stochastic method selects an initial random $A_{ij}$ for each unit, and uses this as a function of range to calculate the $A_{ij}$ for each time step.

B. ATTACK ROUTES

The second major option available to the user is the method of route computation for the attacking forces. The user has to decide whether to allow the program to compute straight west to east routes or to input the routes.
he desires each attacking unit to follow. To select his own routes, a user must input the number of nodes he wishes to have on each route and the coordinates of each of these nodes. The program will then compute routes through each node. The nodes must be inputed in order from west to east and should not create an angle between the west to east axis and the route direction that exceeds $45^\circ$.

C. ALTERNATE DEFENSIVE POSITIONS

The third option the user must consider is the use of alternative defensive locations. This option permits the user, if he desires, the capability to move the defenders to alternate positions if the attackers close within some specified distance. This breakpoint distance is decided and inputed by the user and is also used as the distance for battle termination. The alternative to moving the defenders is to terminate the battle when the breakpoint distance is reached.

D. OTHER INPUTS

There are other inputs that are required by the program that the user must provide. These include force sizes, weapon characteristics, unit locations and hit and kill probabilities. The required formats for all inputs are outlined in the next section.
IV. INPUT DATA

All user input data is read in from cards at the start of the main program, MAIN. A brief description of each data element as well as the required format for inputting this data is discussed next. A sample data deck has also been included as Figure 3, with the referenced data element number appended on the right. This sample data deck reflects the use of all data elements. However, not all the elements described below are required, and those that are needed only if a particular option is selected are noted with an asterisk next to the data element number.

<table>
<thead>
<tr>
<th>Data Element</th>
<th>Description and Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Selection of attrition option and random number seed. A 1 is entered for deterministic or a 0 for stochastic attrition. This number is followed by a five digit random number seed. Format for this card is I1,IX,I5.</td>
</tr>
<tr>
<td>2.</td>
<td>Number of blue units (NBU) and number of red units (NRU). Both of these elements are two digit integers. Card format is I2,iX,I2.</td>
</tr>
<tr>
<td>3.</td>
<td>Effective weapon ranges listed in the following order. Red minimum, red</td>
</tr>
</tbody>
</table>
maximum, blue minimum and blue maximum.
Card format is 4(F6.1,1X).

4. Red force levels for each unit.
Card format is F3.1, 1X. The card will contain the number of entries that equals the number of red units (NRU) in data element 2.

5. Type of route computation desired followed by vehicle speed. Use a 1 for user determined routes or a 0 for program determined routes. A one digit entry will designate desired speed as follows:
   1 for 9 mph
   2 for 12 mph
   3 for 15 mph
   4 for 18 mph
Card format for these elements is I2, 1X, I2.

6. X, Y coordinates for each initial red location. One card is needed for each red unit with format F6.1, 1X, F6.1.

7.* If the user has selected to enter his own routes in data element 5, the following is required for each route.
   a. The number of nodes in the route (from 01 to 10). This card's format is I2.
b. X, Y coordinates of each node along the route in order from west to east. One card for each node in format F6.1, 1X, F6.1.

3. X, Y coordinates for blue location, force level for each unit, primary search direction for that unit and desired search width. Force level must be between 1.0 and 3.0, search direction between 135° and 225°, and search width from 30° to 120°. One card for each blue unit with format F6.1, 1X, F6.1, 1X, F3.1, 1X, I3, 1X, I3.

9. Specify if blue is to move to alternate defensive positions, breakpoint distance, and number of 10 second time intervals allowed for the move. A 1 for no move or 0 for option to move. Format for this card is I1, I7, F6.1, I7, I2.

10.* Alternate blue position X and Y coordinates if move is desired (data element 9). One card for each location with format F6.1, 1X, F6.1.

11. Red weapon system hit and kill probabilities. The probabilities are entered for 6 range bands, with one card for each range band. Four probabilities for each range are probability of a 1st round hit,
The 6 range bands are from 0 to 500, 500 to 1000, 1000 to 1500, 1500 to 2000, 2000 to 2500 and 2500 to 3000. Each of these six cards has the format 4(F4.2, 1X).

12. Same as item 11, except the probabilities for the blue weapon systems are entered. Six cards with format 4(F4.2, 1X).

In the formats listed above, a 1X is a space, an IX is an integer of x digits, and a Fx.y is a real number of length x with y digits to the right of the decimal. For example, an I2 could be a 25, an F6.1 could be 3487.4. A format containing these specifications could be I2, 1X, F6.1. The data card would then look like 25 3487.4, with the 25 beginning in card column one.
<table>
<thead>
<tr>
<th>card</th>
<th>column</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>data number</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>28943</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>03</td>
<td>03</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>0000.0</td>
<td>2500.0</td>
<td>0500.0</td>
<td>4000.0</td>
<td></td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>3.0</td>
<td>3.0</td>
<td>3.0</td>
<td></td>
<td></td>
<td></td>
<td>4</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>5</td>
</tr>
<tr>
<td>2500.0</td>
<td>7300.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>6</td>
</tr>
<tr>
<td>2400.0</td>
<td>5900.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2450.0</td>
<td>5300.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>02</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4900.0</td>
<td>7200.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7100.0</td>
<td>6700.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>02</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5350.0</td>
<td>6000.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7000.0</td>
<td>6300.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>01</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6900.0</td>
<td>5500.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5700.0</td>
<td>7800.0</td>
<td>3.0</td>
<td>190</td>
<td>120</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5800.0</td>
<td>6500.0</td>
<td>3.0</td>
<td>180</td>
<td>120</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5900.0</td>
<td>5900.0</td>
<td>3.0</td>
<td>180</td>
<td>120</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0500.0</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7000.0</td>
<td>7200.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7100.0</td>
<td>6500.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7100.0</td>
<td>5800.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.85</td>
<td>0.85</td>
<td>0.75</td>
<td>0.70</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.80</td>
<td>0.80</td>
<td>0.75</td>
<td>0.70</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.75</td>
<td>0.75</td>
<td>0.70</td>
<td>0.65</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.60</td>
<td>0.65</td>
<td>0.60</td>
<td>0.55</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.45</td>
<td>0.50</td>
<td>0.50</td>
<td>0.35</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.20</td>
<td>0.20</td>
<td>0.20</td>
<td>0.20</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 3. Sample Data Deck
<table>
<thead>
<tr>
<th>card</th>
<th>column</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1234567890123456789012345678901234567890</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>0.60</td>
<td>0.70</td>
<td>0.65</td>
<td>0.85</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.85</td>
<td>0.90</td>
<td>0.85</td>
<td>0.90</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.80</td>
<td>0.85</td>
<td>0.85</td>
<td>0.80</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.75</td>
<td>0.80</td>
<td>0.75</td>
<td>0.70</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.60</td>
<td>0.70</td>
<td>0.65</td>
<td>0.65</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.40</td>
<td>0.45</td>
<td>0.40</td>
<td>0.50</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Figure 3.** Sample Data Deck (cont)
V. REQUIRED JCL

The model and line-of-sight data are residing on permanent disk in the W. R. Church Computer Center. In order to exercise the model, the required job control language (JCL) is illustrated in Figure 4. Due to the CPU time required for execution, a time parameter of \( \text{TImE} = 2 \) must be used on the job card.

```
// JOB CARD, TIME = 2
// EXEC FORTCLG
// FORT.SYSIN DD UNIT = 2314, VOL = SER = PAT002, DISP = SHR,
// DSN = S1360.SMOLET.PGM, DCB = (RECFM = FB, LRECL = 80, BLKSIZE = 3200)
// G0.FT08FO01 DD UNIT = 2314, VOL = SER = PAT002, DISP = SHR,
// DSN = S1360.LOSDATA, DCB = (RECFM = FB, LRECL = 80, BLKSIZE = 3200)
//
// GO.SYSIN DD *

USER DATA DECK

/*

Figure 4. Required JCL
```
VI. EXPECTED OUTPUT

Once the model has been executed, the user can expect the following output:

1 - Program listing of the model.

2 - A summary of the initial battle conditions, including starting locations and options selected.

3 - A summary of battle conditions after each 10 second time step, including unit locations, force levels, unit status, percentage lost and targets on each unit's target list. The status will show one of the following:

   0 - Unit alive, not firing
   1 - Unit alive and firing
   2 - Unit killed
   3 - Unit moving

4 - A statement of the reason for battle termination.

A sample of the initial battle summary is at Figure 5 and the time step summary is at Figure 6.
**INITIAL BATTLE INFORMATION**

<table>
<thead>
<tr>
<th>UNIT</th>
<th>X</th>
<th>Y</th>
<th>FORCE LEVEL</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1900.0</td>
<td>7800.0</td>
<td>6.0</td>
</tr>
<tr>
<td>2</td>
<td>1700.0</td>
<td>6500.0</td>
<td>6.0</td>
</tr>
<tr>
<td>3</td>
<td>1700.0</td>
<td>4900.0</td>
<td>6.0</td>
</tr>
<tr>
<td>4</td>
<td>5800.0</td>
<td>7700.0</td>
<td>3.0</td>
</tr>
<tr>
<td>5</td>
<td>7400.0</td>
<td>6300.0</td>
<td>3.0</td>
</tr>
<tr>
<td>6</td>
<td>6200.0</td>
<td>5150.0</td>
<td>3.0</td>
</tr>
</tbody>
</table>

ATTRITION IS STOCHASTIC

ROUTES DETERMINED BY USER

RED VEHICLE SPEED IS 15.0

BREAKPOINT DISTANCE IS 1500.0

BLUE WILL MOVE TO ALTERNATE POSITIONS

ALTERNATE POSITIONS ARE:

<table>
<thead>
<tr>
<th>UNIT</th>
<th>X</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>9300.0</td>
<td>7300.0</td>
</tr>
<tr>
<td>5</td>
<td>9800.0</td>
<td>6200.0</td>
</tr>
<tr>
<td>6</td>
<td>9200.0</td>
<td>5800.0</td>
</tr>
</tbody>
</table>

**RED KILL PROBABILITIES**

<table>
<thead>
<tr>
<th>RANGE</th>
<th>P</th>
<th>PHH</th>
<th>PHM</th>
<th>PKH</th>
</tr>
</thead>
<tbody>
<tr>
<td>500</td>
<td>0.95</td>
<td>0.95</td>
<td>0.95</td>
<td>1.00</td>
</tr>
<tr>
<td>1000</td>
<td>0.80</td>
<td>0.83</td>
<td>0.81</td>
<td>0.95</td>
</tr>
<tr>
<td>1500</td>
<td>0.65</td>
<td>0.70</td>
<td>0.70</td>
<td>0.90</td>
</tr>
<tr>
<td>2000</td>
<td>0.40</td>
<td>0.50</td>
<td>0.55</td>
<td>0.75</td>
</tr>
<tr>
<td>2500</td>
<td>0.30</td>
<td>0.40</td>
<td>0.40</td>
<td>0.75</td>
</tr>
<tr>
<td>3000</td>
<td>0.20</td>
<td>0.30</td>
<td>0.25</td>
<td>0.60</td>
</tr>
</tbody>
</table>

**BLUE KILL PROBABILITIES**

<table>
<thead>
<tr>
<th>RANGE</th>
<th>P</th>
<th>PHH</th>
<th>PHM</th>
<th>PKH</th>
</tr>
</thead>
<tbody>
<tr>
<td>500</td>
<td>0.80</td>
<td>0.80</td>
<td>0.80</td>
<td>1.00</td>
</tr>
<tr>
<td>1000</td>
<td>0.80</td>
<td>0.80</td>
<td>0.80</td>
<td>0.95</td>
</tr>
<tr>
<td>1500</td>
<td>0.70</td>
<td>0.70</td>
<td>0.70</td>
<td>0.90</td>
</tr>
<tr>
<td>2000</td>
<td>0.65</td>
<td>0.65</td>
<td>0.65</td>
<td>0.85</td>
</tr>
<tr>
<td>2500</td>
<td>0.45</td>
<td>0.55</td>
<td>0.45</td>
<td>0.70</td>
</tr>
<tr>
<td>3000</td>
<td>0.45</td>
<td>0.55</td>
<td>0.45</td>
<td>0.60</td>
</tr>
</tbody>
</table>

Figure 5. Initial Battle Summary
TIME = 480 SECONDS

<table>
<thead>
<tr>
<th>UNIT</th>
<th>X</th>
<th>Y</th>
<th>FORCE LEVEL</th>
<th>STATUS</th>
<th>LOST-PCT</th>
<th>TARGETS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4933.2</td>
<td>7193.2</td>
<td>5.0</td>
<td>0</td>
<td>0.160</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>4832.3</td>
<td>5799.9</td>
<td>4.7</td>
<td>0</td>
<td>0.214</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>4902.6</td>
<td>4579.8</td>
<td>6.0</td>
<td>0</td>
<td>0.0</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>5800.0</td>
<td>7700.0</td>
<td>0.0</td>
<td>2</td>
<td>1.000</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>7400.0</td>
<td>6300.0</td>
<td>3.0</td>
<td>1</td>
<td>0.0</td>
<td>2</td>
</tr>
<tr>
<td>6</td>
<td>6200.0</td>
<td>5150.0</td>
<td>3.0</td>
<td>3</td>
<td>0.0</td>
<td></td>
</tr>
</tbody>
</table>

Figure 6. Time Step Summary
APPENDIX B

Program Listing

```
1 COMMON /GAP1/ IPADIR (0), ISECCHO (0), NVTDIR (0), X (0), T (0), SPD (0)
2 COMMON /GAP2/ TA (2), T (2), T (2), TM (2), TF1 (2), TF2 (2), TFS (2),
3 IP (2, 8), PNM (2, 8), PNM (2, 8), PNM (2, 8), TF (2)
4 COMMON /GAP3/ NBUL, NRU, FL (6), FO (6), NOI (3), XIC (3, 200), YIC (3, 200),
5 11D1R (3, 200), AVSP, ISPD
6 1, IUSTAT (8), IT (6), LOST (6, 6), VISFRA, VISFRA, SIZE, TK,
7 1SIZE, NT (6), NF (6), SRF, DSNAX,
8 1HLSCH (6, 6), VISFRA (6, 6), RINTK, XMKT, RINTW, DP, TOWFR, TINFR.
9 IPTT (3, 3), AF, PDA (6, 6), PDA (6, 6), LDA (6, 6), NR (6, 6), GFL (6), GOL (6)
10 COMMON /GAP4/ TPOL (6), ODLQ (6, 6), O (6, 6)
11 COMMON /GAP5/ LOT (6, 6), ROT (6, 6)
12 COMMON /HILLS/ XC (100), YC (100), PEAK (100), SX (100), SY (100), HMS (100)
13 COMMON /HILLS/ SCALE (100), TWORMO (100), TMOSSCL (100), BASE
14 COMMON /HILLS/ NHILLS
15 COMMON /COVER/ CXC (150), CYC (150), CPEAK (150), CPXX (150), CPYT (150)
16 COMMON /COVER/ CPXY (150), NCVELS
17 COMMON /COUNTR/ KH, KHW, KN, KAS, KELV, KINT
18 COMMON /GRID/ LST (10, 10), MLSH (10, 10), LISTH (150), KCREP (100), KTREP
19 COMMON /GRID/ LSTC (10, 10), NC (10, 10), LSTC (150), KCREP (150)
20 COMMON /GAP6/ ALPHA (6)
21 COMMON /GAP7/ XA (6), YA (6), INFVE (6)
22 C
23 C INITIALIZATION.
24 C
25 CALL OYFLOW
26 BL=0.0
27 AL=0.0
28 NP=0
29 PRI=22.0/7.0
30 ZL=0.00
31 C
32 C READ TERRAIN DATA FOR LINE OF SIGHT
33 C CHECK FOR STOCHASTIC OR DETERMINISTIC ATTITRITION
34 C
35 READ (15, 130) ITAIT, I3
36 130 FORMAT (11, 1X, 15)
37 DO 132 I=1,6
38 CALL RANDOM (15, TRAN, 1)
39 ALPHA (I) = (-2. * TRAN + 2) * (12. * TRAN + 3)
40 132 CONTINUE
41 C
42 C READ IN NUMBER OF BLUE AND RED UNITS
43 C
44 READ (15, 200) NBUL, NRU
45 200 FORMAT (12, 1X, 12)
47 C
48 C INITIALIZE WEAPON SIZES
49 C
50 SIZETK=2.5
```
SIZEW=2.5

C READ IN EFFECTIVE WEAPON RANGES
C
C READ(S,102) AMINTK,AMXTK,AMINTW,AMXTW
102 FORMAT(F8.1,1X,F6.1,1X,F6.1,1X,F6.1,1X)

C INITIALIZE PM, RF, TOWFR, TNKFR AND NOD
C
PM=.352
RF=.5
TOWFR=.03
TNKFR=.1
NOD=2

DO 101 I=1, NRU
NOI(I)=125
101 CONTINUE

K=NRU+1
L=NRU+NU
DO 111 I=1, L
II(I)=0
111 CONTINUE

C READ IN FORCE LEVELS OF EACH RED UNIT
C
C READ(S,103) (FL(I), I=1, NRU)
103 FORMAT(F3.1,1X)

C CHECK FOR TYPE OF ROUTE DETERMINATION
C
C READ(S,106) IATE, ISP0
106 FORMAT(I1,1X,1I1)

IF(ISPO.EQ.1) AVSP=9.0
IF(ISPO.EQ.2) AVSP=12.0
IF(ISPO.EQ.3) AVSP=15.0
IF(ISPO.EQ.4) AVSP=18.0
IF(ISPO.EQ.5) AVSP=20.0

C READ IN INITIAL RED LOCATIONS
C
DO 6 I=1, NRU
6 READ(S,107) XIC(I, 1), YIC(I, 1)
107 FORMAT(F8.1,1X,F6.1)
B CONTINUE
IF(IATE.EQ.1) GO TO 108
DO 2 I=1, NRU
DO 2 J=2, 125
101  YIC(I,J)=YIC(I,J-1)+DST=(J-1)
102  XIC(I,J)=XIC(I,J-1)+DST=(J-1)
103  IDIR(I,J)=0
104  2 CONTINUE
105  GO TO 108
106  CALL ROUTE
107  109 SUMBO=0.0
108  DO 3 I=1,NAI
109  F0(I)=FL(I)
110  SUMBO=SUMBO+F0(I)
111  X(I)=XIC(I,1)
112  Y(I)=YIC(I,1)
113  HYTDIR(I)=IDIR(I,1)
114  SPD(I)=AVSP
115  JUSTR(I)=0
116  IpadI(R)=IDIR(I,1)
117  JsecWD(I)=120
118  NF(I)=1
119  NI(I)=1
120  3 CONTINUE
121  C
122  C READ IN INITIAL BLUE LOCATIONS
123  C
124  SUMBO=0.0
125  DO 4 I=K,L
126  READ(5,104) X(I),Y(I),FL(I),IpadI(R),JsecWD(I)
127  104 FORMAT(F6.1,IX,F6.1,IX,F6.1,IX,13.1X,13)
128  FO(I)=FL(I)
129  SUMBO=SUMBO+F0(I)
130  HYTDIR(I)=0
131  SPD(I)=0.0
132  JUSTR(I)=0
133  UMOVE(I)=0
134  4 CONTINUE
135  C
136  C CHECK FOR ALTERNATE BLUE POSITIONS AND READ IN IF WANTED
137  C
138  C READ(5,400) IALT,BREAK,ITEM
139  400 FORMAT(11,1X,F6.1,IX,12)
140  IF (IALT.EQ.1) GO TO 401
141  DO 402 I=K,L
142  READ(5,107) XA(I),YA(I)
143  402 CONTINUE
144  401 DELT=10.
145  TA(I)=20.
146  Ti(I)=6.
147  TH(I)=6.
148  TM(I)=10.
149  TF1(I)=1.
150  TF2(I)=1.
TF3(1) = 1.0
TF3(2) = 2.0
T1(2) = 0.0
TH(2) = 0.0
THM(2) = 0.0
TF1(2) = 10.0
TF2(2) = 12.0
TF3(2) = 15.0

C READ IN HIT AND KILL PROBABILITIES
DO 5 J=1,2
DO 514 J=1,6
READ (5,515) P(I,J), PHM(I,J), PHM(I,J), PHM(I,J)
515 FORMAT (4(F4.2,1X))
514 CONTINUE
5 CONTINUE
PTT(1,1) = 1.0
PTT(1,2) = 0.0
PTT(2,2) = 0.2
PTT(3,3) = 0.05
DO 31 I=1,NAU
DO 31 J=K,L
NLSCI(I,J) = 0
QL(I,J) = 1.0
Q(I,J) = 1.0
VISFR(I,J) = 0.0
VISFR(J,I) = 0.0
31 CONTINUE
IC=1
C PRINT INITIAL BATTLE INFORMATION
WRITE (6,599)
599 FORMAT ('1',1X,'INITIAL BATTLE INFORMATION')
WRITE (6,600)
600 FORMAT ('/X,'UNIT',7X,'X',8X,'Y',4X,'FORCE LEVEL')
DO 601 I=1,L
WRITE (6,602) I,X(I),Y(I),FL(I)
602 FORMAT (I1,13.3X,F7.1,2X,F7.1,7X,F3.1)
601 CONTINUE
IF (ITRIT.EQ.1) GO TO 603
WRITE (6,604)
604 FORMAT (/X,'ATTRITION IS STOCHASTIC')
GO TO 605
603 WRITE (6,606)
606 FORMAT (/X,'ATTRITION IS DETERMINISTIC')
201 605 IF (IALE.EQ.0) GO TO 607
202 WRITE (6,608)
203 608 FORMAT (1X,'ROUTES DETERMINED BY USER'/)
204 607 WRITE (6,609) AVSP
205 609 FORMAT (1X,'RED VEHICLE SPEED IS ',F4.1/
206 WRITE (6,610) BREAK
207 610 FORMAT (1X,'BREAKPOINT DISTANCE IS ',F6.1/
208 IF (IAL.EQ.0) GO TO 615
209 WRITE (6,620)
210 620 FORMAT (1X,'BLUE WILL NOT MOVE TO ALTERNATE POSITIONS'/)
211 GO TO 625
212 615 WRITE (6,630)
213 630 FORMAT (1X,'BLUE WILL MOVE TO ALTERNATE POSITIONS'/,
214 'ALTERNATE POSITIONS ARE: ','UNIT',S,X,'X',S,X,'T'
215 DO 635 I=K,L
216 WRITE (6,640) I,KA(I),TA(I)
217 640 FORMAT (1X,K13.5X,F7.2)
218 635 CONTINUE
219 625 IRAN=500
220 WRITE (6,645)
221 645 FORMAT (/4X,'RED KILL PROBABILITIES'/,
222 'RANGE',4X,'P',
223 DO 650 I=1,6
224 WRITE (6,655) IRAN,P(I),PHM(I),PKH(I)
225 655 FORMAT (2X,I4.4,2X,F4.2)
226 IRAN=IRAN+500
227 650 CONTINUE
228 IRAN=500
229 WRITE (6,660)
230 660 FORMAT (/4X,'BLUE KILL PROBABILITIES'/,
231 'RANGE',4X,'P',
232 DO 665 I=1,6
233 WRITE (6,675) IRAN,P(I),PHM(I),PKH(I)
234 IRAN=IRAN+500
235 665 CONTINUE
236 WRITE (6,670)
237 670 FORMAT ('1',10X,'BATTLE BEGINS'/)
238 C C UPDATE LOCATION OF RED UNITS.
239 C
240 DISMAX=5000.0
241 67 DD 9 I=1,NAU
242 IF (IUSTAT(I).EQ.2) GOTO 9
243 IF (IUSTAT(I).EQ.0) GOTO 76
244 NF(I)=NF(I)+1
245 IF (NF(I).LT.NOD) GOTO 9
246 NF(I)=1
247 76 DD 11 J = 1, NAU
248 IF (J .EQ. 1) GO TO 11
249 IF (IUSTAT(I) .EQ. 2) GO TO 11
DIST = X(I) - X(J)
IF (DIST .GT. DISMAX) GO TO 9
CONTINUE
II(I) = II(I) + 1
K7=II(I)
X(I)=XIC(1,K7)
Y(I)=YIC(1,K7)
MVTOIR(I)=IDIR(1,K7)
IPAOIR(I)=IDIR(1,K7)
CONTINUE
9 CONTINUE

251 C LINE--OF--SIGHT CHECK BETWEEN UNITS AND TARGETS SELECTION
252 C DO 17 J=K,L
253 NT(J)=0
254 CONTINUE
255 17 CONTINUE
256 DO 12 I=1,NRU
257 NT(I)=0
258 IF(IUSTAT(I).EQ.2) GOTO 12
259 J=K,L
260 IF(IUSTAT(J).EQ.2.OR.IUSTAT(J).EQ.3) GO TO 16
261 XX=X(I)
262 YY=Y(I)
263 CALL ELEV(XX,YY,TMAC)
264 XX2=X(J)
265 YY2=Y(J)
266 CALL ELEV(XX2,YY2,TMACJ)
267 LATOB=1
268 LBTOA=1
269 CALL LOS(XX1,YY1,TMACI,0.0,SIZETK,XX2,YY2,TMACJ,0.0,SIZETW)
270 ILATOB,LBTOA,VISFRA,VISFRAI
271 VISFR(1,J)=VISFRA
272 VISFR(1,J)=VISFRA
273 IF(VISFRA.GT.ZL) GOTO 18
274 LOST(I,J)=0
275 LOST(I,J)=0
276 NLOSC(I,J)=NLOSC(I,J)+1
277 NLOSC(I,J)=NLOSC(I,J)
278 GOTO 16
279 16 LOST(I,J)=J
280 LOST(I,J)=J
281 NLOSC(I,J)=0
282 NLOSC(J,J)=0
283 RANGE=SQR((X(I)-X(J))^2+(Y(I)-Y(J))^2)
284 IF(RANGE.LT.RMINTK.OR.RANGE.GT.RMXTK) GOTO 20
285 IF(II(J).EQ.1,D) GOTO 20
286 IUSTAT(1)=1
287 NT(I)=NT(I)+1
288 M=NT(I)
289 CONTINUE
301  ROT(I,M) = RANGE
302  IF (I .EQ. 1) GOTO 20
303  CALL SORT(I,M)
304  IF (RANGE .LT. AMINTM .OR. RANGE .GT. AMXTM) GOTO 16
305  IF (G(J,1) .EQ. 1.0) GOTO 16
306  IUSTAT(J) = 1
307  NT(J) = NT(J) + 1
308  M = NT(J)
309  ROT(J,M) = RANGE
310  IF (I .EQ. 1) GOTO 16
311  CALL SORT(J,M)
312  16 CONTINUE
313  12 CONTINUE
314  DO 25 I = 1, M
315  IF (IUSTAT(I) .EQ. 2) GOTO 25
316  IF (NT(I) .NE. 0) GOTO 25
317  IUSTAT(I) = 0
318  NF(I) = 0
319  25 CONTINUE
320  DO 79 J = K, L
321  IF (IUSTAT(J) .EQ. 2 .OR. IUSTAT(J) .EQ. 3) GOTO 79
322  IF (NT(J) .EQ. 0) IUSTAT(J) = 0
323  79 CONTINUE
324  C UPDATE OF THE ACCUMULATED DETECTION PROBABILITIES.
325  C
326  19 C
327  19 C
328  IAA = 1
329  IBB = MRAU
330  ICC = K
331  IOD = L
332  FR = TONFR
333  OP = PH
334  DO 14 I = IAA, IBB
335  IF (IUSTAT(I) .EQ. 2 .OR. IUSTAT(I) .EQ. 3) GOTO 14
336  DO 19 J = ICC, IOD
337  PROP = 0.0
338  IF (IUSTAT(J) .EQ. 2 .OR. IUSTAT(J) .EQ. 3) GOTO 19
339  NLOO(I,J) = 0
340  IF (LOST(I,J) .EQ. 0) GOTO 15
341  IF (NT(I) .GT. 0) GOTO 22
342  PCTVIS = VISFR(I,J)
343  CALL LAMDA(I,J,PCTVIS,DETRET,PSUBK)
344  QY = EXP(-FL(I) .EQ. DETRET .EQ. OP .EQ. DELT .EQ. FL(J))
345  IF (NT(J) .GT. 0) GOTO 23
346  Q(I,J) = Q(I,J)
347  23 GOTO 19
348  QP = (1.0 - PSUBK) * (FR .EQ. DELT .EQ. FL(J))
349  Q(I,J) = Q(I,J) * (QV .EQ. OP .EQ. QP)
350  GOTO 19

55
351  22 NS=NT(I)
352      DO 24 I=1,NS
353  23      K1=LOT(I,II)
354      ANG1=ATAN2(Y(K1)-Y(I),X(K1)-X(I))
355      ANG2=ATAN2(Y(J)-Y(I),X(J)-X(I))
356      IF ((ANG1+ANG2).GE.0.0) GOTO 77
357      IF (ANG2.LT.0.0) GOTO 32
358      ANG=2*PAI+ANG1+ANG2
359      GOTO 35
360  32      ANG=2*PAI+ANG2-ANG1
361      IF (ANG.GT.PAI) ANG=2*PAI-ANG
362      GOTO 33
363      77      ANG=ABS(ANG2-ANG1)
364      33      AA=15.0+PAI/180.0
365      IF (ANG.GT.RA) GOTO 24
366      PROP=PROP+PTT(I1,N5)
367      24      CONTINUE
368      IF (PROP.EQ.0.0) GOTO 34
369      IF (NT(J).GT.0) GOTO 36
370      CALL LAMDA(I,J,PCTVIS,DETRAT,PSUBK)
371      DETRAT=DETRAT+RF
372      QV=EXP(-1*FL(I))*PROP*DETRAT*DELT+FL(J))
373      O(I,J)=Q(I,J)=QV
374      GOTO 19
375      36      Q(I,J)=0.0
376      GOTO 19
377      34      IF (IAA.EQ.1) GOTO 19
378      O(I,J)=1.0
379      GOTO 19
380      15      IF (NLOSC(I,J).LE.3) GOTO 19
381      O(I,J)=1.0
382      19      CONTINUE
383      14      CONTINUE
384      IF (IAA.EQ.K) GOTO 38
385      FA=TNKFA
386      IAA=K
387      1BB=L
388      ICC=I
389      IDD=NAV
390      OP=1.0
391      GOTO 37
392      C
393      C   FIRE ALLOCATION.
394      C
395      38      DO 28 I=1,L
396      20      NA(I)=0
397      DO 26 I=1,L
398      IF (IUSTAT(I).EQ.2.OR.IUSTAT(I).EQ.3) GO TO 26
399      IF (NT(I).EQ.0) GOTO 26
400      26      DO 27 J=1,9

56
401 APOA(I,J) = 0.0
402 27 CONTINUE
403 IF NT(I).EQ.1 GOTO 78
404 IF NT(I).EQ.2 GOTO 29
405 NOT=3
406 MM1=LOT(I,1)
407 MM2=LOT(I,2)
408 MM3=LOT(I,3)
409 PROB=(1.0-Q(I,MM1))*(Q(I,MM2))*(Q(I,MM3))
410 APOA(I,1)=APOA(I,1)*PTT(I,1)*PROB
411 PROB=(1,MM1)*(1.0-Q(1,MM2))*(Q(1,MM3))
412 APOA(I,2)=APOA(I,2)*PTT(I,1)*PROB
413 PROB=(1,MM1)*(Q(1,MM2))*(1.0-Q(1,MM3))
414 APOA(I,3)=APOA(I,3)*PTT(I,1)*PROB
415 PROB=(1.0-Q(I,MM1))*(1.0-Q(I,MM2))*(Q(I,MM3))
416 APOA(I,1)=APOA(I,1)*PTT(I,2)*PROB
417 APOA(I,2)=APOA(I,2)*PTT(I,2)*PROB
418 PROB=(1.0-Q(I,MM1))*(Q(I,MM2))*(1.0-Q(I,MM3))
419 APOA(I,1)=APOA(I,1)*PTT(I,2)*PROB
420 APOA(I,3)=APOA(I,3)*PTT(I,2)*PROB
421 PROB=(1,MM1)*(1.0-Q(1,MM2))*(1.0-Q(1,MM3))
422 APOA(I,2)=APOA(I,2)*PTT(I,2)*PROB
423 APOA(I,3)=APOA(I,3)*PTT(I,2)*PROB
424 PROB=(1.0-Q(I,MM1))*(1.0-Q(I,MM2))*(1.0-Q(I,MM3))
425 APOA(I,1)=APOA(I,1)*PTT(3,1)*PROB
426 APOA(I,2)=APOA(I,2)*PTT(3,2)*PROB
427 APOA(I,3)=APOA(I,3)*PTT(3,3)*PROB
428 30 DO 44 J=1,NOT
429 KK=LOT(I,J)
430 NA(KK)=NA(KK)+1
431 IN=NA(KK)
432 LOA(KK,IN)=1
433 POA(KK,IN)=APOA(I,J)
434 44 CONTINUE
435 GOTO 26
436 29 NOT=2
437 MM1=LOT(I,1)
438 MM2=LOT(I,2)
439 PROB=(1.0-Q(I,MM1))*(Q(I,MM2))
440 APOA(I,1)=APOA(I,1)*PTT(I,1)*PROB
441 PROB=(1,MM1)*(1.0-Q(1,MM2))
442 APOA(I,2)=APOA(I,2)*PTT(I,1)*PROB
443 PROB=(1.0-Q(I,MM1))*(1.0-Q(I,MM2))
444 APOA(I,1)=APOA(I,1)*PTT(I,2)*PROB
445 APOA(I,2)=APOA(I,2)*PTT(I,2)*PROB
446 GOTO 30
447 78 NOT=1
448 MM1=LOT(I,1)
449 PROB=1.0-Q(I,MM1)
450 APOA(I,1)=APOA(I,1)*PTT(I,1)*PROB
GOTO 30
26 CONTINUE
C
452 ATTRITION.
C
SUM=0.0
SUMB=0.0
DO 40 I=1,L
453 IF (IUSTAT(I).EQ.2.AND.IUSTAT(I).EQ.3) GO TO 40
454 M6=WAR(I)
455 SUM=0.0
456 IF (M6.EQ.0) GOTO 47
457 DO 41 J=1,M6
458 M7=LOA(I,J)
459 IF (M7.LT.K) GOTO 42
460 ITYPE=2
461 GOTO 43
462 ITYPE=1
463 RANGE=SQR((X(I)-X(M7))^2+Y(I)-Y(M7)^2)
464 IF (ITRIT.EQ.1) GO TO 131
465 CALL STOCH(ITYPE,RANGE,AJI)
466 GO TO 5000
467 CALL ETK(ITYPE,RANGE,T)
468 AJI=1.0/T
469 SUM=SUM+AJI*FL(M7)*POA(I,J)*DELT
470 CONTINUE
FL(I)=FL(I)
471 IF (FL(I).GT.ZL) GOTO 46
472 FL(I)=0.0
473 IUSTAT(I)=2
474 IF (I.LT.K) GOTO 60
475 SUMB=SUMB+FL(I)
476 TPOL(I)=(FO(I)-FL(I))/FO(I)
477 GO TO 40
478 SUM=SUM+FL(I)
479 TPOL(I)=(FO(I)-FL(I))/FO(I)
480 CONTINUE
C
PRINT AND CHECK FOR BATTLE DETERMINATION.
C
TIME=IC=10
DO 57 I=K,L
481 IF (IUSTAT(I).EQ.2) GO TO 57
482 DO 58 J=1,NUA
483 IF (IUSTAT(J).EQ.2) GO TO 58
484 CHECK=X(I)-X(J)
485 AVO=SQR((X(I)-X(J))^2+(Y(I)-Y(J))^2)
486 IF (AVO.LT.BREAK.OR.CHECK.LT.S0.) GO TO 250
500 58 CONTINUE
501 57 CONTINUE
502 GO TO 99
503 C
504 C COMPLETE BLUE MOVE
505 C
506 250 DO 251 I=K,L
507 IF (ALT.EQ.1 .OR. IMOVE(I).EQ.ITEM) GO TO 6000
508 IF (IUSTAT(I).EQ.0) IUSTAT(I)=3
509 IMOVE(I)=IMOVE(I)+1
510 IF (IMOVE(I).LT.ITEM) GO TO 251
511 X(I)=X(R(I))
512 Y(I)=Y(R(I))
513 IF (IUSTAT(I).EQ.3) IUSTAT(I)=0
514 251 CONTINUE
515 99 WRITE (6,112) TIME
516 112 FORMAT (///1X,'TIME = ',I4,1X,'SECONDS'///)
517 WRITE (6,113)
518 113 FORMAT (1X,'UNIT',5X,'X',5X,'Y',5X,'FORCE LEVEL',2X,'STATUS',
519 12X,'LOST-PCT',2X,'TARGETS')
520 DD 59 I=1,L
521 NB=NT(I)
522 IF (NB.NE.0) GO TO 48
523 WRITE (6,204) I,X(I),Y(I),FL(I),IUSTAT(I),TPOL(I)
524 204 FORMAT (3X,I,3X,F7.1,2X,F7.1,6X,F3.1,9X,1,6X,F5.3)
525 C GO TO 59
526 48 WRITE (6,114) I,X(I),Y(I),FL(I),IUSTAT(I),TPOL(I),
527 114 FORMAT (3X,I,3X,F7.1,2X,F7.1,6X,F3.1,9X,1,6X,F5.3,3X,3(I,1X))
528 59 CONTINUE
530 C
531 C CHECK FOR BATTLE DETERMINATION.
532 C
533 IOT=0
534 DO 59 I=1,NRU
535 IF (FL(I).EQ.0) GO TO 53
536 IOT=1
537 53 CONTINUE
538 IF (IOT.EQ.1) GO TO 54
539 WRITE (6,117)
540 117 FORMAT (1X,'** RED FORCE IS ELIMINATED. END OF BATTLE.**')
541 GO TO 68
542 54 IOT=0
543 DO 55 I=K,L
544 IF (FL(I).EQ.0) GO TO 55
545 IOT=1
546 55 CONTINUE
547 IF (IOT.EQ.1) GO TO 65
548 WRITE (6,118)
549 118 FORMAT (1X,'** BLUE FORCE IS ELIMINATED. END OF BATTLE.**')
550 GO TO 66

59
GOOD WRITE (6,119)
119 FORMAT(IX, '*** DISTANCE BETWEEN FORCES IS TOO CLOSE. END OF BATTLE')

GOTO 66
66 IC=IC+1
GOTO 67
STOP
END
SUBROUTINE LOC(XA,YA,THACR,THICA,SIZA,XB,YB,THACB,THICB,SIZEB,
-LATOA,LATOB,VISFRA,VISFRA);

C COMMON /HILLS/ XC(100),YC(100),PEAK(100),SX(100),SY(100),ANG(100)
5 COMMON /HILLS/ SCALE(100),THMHO(100),THOSCL(100),BASE
6 COMMON /HILLS/ WMILLS
7 COMMON /COVER/ CXC(150),CTC(150),CPFRA(150),CFXX(150),CFYY(150)
0 COMMON /COVER/ CPXY(150),NCVELS
9 COMMON /COUNTRA/KM, WH, KN, KGS, KELL, KINT
10 COMMON /GRID/ LST(10,10),MHL(10,10),LISTH(150),KHAEP(100),KTRAP
11 COMMON /GRID/ LSTC(10,10),NC(10,10),LISTC(150),KKREAP(150)
12 DIMENSION 1G(100),1G1(100),1G2(100),1G3(100),1G4(100),1G5(100),1G6
13 DATA NGRAI/10,NGSIZE/1000./
14 C SUBROUTINE TO COMPUTE FRACTION VISIBLE FOR OBSERVER TARGET PAIRS
15 VISFRA=1.
16 VISFRA=1.
17 XBA=XB-XA
18 YBA=YB-YA
19 IF((XBA.EQ.0.).AND.(YBA.EQ.0.)) RETURN
20 IF(SIZA.THICA.LE.0.) GO TO 510
21 IF(SIZEB.THICB.LE.0.) GO TO 510
22 IF(TNACA.LT.0.) VISFRA=1.0+THICA/SIZA
23 IF(TNACB.LT.0.) VISFRA=1.0+THICB/SIZEB
24 ZA=THACR+THICA+SIZA
25 ZB=THACB+THICB+SIZEB
26 KTRAP=KTRAP+1
27 ZBA=ZB-ZA
28 XBSQ=XBA=XBA
29 YBSQ=YBA=YBA
30 XBA=XBA=YBA
31 YB=XB=2.XB
32 XINC=GSIZE/XBA
33 IXIC=GSIZE/XBA
34 IF(XBA) 110,95,100
35 95 XBA=0.1
36 100 I3G=1
37 XINC=GSIZE/XBA
38 GO TO 120
39 120 YBA=0.1
40 130 I3G=1
41 YINC=GSIZE/YBA
42 IF(YBA) 140,125,130
43 125 YBA=0.1
44 130 I3G=1
45 YINC=GSIZE/YBA
46 GO TO 150
47 150 XINC=GSIZE/XBA
48 IF(XBA.5.GT.NGRID) IX=NGRAI
50
IY=1+IFIX(YB/FSIZE)
IF(IY.GT.NGRID) IY=NGRID
XNEXT=FSIZE*(FLOAT(IX)+0.5*(ISGX-1.1))
YNEXT=FSIZE*(FLOAT(IY)+0.5*(ISGY-1.1))
XSTEP=(XB-XNEXT)/XBA
YSTEP=(YB-YNEXT)/YBA
NGRSQ=NGRSQ+1
IXG(NGRSQ)=IX
IGT(NGRSQ)=IY
IF((XSTEP.GT.1.1).AND.(YSTEP.GT.1.1)) GO TO 200
IF(XSTEP.YSTEP) 170,180,190
170 IX=IX+ISGX
XSTEP=XSTEP*XINC
GO TO 160
180 IX=IX+ISGX
XSTEP=XSTEP*XINC
190 IY=IY+ISGY
YSTEP=YSTEP*YINC
GO TO 160
200 KGRS=KGRS+NGRSQ
C GRID SQUARE LIST_NOW COMPLETE IN IXG, IGT WITH NGRSQ ENTRIES

C NOW FIND WHICH COVER ELLIPSES TOUCH THE A TO B LINE,
C CHECK ELEVATIONS AT S1 AND S2 FOR EACH SUCH ELLIPSE
NELS=0
CHMTX=0.
IF(INCVELS.EQ.0) GOTO 270
DO 260 K=1,NGRSQ
IX=IXG(K)
IY=IGT(K)
M=INC(IX,IY)
IF(IN.EQ.0) GOTO 260
LS=LSTC(IX,IY)
LEND=LS+M-1
DO 250 LS=LENS,LEND
KELL=KELL+1
IC=LSTC(L)
IF(KCREP(IC).EQ.KTREP) GO TO 250
KCREP(IC)=KTREP
RX=XACX(CIC)
TY=TACY(CIC)
PXX=CPXX(CIC)
PTY=CPYY(CIC)
PXY=CPXY(CIC)
AR=PXK*XYASQ+PTY*YBASQ+PXK*XYBA
BB=PXX*XYASR+RX*PTY*YWOTBA+RT*PXY*(RX=YBA+AT=XB)
CC=PXX+RX*PTY+RT*PXY+RX*AT-1.0
ARG=BB=88.0=AR*CC
IF(ARG.LE.0.) GOTO 250
S=S+ARG(ARG)
GO TO 250
101  S1 = (BB+50)/(2.0-RA)
102  S2 = (50-BB)/(2.0-RA)
103  IF(S1.GE.1.)  GO TO 250
104  IF(S2.LE.0.)  GO TO 250
105  IF(S1.LE.0.)  GO TO 510
106  IF(S2.GE.1.)  GO TO 510
107  C NOW CHECK LOS AT S1 AND S2
108    KINT=KINT+1
109    CPK=CPK+1(IC)
110  XI=XR+S2*YBA
111    YS=TR+S2*YBA
112  CALL ELEV(XS,TS,HTS)
113  HTS=HTS+CPK
114    ZS=ZS+S2*ZBA
115  IF(LATOB.EQ.0) GO TO 210
116  CALL KOVER(ZB,THACB,SIZEB,ZS,HTS,TS,VSFRA)
117  IF(VISFRA.LE.0) GO TO 510
118  210  IF(LBTOR.EQ.0) GO TO 220
119    S=1.-S2
120  CALL KOVER(ZB,THACB,SIZEZ,ZA,S,HTS,TS,VSFRA)
121  IF(VISFRA.LE.0) GO TO 510
122  220  XS=XA+S1*YBA
123    YS=TR+S1*YBA
124  CALL ELEV(XS,TS,HTS)
125  HTS=HTS+CPK
126    ZS=ZS+S1*ZBA
127  IF(LATOB.EQ.0) GO TO 230
128  CALL KOVER(ZA,THACB,SIZEZ,ZS,HTS,TS,VSFRA)
129  IF(VISFRA.LE.0) GO TO 510
130  230  IF(LBTOR.EQ.0) GO TO 240
131    S=1.-S1
132  CALL KOVER(ZB,THACB,SIZEZ,ZA,S,HTS,TS,VSFRA)
133  IF(VISFRA.LE.0) GO TO 510
134  240  NELS=NELS+1
135  IEL(NELS)=IC
136  CS1(NELS)=S1
137  CS2(NELS)=S2
138  IF(CPK.GT.CHMMAX) CHMMAX=CPK
139  250  CONTINUE
140  260  CONTINUE
141  C ALL ELLIPSES CHECKED
142  C
143  C NOW START ON THE HILLS
144  270  DO 600 K=1,NMAX
145  JJ=1.6KX(IN)
146  IT=ITY(K)
147  IF(NHL(JX,IT).EQ.0) GO TO 600
148  LS=LST(JX,IT)
149  LEND=LS+NHL(JX,IT) -1
150  DO 500 L=LS,LEND
500  CONTINUE
I=LISTM(1)
IF(KHREP(I).EQ.KTREP) GO TO 500
KHREP(I)=KTREP
C PROCESSING FOR HILL I STARTS HERE
KM=KM+1
C COMPUTE W-TOP OF THIS HILL ALONG O-T LINE
CX=XBA/SX(1)
CT=YBA/SY(1)
DX=(XA-XC(I))/SX(I)
DY=(YA-YC(I))/SY(I)
FG=TNOSCL(I)*((CX+DX)*CT+DY)*RHO(I) = (CX+DY)*CY*DX)
GO=SCALE(I)*((CX+CX*CT+TNORHO(I)) =CX*CT)
IF(GO .EQ.0.) GO TO 500
W=FG/(2.*GO)
IF(ABS(W).GT.5.) GO TO 500
FSQ=FG*FG
EQ=SCALE(I)*((DX+DY)*DY+TNORHO(I)) =DX*DY)
POWER=EQ-FSQ/(4.*GO)
IF(Power .LT. -3.) GO TO 500
MH=PEAK(I)*EXP(Power)
KH=KH+1
 IF(MH.LT.LE.BASE) GO TO 500
ZW=ZA+MH*ZBA
 IF(W.LT.0.).OR.(W.GT.11)) GO TO 300
 IF(MH .GE. ZW) GO TO 510
CVHTW=Q0.
 IF(NELS.EQ.0) GO TO 300
 DO 280 N=1,NELS
 IF(CS1(M).GE.MI.OR.(CS2(M).LE.MI)) GO TO 280
 IC=IEL(N)
 IF(CVHTW.LT.CPRE(K)) CVHTW=CPEAK(K)
 CONTINUE
 IF(CVHTW.LT.CPRE(K)) CVHTW=CPEAK(K)
 280 CONTINUE
 IF(INHM+CVHTW).GE.ZW) GO TO 510
 IF(INHM+CVTM).GE.ZH) GO TO 510
 IF(INHM+CVTM).GE.ZW) GO TO 510
 IF(INHM+CVTM).GE.ZH) GO TO 510
C IF WE GET TO HERE THEN NEED TO FIND LOWEST SIGHT LINE OVER HILL
C NEWTON ITERATION A TO B GIVING VISFAD
 IF(LATOB.EQ.O) GO TO 400
 KV=KV+1
 V=W
 MH=MHV
 NCT=O
 FY=FSQ*V
 TwOGy=2.*GO=V
 IF(TwOGy=2.*GO=V) GO TO 400
 FCNV=ZA+HHV*(TwOGy=V+FY-1.)
 KN=KN+1
 FACTOR= (TwOGy=TwOGy+2.* (GO+TwOGy=FG) *FSQ)
 DFCNV=HHV*V*FACT0R
 IF(ABS(DFCNV) .LT.1.E-10) GO TO 350
 V=V*FCNV/DFCNV
 IF(ABS(V).GT.5.) GO TO 400
201   FY=FQ=V 202   THQGV=2.*GQ=V 203   POWER = EQ+FV=GQ=V=V 204   IF (POWER .LT. -3.) GO TO 400 205   MHV=PEAK(1)=EXP(POWER) 206   DNHV=MHV+(FQ+TWQGV) 207   ELV=ZB+DNHV=W 208   IF (ABS(MHV-ELV) .LT. 1.) GO TO 350 209   NCT=NCT+1 210   IF (NCT.LT.10) GO TO 330 211   350  IF ((V.LT.0.) .OR. (V.GT.1.)) GO TO 400 212   CVHTV=0. 213   IF (NELS.EQ.0) GO TO 390 214   GO 380 M=1,NELS 215   IF ( (C(1).GE.0) .OR. (C(2).LE.0) ) GO TO 380 216   IC=1EL(M) 217   IF (CVHTV.LT.CPEAK(IC)) CVHTV=CPEAK(IC) 218   CONTINUE 219   380 400  IF ((LBTOR.EQ.0) GO TO 500 220   KV=KV+1 221   V=N 222   VM1=V-1. 223   MHV=MHW 224   NCT=0 225   FY=FQ=V 226   THQGV=2.*GQ=V 227   430  FCNV=ZB+MHV+((FQ+TWQGV)=VM1-1.) 228   KN=KN+1 229   FACTOR=(THQGV+TWQGV=2.*GQ+TWQGV=FD)*FSQ) 230   DFCNV=MHV=VM1*FACTOR 231   IF (ABS(DFCNV) .LT. 1.E-10) GO TO 450 232   V=V-FCNV/DFCNV 233   IF ((V.LT.0.3) .OR. (V.GT.1.33) GO TO 500 234   VM1=V-1. 235   FY=FQ=V 236   THQGV=2.*GQ=V 237   POWER = EQ+FV=GQ=V=V 238   IF (POWER .LT. -3.) GO TO 500 239   HHV=PEAK(1)=EXP(POWER) 240   DNHV=MHV+(FQ+TWQGV) 241   ELV=ZB+DNHV=W 242   IF (ABS(MHV-ELV) .LT. 1.) GO TO 450 243   NCT=NCT+1 244   IF (NCT.LT.10) GO TO 430 245   450  IF ((V.LT.0.) .OR. (V.GT.1.)) GO TO 500
251  CVHTV=0.
252  IF (NELS.EQ.0) GO TO 490
253  DO 480 M=1,NELS
254  IF ((C51(N).GE.VI).OR.(C52(N).LE.VI)) GO TO 480
255  IC=IEL(N)
256  IF (CVHTV.LT.CPEAK(IC)) CVHTV=CPEAK(IC)
257  480  CONTINUE
258  490  HTV=MHV+CVHTV
259  ZV=ZP+V*ZBA
260  S=-VMI
261  CALL KOVER(ZB,THACA,SIZEA,ZA,S,HTV,ZV,VISFRA)
262  IF (VISFRA.LE.0.) GO TO 510
263  500  CONTINUE
264  600  CONTINUE
265  RETURN
266  510  VISFRA=0.
267  VISFRB=0.
268  RETURN
269  END
SUBROUTINE SETUP

C COMMON /HILLS/ XC(100), YC(100), PEAK(100), SX(100), ST(100), RHO(100)
C COMMON /HILLS/ SCALE(100), THORHO(100), TWOSCL(100), BASE
C COMMON /HILLS/ NHILLS
C COMMON /COVER/ XCIC150), YCIC150), PEAK(150), CPXX(150), CPXY(150)
C COMMON /COVER/ CPXT(150), MCEVELS
C COMMON /GRID/ LSTC10,10), NC(10,10), LISTC(100), KCREP(150)
C COMMON /GRID/ NHILLS
C COMMON /COVER/ XC(1501, YC(150), PEAK(150), CPXX(150), CPYT(150)
C COMMON /GRID/ NC(10,10), LISTC(100), KCREP(150)

READ(8,7) NHILLS
READ(8,47) BASE
READ(8,7) LST
READ(8,7) NHILLS
READ(8,7) NCVELS
READ(8,7) NCTOT
READ(8,7) (LISTC(I,1), I=1,NCTOT)
READ(8,7) (LISTH(I,1), I=1,NHTOT)
READ(8,7) LSTC
READ(8,7) NC
READ(8,7) NCTOT
READ(8,7) (LISTC(I,1), I=1,NCTOT)
READ(8,7) (LISTH(I,1), I=1,NHTOT)

DO 100 I=1,NHILLS

SCALE(I)=1.25
ST(I)=ST(I)*1.25
XC(I)=(XC(I)-500.)/100.
YC(I)=(YC(I)-900.)/100.
THORHO(I)=TWO+RHO(I)
SCALE(I)=1./(-2.*RHO(I)=2))
TWOSCL(I)=2.*SCALE(I)
KCREP(I)=2147483600

C ALL VALUES NOW IN METERS ON 0 -- 10,000 GRID

CONTINUE
51 KGRS=0
52 KELL=0
53 KINT=0
54 RETURN
55 END
SUBROUTINE ROUTE

COMMON /GRPS/ NBU, NNU, FL(6), FO(6), NOI(3), XIC(3,200), YIC(3,200),
1IDIIR(3,200), AVSP, ISPD
1, IUSTAT(6), IL(6), LOST(6,6), VISFRA, VISFAB, SIZETK,
7 SIZETM, NT(6), NF(6), SRF, DSMAX,
8 INLOC(6,6), VISF(6,6), RMINTK, RMXTK, RMINTM, DQ, TOWFD, TOWFR,
9 IPTT(3,3), RF, POR(6,6), APC0(6,6), LOA(6,6), NA(6), QFL(6), POL(6)
10 DIMENSION XLOC(3,20), TLOC(3,20), N(3)
11 IF(IPSP.EQ.4) DST=80.463
12 IF(IPSP.EQ.3) DST=67.053
13 IF(IPSP.EQ.2) DST=33.643
14 IF(IPSP.EQ.1) DST=40.232
15 DO 300 I=1, N
16 READ(5,15) N(I)
17 FORMAT(12)
18 NL=N(I)+1
19 DO 200 IN=2, NL
20 READ(5,201) XLOCs, TLOCs
21 FORMAT(F6.1,IX,F6.1)
22 XLOC(I,IN)=XLOCs
23 TLOC(I,IN)=TLOCs
24 CONTINUE
25 XLOC(I,1)=XIC(I,1)
26 TLOC(I,1)=YIC(I,1)
27 IDIR(I,1)=0
28 NL=N(1)
29 NUM=2
30 DO 305 J=1, NL
31 XL=XLOC(I,J+1)-XLOC(I,J)
32 YL=TLOC(I,J+1)-TLOC(I,J)
33 DIST=SQR(XL**2+YL**2)
34 T=ABS(YL)
35 Z=T/XL
36 ANGL=ATAN(Z)
37 DEG=ANGL*57.2958
38 IF(J.EQ.1) GO TO 320
39 XLN=(DIST-EXTRA)*COS(ANGL)
40 DIST=(DIST-EXTRA)-DIST
41 YLN=(DIST-EXTRA)*SIN(ANGL)
42 XIC(I,NUM)=XIC(I,NUM-1)+XLN+XLE
43 IF(YL.GT.0.) GO TO 325
44 YLN=-YLN
45 325 YIC(I,NUM)=YIC(I,NUM-1)+YLN+YLE
46 IF(YL.GT.0.) GO TO 340
47 IDIR(I,NUM)=-IFIX(DEG)
48 GO TO 341
49 340 IDIR(I,NUM)=IFIX(DEG)
50 341 NUM=NUM+1
51 XLN=DST=COS(ANGL)
52 YLN=DST=SIN(ANGL)
53 IF (YL.GT.0.) GO TO 310
54 YLN=-YLN
55 310 IF (DIST.LT.DST) GO TO 315
56 XIC(I,NUM)=XIC(I,NUM-1)+XLN
57 TIC(I,NUM)=TIC(I,NUM-1)+YLN
58 IF (YL.GT.0.) GO TO 342
59 IDIR(I,NUM)=-IFIX(DEG)
60 GO TO 343
61 342 IDIR(I,NUM)=FIX(DEG)
62 343 DIST=DIST-DST
63 NUM=NUM+1
64 GO TO 310
65 315 EXTR=DIST
66 XLE=EXTRA=COS(ANGL)
67 YLE=EXTRA=SIN(ANGL)
68 IF (YL.GT.0.) GO TO 305
69 YLE=-YLE
70 305 CONTINUE
71 300 CONTINUE
72 RETURN
73 END
SUBROUTINE LANDA(I,J,PCTVIS,DETRAT,PK)
SUBROUTINE TO COMPUTE DETECTION RATE (DETRAT) OF TARGET J
BY OBSERVER I GIVEN THE VISIBLE FRACTION (PCTVIS).

COMMON /GRPI/ IPADIR(8), ISECWO(8), MVTDIR(6), X(8), Y(8), SPD(6)
TCFACT=1.0
ZEROL=0.00001

PAI=22.0/7.0

O=ISECWO(1)*PAI/180.0 / 2.0
BBB=(1.0/(2.0*(SIN(O)-D+COS(O))))
AAA=1-BBB*COS(O)
IF(AABS(AAA).LT.ZEROL) AAA=0.0
QATN=ATAN2((Y(J)-Y(I)),(X(J)-X(I)))
PD=IPADIR(I)*PAI/180.0
IF((PDQATN).GE.0.0) GOTO 1
IF(PD_LT.0.0) GOTO 9
ANGLE=2*PAI+QATN-PD
GOTO 10

9 ANGLE=2*PAI-PD-OATN

IF(ANGLE.GT.PAI) ANGLE=2*PAI-ANGLE
GOTO 2
1 ANGLE=AABS(PD-OATN)
GOTO 3
2 IF(ANGLE.GT.D0) GOTO 3
DUP=PD*D
DLCN=PD-D
ANGLFT=QATN+(360*PAI/180.0)
IF(ANGLFT.GT.DUP) ANGLFT=DUP
ANGLRT=QATN-(360*PAI/180.0)
IF(ANGLRT.DL low) ANGLRT=ANGLRT
PK=AAA*(SIN(ANGLFT)-SIN(ANGLRT))*AAA*(ANGLFT-ANGLRT)
IF(PK_LT.0.0) GOTO 3
IF(PK_GT.1.0) GOTO 5
GOTO 8

3 PK=0.0
DETRAT=0.0
GOTO 6

5 PK=1.0
RANGE=SQRT((X(J)-X(I)**2+(Y(J)-Y(I)**2)
RR=0.001*RANGE/PCTVIS
TOANG=ATAN2((Y(J)-Y(I)),(X(J)-X(I)))
AD=MVTDIR(J)+PAI/180.0
HAVAL=ABS(SPD(J)*SIN(TOANG-AD))
DENOM=1.453*TCFACT* (0.59753*2.188*(RR**2)-0.5038*HAVAL)
IF(DENOM_LE.ZEROL) DENOM=ZEROL
DETRAT=0.003*1.088/DENOM
DETRAT=DETRAT+PK
RETURN
SUBROUTINE ELEV(X,Y,TMAC)

C

COMMON /HILLS/ XC(100),YC(100),PEAK(100),SX(100),ST(100),RHS(100)
COMMON /HILLS/ SCALE(100),THRM(100),THOSCL(100),BASE
COMMON /HILLS/ MIILLS
COMMON /GRID/ LST(10,10),NML(10,10),LISTH(4501),KHP(100),KTRP
COMMON /GRID/ LSTC(10,10),NC(10,10),LISTC(1000),KCP(150)
DATA NGRID/10/,GSIZE/1000./
C FUNCTION TO COMPUTE TERRAIN ELEVATION FOR GIVEN X, Y COORDINATES.
ZMAX=BASE
IX=IX+IFIX(X/GSIZE)
IF(IX.GT.NGRID) IX=NGRID
IY=IY+IFIX(Y/GSIZE)
IF(IY.GT.NGRID) IY=NGRID
IF(NML(IX,IY).EQ.0) GO TO 150
LS=LST(IX,IY)
LEND=LS-NML(IX,IY)-1
DO 100 L=LS,LEND
I=L-1
QX=(X-XC(I))/SX(I)
QXSQ=QX*QX
IF (QXSQ .GE. 9.) GO TO 100
QT=(Y-YC(I))/ST(I)
QTSQ=QT*QT
IF (QTSQ .GE. 9.) GO TO 100
OXY=THRM(I)+QX*QT
FACTOR=SCALE(I)*(QXSQ+QTSQ+QXT)
IF (FACTOR.LT.-3.) GO TO 100
HT=PEAK(I)*EXP(FACTOR)
IF (HT.LE.ZMAX) GO TO 100
ZMAX=HT
100 CONTINUE
TMAC=ZMAX
RETURN
END
SUBROUTINE STOC(I,RANGE,A)
C
SUBROUTINE TO COMPUTE STOCHASTIC ATTRITION COEFFICIENT
C
COMMON /GRP6/ ALPH(6)
COMMON /GRPS/ NBU,NAU,FL(6),F0(6),NOI(3),XIC(3,200),YIC(3,200),
10IR(3,200),RVSP,ISPD
8 I,USTAT(6),IL(6),LOST(6,6),VISFRA,VISFRA,SIZEK,
9 ISIZETM,NT(6),MF(6),SAF,DISHAX,
10 INLOS(A,6),VISFR(6,6),RMINTK,RMINTK,RMINTK,RMINTK,OP,TOWFA,TNKFR,
11 IPTT(3,3),RF,PDA(6,6),APDA(6,6),LDA(6,6),NA(6),DFL(6),PDL(6)
12 IF(1.EQ.2) GO TO 5003
13 A=ALPHA(1)*((1.0-RANGE/RMXTW)**2)
14 GO TO 5004
5003 A=ALPHA(1)*((1.0-RANGE/RMXTK)**2)
5004 RETURN
END
SUBROUTINE ETHLI.RRNGE.T

SUBROUTINE TO COMPUTE EXPECTED TIME TO KILL A TARGET.

COMMON /GRP2/ TR(2), T1(2), TH(2), TM(2), TF1(2), TF2(2), TF3(2),
IP(2,6), PHM(2,6), PHM(2,6), PKH(2,6), TF(2)
IF (I.EQ.2) GOTO 5
TF(1)=TF(1)
GOTO 6
5 IF (RANGE.GT.1000.0) GOTO 7
TF(1)=TF(1)-(TF(1)*(1000.0-RANGE)/1000.0)
GOTO 6
7 IF (RANGE.GT.2000.0) GOTO 8
TF(1)=TF(1)-(TF(1)-TF(1))*(2000.0-RANGE)/1000.0)
GOTO 6
8 TF(1)=TF(1)-(TF(1)-TF(1))*(3000.0-RANGE)/1000.0)
J=(RANGE+250.0)/500.0
IF (J.GT.8) J=8
T=TR(1)-T1(1)-TH(1)+(TM(1)+TF(1))/PKH(1,1)+((TM(1)+TF(1))/
1PHM(1,1))/PKH(1,1) PHM(1,1) PHM(1,1) PHM(1,1) PHM(1,1) PHM(1,1)
RETURN
END
C
SUBROUTINE SORT(I,M)
COMMON /GRPS/ LOT(6,6), ROT(6,6)
DO 19 J=1,M
19 IF (ROT(I,M).GE.ROT(I,J)) GOTO 19
21 A=ROT(I,J)
   NN=LOT(I,J)
   ROT(I,J)=ROT(I,M)
   LOT(I,J)=LOT(I,M)
   ROT(I,M)=A
   LOT(I,M)=NN
19 CONTINUE
RETURN
END
SUBROUTINE KOVER (Z0, TMACT, SIZET, ZT, S, HTS, ZS, VISFAT)

C

IF (S.EQ.0.) GO TO 2000

IF (HTS.GE.ZS) GO TO 2050

NEXT = Z0 + (HTS - Z0) / S

EVIST = AMAX1 (NEXT, TMACT)

IF (EVIST.GE.ZT) GO TO 2050

IF (EVIST.LE.ZT - SIZET) RETURN

VIS = (ZT - EVIST) / SIZET

IF (VIS.LT.VISFRT) VISFRT = VIS

RETURN

2000 IF (HTS.LT.ZO) RETURN

2050 VISFRT = 0.0

RETURN

END
APPENDIX C

Definition of Variables in Computer Program

<table>
<thead>
<tr>
<th>Variable</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALPHA(I)</td>
<td>Initial attrition-rate coefficient for stochastic attrition module.</td>
</tr>
<tr>
<td>APOA(I,J)</td>
<td>The average proportion of the ( j^{th} ) attacker of unit ( i ) allocated to fire on unit ( i ).</td>
</tr>
<tr>
<td>AVSP</td>
<td>Average speed of moving attacking units.</td>
</tr>
<tr>
<td>BREAK</td>
<td>Breakpoint distance between attackers and defenders.</td>
</tr>
<tr>
<td>DISMAX</td>
<td>Maximum distance allowed between attacking units before the leading unit is delayed.</td>
</tr>
<tr>
<td>DIST</td>
<td>The straight-line distance between two movement nodes inputed by the user.</td>
</tr>
<tr>
<td>DST</td>
<td>The distance to be moved each time step by attacking units.</td>
</tr>
<tr>
<td>FL(I)</td>
<td>Force level of unit ( i ).</td>
</tr>
<tr>
<td>FC(I)</td>
<td>Initial force level of unit ( i ).</td>
</tr>
<tr>
<td>IALT</td>
<td>Denotes whether the user desires alternate defensive positions or not.</td>
</tr>
<tr>
<td>IC</td>
<td>Counts number of time units a defender has been moving.</td>
</tr>
<tr>
<td>IDIR(I,J)</td>
<td>Direction of ( j^{th} ) interval in ( i^{th} ) route.</td>
</tr>
<tr>
<td>II(I)</td>
<td>Interval index for unit ( i ).</td>
</tr>
<tr>
<td>IMOVE</td>
<td>Number of time units a defender is allowed for moving to an alternate position.</td>
</tr>
<tr>
<td>IPRDIR(I)</td>
<td>Primary direction of movement for unit ( i ).</td>
</tr>
<tr>
<td>IRTE</td>
<td>Denotes whether user wants to input routes or not.</td>
</tr>
<tr>
<td>IS</td>
<td>Random number seed used for stochastic attrition.</td>
</tr>
<tr>
<td>ISECWD(I)</td>
<td>Width of search sector for unit ( i ).</td>
</tr>
</tbody>
</table>
ISPD = Input variable to denote user's desired speed for attackers movement.
ITEM = Input variable denoting number of time steps allowed for defender's move.
ITIME = Current time, in seconds, of battle.
ITRIT = Input variable denoting whether attrition will be stochastic or deterministic.
IUSTAT(I) = Current status of unit i.
LCA(I,J) = The number of the jth attacker of unit i.
LOST(I,J) = Denotes whether line-of-sight exists between unit i and j or not.
LOT(I,J) = The number of the jth target of unit i.
MVTDIR(I) = Movement direction of unit i.
N(I) = Number of nodes inputed by user for route i.
NA(I) = Number of attackers of unit i.
NBU = Number of blue units.
NF(I) = Number of time units unit i is allowed to fire at the same location.
NLCSC(I,J) = Number of continuous time steps that line-of-sight does not exist between unit i and unit j.
NOI(I) = Number of intervals in the ith route.
NRU = Number of Red Units.
NT(I) = Number of targets of unit i.
OFL(I) = Force level of unit i during previous time step.
P(I,J) = Probability of 1st round hit by unit i in range band j.
PHH(I,J) = Probability of a hit following a hit by unit i in range band j.
\( P_{HI}(I,J) \) = Probability of a hit following a miss by unit i in range band j.

\( P_{K}(I,J) \) = Probability of a kill given a hit by unit i in range band j.

\( F_{t} \) = The proportion of time a moving unit is searching for targets.

\( PCA(I,J) \) = The proportion of the \( j^{th} \) attacker of unit i allocated to fire on unit i.

\( PCL(I) \) = Percent of unit i lost during the current time step.

\( FTT(I,J) \) = Proportion of surviving firepower allocated to the \( i^{th} \) target if there are j targets available.

\( RANGE \) = Current minimum distance between attackers and defenders.

\( RMTK \) = Minimum effective range for attacking weapon system.

\( RMTW \) = Minimum effective range for defending weapon system.

\( RMTK \) = Maximum effective range for attacking weapon system.

\( RMTW \) = Maximum effective range for defending weapon system.

\( RCT(I,J) \) = The range of the \( j^{th} \) target of unit i.

\( SIZET \) = Size of attacking vehicle.

\( SIZET' \) = Size of defending vehicle.

\( TA(K) \) = Time to acquire a target for \( k^{th} \) weapon system type \((k = 1,2)\).

\( TF1(K) \) = Time of flight to 1000m for \( k^{th} \) weapon system type \((k = 1,2)\).

\( TF2(K) \) = Time of flight to 2000m for \( k^{th} \) weapon system type \((k = 1,2)\).

\( TF3(K) \) = Time of flight to 3000m for \( k^{th} \) weapon system type \((k = 1,2)\).
\( TH(k) \) = Time to fire a round following a hit for weapon system type \( k \) (\( k = 1,2 \)).

\( TI(k) \) = Time to fire first round after target has been acquired for weapon system type \( k \) (\( k = 1,2 \)).

\( Tn(k) \) = Time to fire a round following a miss for weapon system type \( k \) (\( k = 1,2 \)).

\( TFR \) = Firing rate for attacking weapon system.

\( TCWFR \) = Firing rate for defending weapon system.

\( TPOL(i) \) = Total percentage of lost since battle began for unit \( i \).

\( VISFR(i,j) \) = The fraction of unit \( i \) seen by unit \( j \).

\( VISFRA \) = Fraction of unit \( A \) as seen by unit \( B \).

\( VISFRA \) = Fraction of unit \( B \) as seen by unit \( A \).

\( X(i), Y(i) \) = Coordinates of unit \( i \).

\( XA(i), YA(i) \) = Coordinates of alternate position for defender \( i \).

\( XIC(i,j), YIC(i,j) \) = Coordinates of the \( j^{th} \) interval endpoint of the route for unit \( i \).

\( XL, YL \) = Distance added to previous interval endpoint for vehicle to move DST during a time step.

\( XLOC(i,j), YLOC(i,j) \) = Coordinates of the \( j^{th} \) node inputed by the user for the route of unit \( i \).
BELICGRAPHY

Bonder, S., "The Lanchester Attrition-Rate Coefficient," 

Bonder, S., "The Mean Lanchester Attrition Rate," 

Clark, G. M., The Combat Analysis Model, Ph.D. Thesis, 
Ohio State University, Columbus, Ohio, 1969.

Karr, A. F., Stochastic Attrition Models of Lanchester 
Type, P-1031, Institute for Defense Analysis, Arlington, 
Virginia, June 1974.

Smoler, J., Operational Lanchester-Type Model of 
Small Unit Land Combat, M.S. Thesis, Naval Postgraduate 
School, Monterey, California, 1979.

Taylor, J. G., Force-on-Force Attrition Modelling, 
p. 20-89, Military Applications Section of Operations 
Research Society of America, January 1980.

Taylor, J. G., "Solving Lanchester-Type Equations 
for 'Modern Warfare' with Variable Coefficients," 

Wallace, W. S., and Hagwood, E. G., Simulation of 
Tactical Alternative Responses (STAR), M.S. Thesis, 
Naval Postgraduate School, Monterey, California, 
December 1978.
<table>
<thead>
<tr>
<th>No.</th>
<th>Distribution List</th>
<th>Copies</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Defense Technical Information Center</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>Cameron Station</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Alexandria, Virginia 22314</td>
<td></td>
</tr>
<tr>
<td>2.</td>
<td>Library, Code 0142</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>Naval Postgraduate School</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Monterey, California 93940</td>
<td></td>
</tr>
<tr>
<td>3.</td>
<td>Department Chairman, Code 55</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Department of Operations Research</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Naval Postgraduate School</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Monterey, California 93940</td>
<td></td>
</tr>
<tr>
<td>4.</td>
<td>Professor James G. Taylor, Code 55Tw</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>Department of Operations Research</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Naval Postgraduate School</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Monterey, California 93940</td>
<td></td>
</tr>
<tr>
<td>5.</td>
<td>Professor Glenn F. Lindsay, Code 55Ls</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Department of Operations Research</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Naval Postgraduate School</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Monterey, California 93940</td>
<td></td>
</tr>
<tr>
<td>6.</td>
<td>Professor Samuel H. Parry, Code 55Py</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Department of Operations Research</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Naval Postgraduate School</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Monterey, California 93940</td>
<td></td>
</tr>
<tr>
<td>7.</td>
<td>Captain Glenn M. Mills</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>710 Manhasset Road</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Charlotte, North Carolina 28209</td>
<td></td>
</tr>
<tr>
<td>8.</td>
<td>Command and General Staff College</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Attn: Education Advisor</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Room 123, Bell Hall</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ft. Leavenworth, Kansas 66027</td>
<td></td>
</tr>
<tr>
<td>9.</td>
<td>Headquarters, Department of the Army</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>MILPERGEN</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Attn: DAPG-OPF-C</td>
<td></td>
</tr>
<tr>
<td></td>
<td>200 Stovall Street</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Alexandria, Virginia 22332</td>
<td></td>
</tr>
</tbody>
</table>