LA-5806-MS
Informal Report

a.3

Special Distribution
Reporting Date: November 1974
Issued: December 1974

Computer Simulation of Tactical Nuclear Warfare

by

John K. Hayes

los alamos
scientific laboratory
of the University of California
LOS ALAMOS, NEW MEXICO 87544

UNITED STATES
ATOMIC ENERGY COMMISSION
CONTRACT W-7405-ENG. 36

In the interest of prompt distribution, this LAMS report was not edited by the Technical Information staff.

COMPUTER SIMULATION OF TACTICAL NUCLEAR WARFARE

by

John K. Hayes

ABSTRACT

The tactical nuclear warfare model used at LASL is explained in detail. A number of conclusions from the model are also given.

I. INTRODUCTION

This report describes our effort at LASL to model a tactical nuclear engagement. We hope the reader will be able to understand how the model works and can offer critical comments about the effort as a whole. The idea behind the model was to be able to study, qualitatively, the interaction of a number of predetermined and hopefully dominant variables involved in a tactical nuclear battle. We felt it imperative to keep the model as simple as possible.

Complexity makes it impossible to program a model that will fight everybody's version of a tactical nuclear war, and, to simplify our model, we had to make some assumptions about the tactical nuclear battlefield. We felt that conventional tube artillery would have no important role in the nuclear battlefield, so we included none in our model. We are presently looking at ground warfare only. The air war may very well be decisive, but, for the problems of interest to us, we believe the air war and the ground war can be studied independently. The model ignores the problem of release, which is a political problem as much as anything else and is beyond our capabilities. We assume that the weapons are released and that the only constraints are those imposed by weapons-system response capabilities. Another important battlefield function we have ignored so far is supply. For the single, short-duration battle studied to date, supply is unimportant; but, if we ever look at battles lasting days rather than hours, we will have to include supply. Any of the above factors except release can be included in the model with little effort.

One point we'd like to express before we go too far is our belief that leadership, morale, training, and mental effects could well be more important than the weapons systems, tactics, and physical effects we have studied in this model. Unfortunately, we've found no way to study these effects, though we are trying.

Three more report sections follow. Section II gives a general explanation of the important functions in the model. Section III summarizes the more important conclusions from running the model. Section IV explains the functions of the various subroutines and tells what information the more important variables carry.

II. GENERAL EXPLANATION

A. Geographical Approximations in the Model

Most distances in the model are given in meters, time is given in minutes, and nuclear-weapon yields are given in kilotons. The computer model is time-dependent and runs with a time step of about 15 to 30 seconds. The controlling aspect of the time step is how far a unit can travel in one time step. A fixed rule is hard to state, but we wanted a unit to travel no more than about 50 m in a time step. This interval is important in a conventional battle because it determines how far a unit can move before the opposition can react.

The battlefield presently used in the model is 20 km by 80 km. The 20 km represents the width of the battle, and the 80 km represents the possible depth of the battle. We tried to approximate the geography in a sector of W. Germany just north of Fulda. The exact area covered extended in latitude from 8° 43' to 9° 50' and in longitude from 50° 37' to 50° 56'. Rather than attempt full approximation of the geography of the area, we only used the features we thought would be important in a tactical nuclear war. Instead of putting the real road network into the model, we just counted the north-south and the east-west roads and set up a checkerboard-type road network that gave us the same road density. City locations and sizes were another geographical feature in the model. We assumed that cities above a given minimum population defined areas where nuclear fire was denied for the defense. This minimum population was an input parameter. Rather than locate the cities in their true positions, we located them at random on our road network.

Line of sight is another geographical feature we included in the model. A true line-of-sight calculation for the 80- by 20-km sector of W. Germany that we used would be difficult to make. We thought that the ridge lines would be the major factor in determining line of sight for the problems of interest to us. We used a contour map to find the major ridge lines, approximated these ridge lines by line segments, and assumed in the model that one could not see across the line segments. The approximation entails a number of deficiencies and would be inadequate for a conventional battle, but, at the distances of interest to us for visual acquisition over the terrain we are studying, it did not seem an important factor.

B. Initialization

The computer model divides naturally into five parts. The first part to be discussed is the initialization, where the forces are deployed and all geographical inputs are put into a usable form. Routes of travel are assigned to units on the offense. The initialization is performed once for each problem at the start of the problem. The computer time spent in initialization is negligible compared with the rest of the problem.

C. Bookkeeping

The remaining four parts of the model are time-dependent and are functions that are treated each cycle. The first of these functions to be discussed is a bookkeeping routine that moves the units in time. A number of variables determine the maximum speed of a given unit, namely: type of unit (infantry, mechanized infantry, tank, etc.), whether it is day or night, whether it is on- or off-road travel, and whether the unit is or is not engaged in conventional combat. The speeds used were taken from Ref. 1. We assumed that a unit under conventional fire could not move, on average, at any more than 1/3 the speed it could move otherwise. We said that a unit could not move at all if it were engaged in conventional combat with an enemy unit of superior strength and the enemy unit was within, say, 300 meters. This routine also digs in any units that stop during a time cycle. These assumptions make more sense in open terrain than in heavily wooded terrain.

D. Target Acquisition

Visual target acquisition will be discussed first, and target acquisition with unattended ground sensors will be discussed second. The logic that determines whether or not one unit observes another stems from a number of principles listed below. These principles are somewhat biased toward terrain, vegetation, and visibility in New Mexico, but hopefully not enough to have radical effects on the results.

The principles hold true only under the following conditions. The observer must be conscientious in his observing and must have line of sight to the object being observed. During the day, the observer must be using his unaided eyes, and colors must be chosen to minimize the likelihood of observation. Observation is limited to ranges of 4000 m and less. The principles are as follows:

(1) The ability of an observer to see a stationary object is a function of the solid angle the object subtends to the observer.

(2) Assume an observer is looking for a stationary object in a sector of 15° width. Unless the circumstances of observation change, the observer will either see the object within 15 to 30 seconds or he will not see the object at all.

(3) If an observer is told the location of a stationary object, he will be able to discern the object about as well as if it were moving.

(4) A stationary observer or a passenger in a vehicle will observe distant objects before the driver at least 4 out of 5 times if the driver is reasonably attentive to his driving.

(5) A single person standing still can be seen up to about 800 m; a walking person, at about 2000 m. A stopped vehicle can be seen at about 2000 m; a moving vehicle, at up to 4000 m.

What are some of the implications of these principles? The first principle gives us a rough measure of the ability to observe a unit as a function of its size. Just how rough the measure is can be seen by considering a squad of men marching in a column. If the observer looks at the column head on, the solid angle is no greater than it would be for one individual. If the observer looks at the column from the side, the solid angle goes up approximately linearly with the number of men in the squad. The average orientation is between the two extremes, and, in our model, we say the solid angle increases with the cube root of the unit strength.

The second principle tells us that, for the size of time step of interest to us, the probability of observing a unit during a given time step is independent of time-step size. The third principle (plus principle 5) tells us that an observer can see a moving unit or a previously observed unit at about twice the distance he can see a previously unobserved stationary unit. The fourth principle tells us that an individual moving without frequent stops for observation will not be able to acquire targets nearly as well as a stationary individual. In the model, we say that a unit moving at full speed will observe targets at half the distance a stationary unit will.

We now need to discuss how target location error is computed. An observer will probably see only a small part of a unit at any one time. Without other information, the observer must guess that this small visible part is at the centroid of the total unit. The model randomly picks a point in the unit and assumes this is the visible part of the unit. This point is designated the centroid of the unit, though it will seldom coincide with the actual centroid. This introduces one error. Another error comes from the observer's estimate of the actual location of the small part of the unit he can see. In our model, the maximum error in this location estimate varies linearly with the distance from the observing unit to the unit being observed. We use an average error of 0.5% of this distance. The two errors described here are independent.

The velocity of each acquired unit is also estimated. We say that an observer can estimate speed within ±30% and direction within ±30°. If the acquired unit is traveling on a road, the estimated direction is exact.

In acquiring a target at night, we believe an observer using a combination of flares and low-light-level devices could see an object at about half the distance he could see it during the day. For those who disagree with this assumption, we have an option to acquire targets with unattended ground sensors that are laid out in a band perpendicular to the expected direction of attack. One person with a monitoring device will monitor 35 sensors in each kilometer of the sensor band. Fig. 1 shows the sensor layout for one monitor. The sensor range is

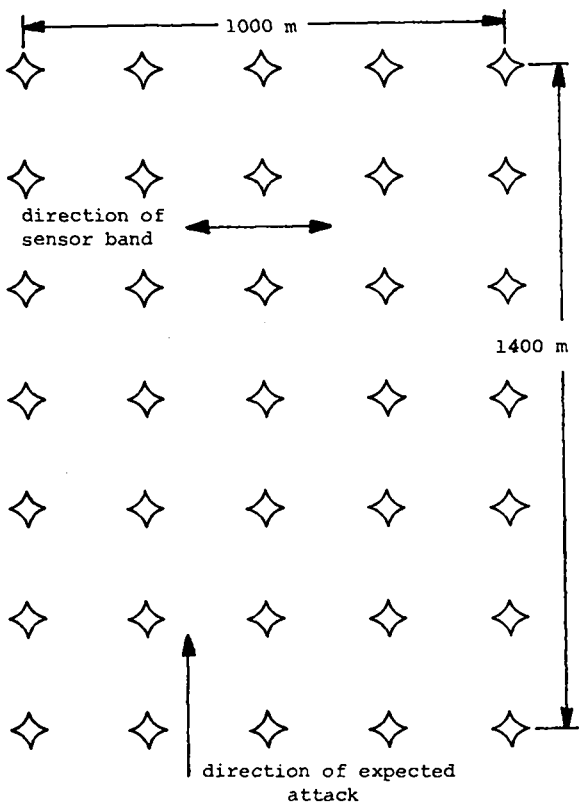

Fig. 1. Sensor field monitored by one observer.

an input variable. We have been using ranges like 125 to 175 m for the detection of armored vehicles. We have also been using a random range variation of about ±30% among the various sensors in the field. For realism, we say that something like 10 to 30% of the sensors, randomly selected, do not work at all in a typical situation. These numbers are within the capability of present-day sensors. Each monitor has a situation display map similar to that described in Ref. 2. The monitor merely looks at his map display, designates a centroid for the activated sensors, and requests fire at the centroid.

E. Conventional Combat

Conventional combat is one of the weaker parts of the model. The main purpose here was to indicate the relative importance of conventional firepower as opposed to nuclear firepower. A number of factors determine whether or not two opposing units will engage in conventional combat. The distance between the units and whether or not they can see each other are two such factors. Equally important is the military mission of the units. A nuclear fire unit might be told to run when it sees the enemy coming; other units might be told to fight only if attacked; and others might be told to seek combat. Once a conventional battle starts, a Lanchester-type equation determines the time rate of casualties for the opposing sides. For two units whose strengths are S_O and S_D, the equations are:

$$\left.\begin{aligned} \frac{dS_O}{dt} &= -\frac{c\alpha_O S_D}{r^2}, \\ \frac{dS_D}{dt} &= -\frac{c\alpha_D S_O}{r^2}, \end{aligned}\right\} \qquad (1)$$

where r is the distance between the two units; α_D (α_O) = 1, 2, or 3 depending respectively on whether the unit S_D (S_O) is dug in, stopped, or moving; and c is an exchange-rate coefficient chosen so that two units of equal strength stopped at a separation distance of 100 m will suffer 5% casualties in one hour. In an engagement involving more than one unit on each side, this calculation is made for each pair of opposing units.

A conventional battle ends whenever one side has either taken the maximum acceptable number of casualties or has moved a sufficient distance from the enemy units to preclude further engagement. A unit is pinned down if it is in a conventional battle at fairly close quarters and cannot inflict substantially more casualties than it is taking.

F. Nuclear Fire

In the problems we have run, the offense has used nuclear fire in preplanned strikes made before they launched an attack. Attacking forces depending on visual target acquisition can not consistently acquire targets for their nuclear weapons among a defensive force unless the defense is inept. By the time an offensive unit can acquire a defensive unit, it is inevitably too close to shoot a nuclear weapon. One idea we wanted to study with the model was that of shooting at attacking offensive maneuver units as they were acquired by the defense. For this concept to work, we had to assume the defense had a quick-reacting, reasonably accurate, nuclear fire unit.

The fire direction center (FDC) in this model requires that a number of tasks be done for each fire mission. The FDC must (1) determine that no two observers are requesting fire on the same target, (2) check to see that a requested fire mission does not endanger friendly troops or violate city safety constraints, (3) find a nuclear fire unit that is within range of the target and ready to fire, and (4) transmit the fire request to the available unit. A reasonable amount of time is allotted for each of these tasks and for the time of flight.

Once a nuclear weapon has been detonated, a damage-assessment routine assigns damage to every unit within the radius of effects, both prompt and delayed. The model defines each unit as a circle or a number of circles, depending on the resolution desired. About six radii of effects are calculated for each unit hit by a nuclear weapon. The radii are based on various radiation dose levels, on blast effects, and on thermal effects, as applicable. Areas common to the circles that define units and to the annular regions that define various effects determine the fractional casualties.

III. CONCLUSIONS FROM USING THE MODEL

We aren't through with the model yet and still have some changes we want to make, but we can state some conclusions now. Most of our results are

probably already known to others, though perhaps in different form. One of the first things we noticed was the difficulty in finding out what had happened in a battle without graphic output. A computer program of this sort is seriously deficient without graphic output. So much is going on that the time required to check results with only printed output is excessive. The problem might be compared to trying to understand what has happened in a conventional battle when you know only the casualties and the coordinates of the individuals involved in the battle as functions of time. Without a map or similar aid to visualize what has happened, one would be lost.

In our calculation where we are trying to shoot at mobile offensive units as they are acquired by the defense during the attack, the most important parameter is time. From the time the defense first sees attacking units until those units have closed to within the minimum safe distance is a matter of minutes. The defense must either deliver nuclear fire on the attacking units during this time or fight the attacking units conventionally, in which case nuclear weapons would be limited in use to rear-area targets and fixed targets. The times we are talking about are somewhere between 5 and 15 minutes. About the only practical way our present-day weapon systems can respond with sufficient accuracy in this short time is if we fire them from presurveyed positions into presurveyed target areas. Presurveyed target areas and launch sites are hardly acceptable in a mobile situation, however, so many people reject the idea of using nuclear weapons against mobile maneuver elements. The only obvious way around this problem is to have direct-fire or terminally guided nuclear weapons. We should note that this application against mobile targets strongly influences weapon yield requirements as compared with something like tube artillery. This is because the velocity of a moving unit is difficult to estimate and, over a 5-minute period for instance, the error in target location can be substantial.

Firing of nuclear weapons from presurveyed positions involves another important time consideration. If we assume the nuclear fire unit is moved after each fire mission, the time required to emplace the weapon in its new position and make it ready to deliver accurate fire strongly affects the number of fire units required to cover a given area.

As mentioned earlier, about the only way the offense can use nuclear weapons is to shoot at areas of activity and hope that defensive units are in that area. If the defense puts part of its force in strong bunkers, the area fire option becomes expensive for the offense. For instance, against bunkers that will withstand (160-psi) loading, a (1-Mt) weapon will only cover about 1 km^2. The cost to assure neutralization of a large area containing bunkers is prohibitive.

Our study of unattended ground sensors used for target acquisition was interesting. With the sensors used as described earlier, we found that, against armored units, our target location errors were comparable to those for visual observation. Sensors decrease the certainty in locating any one part of a target, but, since they do indicate the whole target, they give a good indication of its centroid. The parameters that affect target location error are sensor range, average variation in range, spacing between sensors, and fraction of inoperative sensors.

IV. SUBROUTINES AND VARIABLES

A. Subroutines in the Model

In this section, we briefly explain the various computational subroutines used in the model, but make no effort to explain the output routines. Nor will we explain the output variables when we later cover the important variables. The programming to set up the output was simple, tedious, and not really relevant to this report. The routines below are grouped by task, but there is some overlap. Routines that make no calls to other subroutines are listed first in each group. The time-dependent routines presently run at the same time step but do not have to.

(1) Initialization

SUBROUTINE ROADNET

Given the variables DELEW, INS, DELNS, and IEW, this routine loads the variables ROADEW, ROADNS, NOEWROA, AND NONSROA, which essentially define the road network.

SUBROUTINE TOWNINI

Given the road network, this routine places on the roads those cities we use in our

population-avoidance calculation. The variables TOWNX, TOWNY, TOWNR, and NOTOWN define these cities. To simplify computation, we handle the cities as circles of various radii. For the area covered by the problem, we have a list of the various city radii and populations. Cities above the minimum population are put on our road network in a random fashion; the others are omitted from the computation.

SUBROUTINE INITIAL

This routine sets up a number of variables for the line-of-sight calculation. By using the ordering on the y coordinates that define the line segments approximating the ridge lines, this routine sets limits. These limits shorten the search needed to determine if two points have line of sight. The line-of-sight search is the most time-consuming part of the model.

SUBROUTINE DEPLOY

This input routine deploys the offense and the defense. The units must be placed in their desired locations, and their destinations must be defined. A number of variables are set in this routine.

BULL

This is the control routine. A number of input variables are defined in this routine, but no real calculations are made in it.

(2) Bookkeeping

LOGICAL FUNCTION ONROAD

Given the coordinates (x, y), ONROAD = .T. if the point is on a road; otherwise, ONROAD = .F.

SUBROUTINE VELCALC

Given the type of unit, whether it is day or night, whether the unit is on or off a road, and whether or not the unit is engaged in conventional combat, this subroutine calculates the maximum speed of the unit.

SUBROUTINE UPCOORD

For each time cycle, this routine updates a number of variables for both the offense and defense. All the visual target-acquisition variables are cleared in this routine, and the variables associated with unit movement are updated. For each time step while a unit is moving, a random variation of no greater than 20% is assigned to the unit's direction and speed. A number of movie variables are also updated, and the variable DAY is set in this routine. The routine uses VELCALC and ONROAD.

(3) Target Acquisition

We will first cover the visual part of target acquisition. The routines BINDATA and RETRDAP set up data for visual target acquisition. The routine LOSIGHT makes the line-of-sight calculation. The routines OBSERVE and OBCONUP decide whether one unit can see another if the two units have line of sight.

SUBROUTINE BINDATA

This routine divides the problem area into 1-km squares. The defensive units inside a 1-km square are stored together. The number of defensive units is a maximum of MAXNOEL - 1 in any one square. The number of storage locations used for each square equals MAXNOEL. The first word of the list tells how many units are in the square; the other words are for unit numbers. If no defensive unit moves during a cycle, this routine need not be called the next cycle.

SUBROUTINE RETRDAP

Given the unit number of an offensive unit this subroutine finds those defensive units within ILOS kilometers. The entry point INIRET first calculates four indices that define a rectangle of kilometer squares that must be searched. Each succeeding call to RETRDAP calculates the unit number of one defensive unit in the rectangle defined by INIRET until they are exhausted. At this point a zero is returned.

LOGICAL FUNCTION LOSIGHT

Given the unit numbers of an offensive and a defensive unit, this routine calculates whether or not there is a line segment approximating a ridge that blocks the line of sight between the two units. For most problems, the majority of computer time is spent in this routine.

SUBROUTINE OBSERVE

In this routine, each offensive unit is checked against the defensive units within ILOS kilometers to see if the units can observe one another. First, we determine the maximum distance over which a unit can be seen. We start by assuming that one dug-in individual can be seen at up to 200 m and set dis max = 200. If the unit is not dug in, we set dis max = 800. To account for the solid angle of the unit, we set dis max = dis max $\sqrt[3]{\text{unit strength}}$. We next set dis max = dis max * 2 if the unit has been observed or is moving.

We set dis max = dis max * 2.5 if the unit is in vehicles. Finally, we set dis max = dis max * 1.5 if the unit is involved in conventional combat. This part of the routine could perhaps be strengthened by taking into account the density of local ground cover.

We also calculate what fraction of the distance a given observing unit can see as compared with a unit in the same position under optimal circumstances. We start by taking into account the strength of the observing unit. Because of fatigue and variations in eyesight, we say one individual will be able, on the average, to see about 0.7 the distance that a unit of say 25 people could see. For larger units, there is little improvement. We therefore set frac = minimum (1.0, 0.7 + 0.06* $\sqrt{\text{unit strength}}$). Earlier we described how a moving unit cannot see as well as a stationary unit. To take this effect into account, we set frac = frac* [1 - 0.5 $(\frac{\text{unit speed}}{\text{maximum unit speed}})$]. We say a unit involved in conventional combat is going to be more absorbed than a unit on the move. Hence, we set frac = frac/3 if the unit is involved in conventional combat. Finally, we set frac = frac/2 if it is night.

Given dis max for a unit being observed and frac for a unit observing, we say that, if the product (dis max) (frac) is greater than the distance between the two units, the observing unit can see the other unit. Once it has been decided that unit A has been observed, data are stored to indicate the observation. Only the two opposition units closest to A will store such data, with one exception: if A is an offensive unit and the opposing side has an observer unit available, we will carry that observer unit, whatever its position, as one of the units observing A. Without an observer unit acquiring an offensive unit, no nuclear fire can be brought to bear on the unit.

We store an estimate for the centroid of unit A. Computation of the error in this estimate is described in our earlier section on target acquisition. An estimated velocity vector is stored for each acquired unit.

SUBROUTINE OBCONUP

For each defensive unit, this routine calculates the numbers for dis max and frac described in the OBSERVE writeup. These numbers are stored for use in OBSERVE.

Next, we cover those routines used to calculate target acquisition from sensors.

REAL FUNCTION RANGESN

Given the indices needed to define a specific sensor, this routine calculates the square of the range for the specified sensor. This number could be calculated once at the start of the problem and stored, but we calculate it each time it is needed to save storage. The range of a sensor has a random variation that is a function of the indices of the sensor. For this variation to remain constant throughout the problem, we compute a random number that is a function only of the indices needed to specify a sensor.

SUBROUTINE TXTYPL

Given the indices of an activated sensor monitored by a specified operator, this subroutine finds the other activated sensors monitored by the same operator that can be reached by following a continuous path of activated sensors. This gives the approximate area occupied by one unit. Once the sensors defining a single target have been found, the centroid is calculated. No information passes from one monitor to the next. The fire direction center must decide if two monitors are seeing the same unit.

REAL FUNCTION YYY

Given a specific row and a specific monitor, this routine calculates the y coordinate of the row of sensors.

SUBROUTINE CLOS

Given the coordinates of an armored vehicle, this routine, along with INDXRNG and RANGESN, calculates what sensors are activated and decides which operators monitoring the field, if any, might get a sensor reading.

SUBROUTINE INDXRNG

Given a point (x, y) and an operator number, this routine activates the sensors monitored by the specified operator within range of the point (x, y). The y coordinate is given relative to the start of the sensor field.

SUBROUTINE SENSINT

Given the output of the routine TXTYPL, this routine sets up the information needed to execute a nuclear fire mission.

(4) Conventional Action

SUBROUTINE CASCOBA

This routine calculates the casualties from conventional battles; it also calculates whether or not a unit is pinned down. For lack of a more precise measure, we say a maneuver unit's fighting ability is proportional to its manpower. The basic equations for determining conventional casualties and whether or not a unit is pinned down were described earlier.

SUBROUTINE ENDCOBA

This routine removes units from conventional battle when their casualties have made them ineffective, when they are sufficiently separated from enemy units involved in the battle, or when they have lost line of sight with enemy units involved in the battle. Each time a unit is removed from battle, the routine provides a printed output. The routine ends conventional battles whenever all the units on one side are removed from the battle. The routine calls ENDBAT, REPOSTO, REPOSTD, PUROFFU, and PURDEFU.

SUBROUTINE ENDBAT

Given the number of a battle that has ended, this routine restores the variables UNENOFF, UNENDEF, NOUNDFG, and NOUNOFE so the given battle is removed from the computation. This routine calls REPOSTO and REPOSTD and has the entry points PUROFFU and PURDEFU. Given a unit number, these entry points purge an offensive unit or a defensive unit from the calculation. The units are assumed to be ineffective at that point.

SUBROUTINE REPOSTO

This routine is called when an offensive unit involved in conventional battle breaks off the engagement but remains effective. The routine, which calls ONROAD, calculates the new posture for the unit. The routine has a second entry point, REPOSTD, which makes the same calculation for defensive units.

SUBROUTINE INICOND

This routine, along with FUENGAG, controls the conventional actions that defensive units take with respect to the offensive units acquired. These actions include starting an ambush, starting a conventional battle, withdrawing to a new position, and hiding in a position while waiting for the enemy to bypass. The choice of action depends on the mission of the defensive unit, the relative strengths of the defensive unit and the acquired offensive unit, the distance between them, and the direction the offensive unit is moving (toward or away from the defensive unit). These decisions must be programmed for each type of unit. Decisions on conventional action are presently made on a local basis without reference to the whole battle area, in line with our view that the defense will want to maneuver mostly on the basis of local conditions.

SUBROUTINE INICONO

This routine makes the same calculations for the offense that INICOND and FUENGAG make for the defense.

SUBROUTINE NEXTBAT

Any time a unit starts a conventional battle with another unit, this routine handles the bookkeeping. It also handles any ambushes. The ambush logic works as follows. Suppose unit A acquires unit B and unit B is not in a defensive position. Unit A can ambush unit B if unit A is not already involved in conventional combat and unit B has not acquired unit A. If unit A ambushes unit B during the day and the units are separated by 150 m or less, we say that half the individuals in unit A will each inflict a casualty on unit B in the opening burst of fire, before unit B can take cover. We assume here that unit B is at least comparable in size to unit A. If unit B is much smaller, it will be destroyed and pose no further problem. At 150 m, nearly everyone should be able to hit an unwarned target; but, for a number of reasons, we say about half the individuals in unit A will not be in a position to fire. If the ambush occurs at night, the equivalent effective distance must be divided by 2. If the ambush occurs at some distance $d > 150$ m, the casualties drop by the factor $(\frac{150}{d})^2$. If an ambush renders a unit ineffective for combat, NEXTBAT removes that unit from the calculation.

REAL FUNCTION DISTMIN

Given an offensive unit M acquired by a defensive unit I, this routine calculates an estimate of how close unit M will come to unit I, if unit M continues on its observed course.

SUBROUTINE FINPTCL

Given coordinates (x, y) and the number of an approximate ridge line, this routine calculates the distance from the point (x, y) to the closest point on the line segment representing the ridge line and calculates that closest point on the line segment. Finally, it calculates the closest point to point (x, y) that is 100 m from the line segment on the side away from point (x, y). This routine is used in conjunction with LOSSAFD to find cover for defensive units from attacking units. The routine also has the entry point FINMIND. Given a point (x, y) and the number of an approximate ridge line, this entry point calculates the shortest distance from the point (x, y) to the line extended from the given line segment.

SUBROUTINE LOSSAFD

Given the locations of an offensive unit and a defensive unit, this routine finds the closest approximate ridge line to the defensive unit that blocks line of sight to the offensive unit. This routine uses FINPTCL.

SUBROUTINE MOVEDEF

This routine sets up the path a defensive unit is to take when it moves for some reason. For instance, each time a nuclear fire unit is given a fire mission, it moves to a new location. This move is made in a random manner. Thus, if the unit is on a north-south road, the decision to go north or south is made by a random number generator. When the unit comes to an intersection, the direction it takes is again determined by a random number generator. Finally, the distance it moves along the chosen route is also determined in a random manner. The above choices are all subject to restrictions that the unit must stay inside the problem area and within range of the defensive line.

(5) Nuclear Fire

REAL FUNCTION PSIRAD

Given a yield and a pressure in psi, this routine calculates the distance at which the horizontal component of peak pressure matches the specified pressure. The routine merely does a linear interpolation for a 1-kt weapon by using data given in Ref. 3. The result is then multiplied by the cube root of the yield.

REAL FUNCTION FISRAD

Given a yield and a radiation dose, this routine calculates the maximum distance at which a fission device will provide the specified dose. The routine assumes the dose at a fixed distance is linear with yield. By dividing the input dose by the input yield, we get a scaled dose (dose per kiloton). By using 1-kt data, we then calculate the distance at which we will get the scaled dose with our input yield. This last calculation is simply a logarithmic interpolation from a series of data points.

The routine also has an entry point, FISDOS, that calculates the radiation dose as a function of yield and distance to the zero point. The calculation is the same as above except that the interpolation is inverted.

REAL FUNCTION THERMDI

Given a yield, this routine calculates the distance at which individuals would get first- or second-degree burns if skies were clear. The routine uses data from Ref. 3.

REAL FUNCTION OVERTRN

Given the yield, this routine calculates the distance at which weapon blast will overturn an armored vehicle. The data used here came from Ref. 3 and are assumed to be the same as for earth-moving equipment.

REAL FUNCTION CIRCCAL

Given two circles C_1 and C_2 with radii R_1 and R_2 and the distance d from center to center, this routine calculates the fractional area of C_1 contained in C_2.

SUBROUTINE FRACCOV

Given the radii of effects against a specified unit for the various classes of radiation doses and for blast damage and thermal damage, given the distance from the zero point to the specified unit, and given the radius of the unit, this routine computes the fractions of prompt casualties and of various classes of delayed casualties. Casualties are assessed on an area basis. This routine uses CIRCCAL.

REAL FUNCTION EFFECTC

This routine calculates radii of effects of nuclear weapons for different types of units in differing postures. It provides the logic to

account for a unit's hardness before it calls other routines to make the desired calculation. This routine uses PSIRAD, FISRAD, THERMDI, and OVERTRN.

SUBROUTINE DELAYCA

For each offensive and defensive unit, this routine calculates any delayed casualties. The casualties in each class are assessed linearly in time. For instance, if ten class 2 casualties are to become incapacitated in ten days, this routine will assess one casualty per day for ten days. If at any time the unit becomes ineffective because of delayed casualties, this routine purges the unit from the calculation. In the problems we have run so far, delayed casualties have not had an important effect.

SUBROUTINE NUCIMPT

Given a nuclear shot, a predicted zero point, and a CEP, this routine computes an actual zero point and assigns casualties to any units that receive prompt casualties from the weapon. If the nuclear shot has terminal guidance, the routine computes a final predicted zero point. This routine purges from the calculation any units whose prompt nuclear casualties are sufficient to render the unit ineffective. The routine computes the radiation dose at the center of each unit to keep track of approximate cumulative dose. The routine sets up the data for delayed casualties. Finally, the routine sets up a number of variables for output prints and plots. The routine uses EFFECTC, FRACCOV, FISDOS, PUROFFU, PURDEFU, THERMDI, FISRAD, and PSIRAD.

REAL FUNCTION FRNDSFO

Given a yield, a predicted zero point, and a specified offensive unit, one can calculate the unit's distance from the zero point minus the minimum safe distance. This routine calculates the smallest such number among all offensive units. If the number is negative, the shot would endanger offensive units. This routine has the entry point FRNDSFD, which makes the same calculation for defensive units. Finally, the routine has the entry point CITYSAF, which, given a yield, will calculate a minimum safe distance for cities. Given next a zero point, this routine calculates the minimum distance, among all cities, to the circle defined by minimum safe distance from the zero point.
In each of the above calculations, one computes the distance that a zero point can be moved before a safety criterion is violated.

INTEGER FUNCTION FINFUND

Given the coordinates (x,y), this routine finds a nuclear fire unit that is within range of the point and ready to fire. Starting from a random point, the routine searches the list of fire units. If a unit is not available to fire at the target, a zero is returned for the unit number.

SUBROUTINE FRMSNDF

This routine, along with FINFUND, FRNDSFD, and CITYSAF, is the model analog of the fire direction center. After the appropriate time delay following target acquisition, this routine looks at all fire-mission requests, checks for troop and city safety, checks to make sure that no two observers are requesting fire on the same target, assigns an available nuclear fire unit (if one exists within range of the target), and sets up the time delays appropriate to carrying out the fire mission.

This routine also sets up the fire missions for execution in conjunction with unattended ground sensors. The safety checks are not needed since they are implicit in the location of the sensors.

B. Variables in the Model

When we were programming the model, we decided to store information as it was generated and use it at a convenient time in the calculation. As a result, many variables carry information from one routine to the next, when the calculation could have been done in one step. This approach makes the programming easier to prepare and much easier to change but uses more storage.

When a variable refers to, say, a defensive unit, there is usually a corresponding variable for the offensive unit. Rather than list both, we list only the defensive variables. The unit numbers given below refer to logical units inside the computer model. One military unit may have more than one logical unit. For instance, each tank of a tank company may be a logical unit, or the whole company may be one logical unit, depending on the resolution desired.

XDEF(I) is the x coordinate of the I*th* defensive unit.

YDEF(I) is the y coordinate of the I*th* defensive unit.

NOUNITD is the number of defensive units at the

start of the problem.

VDEF(I) is the speed of the I*th* defensive unit.

[UXDEF(I), UYDEF(I)] is the unit velocity vector for the I*th* defensive unit.

The pairs of variables [X1DEF(I), Y1DEF(I)], [X2DEF(I), Y2DEF(I)], and [X3DEF(I), Y3DEF(I)] define paths of travel for the I*th* defensive unit. The unit is first to go to point [X1DEF(I), Y1DEF(I)], then to [X2DEF(I), Y2DEF(I)], and finally to [X3DEF(I), Y3DEF(I)]. These variables are used if a unit has multiple objectives or are used to define alternative paths of travel if the planned path of travel is blocked. Pairs of zeros are used to indicate no path is defined.

UNSTRID(I) is the strength of defensive unit I at the start of the problem. UNSTRD(I) is the strength of defensive unit I at later times. Whenever the fraction UNSTRD(I)/UNSTRID(I) goes below a certain level, the unit is defeated.

RADDEF(I) is the radius of defensive unit I. Each unit is logically thought of as a circle.

TYPEUND(I) is an integer defining the function of defensive unit I. At present, TYPEUND = 1 is a nuclear fire unit, TYPEUND = 2 is a forward observer unit, TYPEUND = 3 is an infantry unit assigned to protect a forward observer unit, TYPEUND = 4 is a local militia unit, TYPEUND = 11 is a mechanized infantry unit, and TYPEUND = 12 is a tank unit. The variable is used to assess nuclear casualties, to determine speed, and to determine what conventional action a unit takes under given circumstances.

POSTURD(I) is an integer defining the posture of the I*th* defensive unit. POSTURD = - 1 means the unit is ineffective, POSTURD = 0 means the unit is stopped and dug in, POSTURD = 1 means the unit is moving cross-country at full speed, POSTURD = 2 means the unit is moving down a road, POSTURD = 3 means the unit is moving cross-country slowly with maximum use of covers, POSTURD = 4 means the unit is stopped and in conventional combat, POSTURD = 5 means the unit is moving and in conventional combat, and POSTURD = 6 means the unit is dug in and in conventional combat. POSTURD and TYPEUND have essentially unlimited possibilities for definitions.

TMPODEF(I) is the length of time that defensive unit I has been in its present posture.

The model carries up to three different classes of delayed casualties, each with a different time to incapacitation. S1DEF(I), S2DEF(I), and S3DEF(I) are the numbers of casualties for the I*th* defensive unit in each of the three classes.

T1DEF(I), T2DEF(I), and T3DEF(I) are the times to incapacitation for the three corresponding classes of casualties.

UN1OBSD(I) is the unit number of the closest offensive unit that has defensive unit I visually acquired at the present time. UN2OBSD(I) is the unit number of the second-closest offensive unit that has defensive unit I visually acquired at the present time.

TMCOOBD(I) is the length of time that the offense has continuously observed defensive unit I visually.

TMLAOBD(I) is the last time defensive unit I was visually observed.

[XEDEF(I), YEDEF(I)] is the set of (x, y) coordinates for the center of defensive unit I as estimated by the offense the last time the unit was visually observed.

[XTDEF(I), YTDEF(I)] is the velocity vector of defensive unit I as estimated by the offense.

CONOBSD(I) tells how far defensive unit I is visible under optimal circumstances. It corresponds to dis max in the previous section on OBSERVE.

CONDOBS(I) tells the fraction of the distance that defensive unit I can see as compared with a unit in the same position under optimal circumstances. This corresponds to frac in the previous section on OBSERVE.

BSTHDDF(I) is the blast pressure in psi needed to destroy defensive unit I.

RADHDDF(I) is the radiation transmission factor for defensive unit I. This factor can vary with the posture of the unit.

UN1BOBD(I) is the unit number of the closest offensive unit that defensive unit I is observing.

D1BOBD(I) is the estimated distance between offensive unit UN1BOBD(I) and defensive unit I.

D2BOBD(I) and UN2BOBD(I) carry the same distance information for the second closest unit.

DMAXDEF(I) and DMINDEF(I) are two variables used in the OBSERVE subroutine that temporarily carry the distances D1BOBD(I) and D2BOBD(I).

CEPDEF is the CEP of the defensive nuclear fire units if they are not range-dependent and use only one type of fire unit.

INUCDEF(I) is the number of nuclear weapons expended

by defensive unit I. This variable is important only for nuclear-capable units.

RADLVLD(I) is the radiation dose received at the center of defensive unit I. This variable is used mostly to monitor friendly casualties.

TMTOMCD(I) is used with forward observer units that provide terminal guidance to nuclear fire. This variable tells how long it will be before the equipment used by defensive unit I for terminal guidance is free for another mission.

EXCOEF corresponds to the c we used back in Eq. (1).

TMTODC3 is an input variable. It tells the maximum time that delayed casualties in class 3 will live.

TMTODC4 carries the same information as TMTODC3, except it is for class 4.

TMTODR is an input variable that tells the maximum time to death for persons receiving a fatal radiation dose.

TIMEINT is the time of day in minutes that the problem started; 6 a.m. is assumed to be zero.

TIME contains the problem time in minutes.

DELT is the time step between cycles and may vary during the problem.

DAY is a logical variable indicating day or night. DAY = .T. implies daytime and DAY = .F. implies nightime.

At any given time, there may be up to 20 nuclear shots being fired. Between the time the request for nuclear fire goes to the fire unit and the time the shot detonates, information on the shot is carried in the following variables. The ordering on the index J merely indicates the order in which fire-mission requests have gone to the fire units. It has no importance.

NOSTRIK is the number of nuclear shots being executed.

YIELD(J) is the yield of the *Jth* shot.

UNFRDAT(J) is the unit number of the target of the *Jth* shot. The number is negative if the target is a defensive unit and positive if the target is an offensive unit. The number is carried so that, if the shot has terminal guidance, the target data can be updated.

[PREDX(J), PREDY(J)] is the predicted ground zero for the *Jth* shot.

TRANMAX(J) is used with terminal guidance as a safety check. Once a shot has been requested and a predicted ground zero has been specified for the *Jth* shot, the terminal guidance can divert the shot from the predicted ground zero by, at most, the distance TRANMAX(J).

TMTOIMP(J) is the time to zero point for the *Jth* shot.

CEPNUC(J) is the CEP for the *Jth* shot.

The following input variables relate to nuclear missions fired by the defense.

TREXFRM is the time required for a nuclear fire unit to execute a mission after it receives a fire-mission request. This time assumes the unit is ready. With an artillery piece, for instance, this would be the time required to load the round, load the powder charge, point the tube, and pull the lanyard.

YIELDDF(J) for J = 1, 2, 3, 4, 5 is the set of yields available for the defensive units. For the problems run so far, only one yield has been used.

TREBGNM is the time a defensive nuclear fire unit requires to get ready to deliver accurate fire after it arrives at a new presurveyed position. For an artillery piece, this would be the time required to dig in the spade and lay the gun.

RANGE is the maximum range for a defensive nuclear fire unit.

RANGEMN is the minimum range for a defensive nuclear fire unit.

TOF is the time of flight at maximum range for a defensive nuclear fire unit.

TMCOMMD is the time required for the defensive FDC to receive a fire mission, make any safety checks, decide whether or not the mission should be fired, and forward the request to an available nuclear fire unit.

SMALOUN is the smallest visually acquired offense unit that will be a nuclear target for the defense. In reality, the defense would have to guess at the size of a unit it acquired. For sensor-acquired offensive units, the defense can only look at the number of sensors that signal an intrusion.

PSIMAXD is the maximum blast overpressure used in specifying troop safety requirements for the defense.

RADMAXD is the prompt radiation dose used in specifying troop safety requirements for the defense.

RADEMGD is the prompt radiation dose used in specifying troop safety requirements for the

defense in emergency situations.

PSIMAX is the maximum blast pressure allowed on a safe city. RADMAX is the maximum prompt radiation dose allowed on a safe city.

The following variables are used to describe the road network.

NONSROA is the number of north-south roads in the problem.

ROADNS(J) is the x coordinate of the *Jth* north-south road.

NOEWROA is the number of east-west roads in the problem.

ROADEW(J) is the y coordinate of the *Jth* east-west road.

DELNS is an input specifying the spacing between north-south roads.

DELEW is an input specifying the spacing between east-west roads.

The following variables are used in the conventional-combat part of the mode.

MAXNOBA is the maximum number of conventional battles allowed to go on at any one time (a storage limitation).

NOBATTL is the number of battles in progress.

NOUNDFE(J) is the number of defensive units engaged in the *Jth* battle. The battles are numbered sequentially in the same order as they start. Any battle that is stopped is removed from the list. Note: $0 \leq J \leq$ NOBATTL.

UNENDEF(J, K) is the unit number of the *Kth* defensive unit involved in the *Jth* battle. Note: $0 \leq K \leq$ NOUNDFE(J). Any unit no longer in the battle is removed from the list.

MAXELEN is the maximum number of elements on either the defense or the offense that are allowed to become engaged in one conventional battle (a storage-size limitation).

STRMIND is the fractional level of strength (an input variable) at which a defensive unit becomes ineffective. In most real situations the mental attitude of a unit is more important that its physical situation. Unfortunately, we know of no way to model the mental attitude.

The following input variables are used in assessing damage from nuclear weapons.

The model uses four different prompt radiation doses: C2DOSE, C3DOSE, C4DOSE, and LETHDOS. We presently use C2DOSE as the minimum dose needed for immediate casualties. The other three levels of radiation dose are used for delayed casualties.

EARBRST is the amount of blast pressure in psi needed to rupture eardrums. We assume this blast pressure to cause instantaneous casualties among personnel who are exposed or in foxholes.

TRFCAPC is the transmission factor for an APC.

TRFCTK is the transmission factor for tanks.

TRFCFOX is the transmission factor for foxholes.

The following miscellaneous variables are related to problem size.

ILOS is the maximum distance in kilometers that a person can see.

MAXNOEL is the maximum number of defensive units allowed in a square kilometer (a storage limitation).

INS is the distance in kilometers that the problem covers in the y direction.

IEW is the distance in kilometers that the problem covers in the x direction.

CONDISS is the square of the maximum distance at which effective conventional combat can occur.

The following variables are used in computing targets on the basis of outputs from unattended ground sensors.

SENSOR (I, J, K) is a logical variable. In our sensor field, we say one operator monitors the sensors on 1 km of front. For each sensor in the field, there is at least one set of indices I, J, and K for the variable SENSOR. The index K tells what operator is monitoring the sensor, and the indices I and J tell the location of the sensor inside that part of the field monitored by operator K.

SENSOR (I, J, K) = .T. implies that the specified sensor is indicating an intrusion.

SENSOR (I, J, K) = .F. implies that the specific sensor is not indicating an intrusion. If a specific sensor is not functional, it is always false.

Note: the sensors on a boundary between two operators are monitored by both operators.

CONSENS (I, J) is a logic variable used to calculate how many tripped sensors in a field monitored by one operator are neighbors of one another. A group of tripped sensors is used to indicate the approximate outline of one unit.

TS1(K) is a logic variable.

TS1(K) = .F. indicates monitor K has no targets acquired by the sensor field.

TS1(K) = .T. implies monitor K has a target and

the coordinates are stored in the variables [TX1(K), TY1(K)]. These coordinates are merely the centroid of a connected group of sensors that have been tripped. The model can store three separate targets for each operator.

The other variables used are TS2(K), TX2(K), TY2(K), TS3(K), TX3(K), and TY3(K).

IROW is the number of rows of sensors. In Fig. 1, IROW = 7.

JCOL is the number of columns of sensors monitored by one operator. In Fig. 1 JCOL = 5.

DXSENS is the x distance between sensors.

DYSENS is the y distance between sensors.

SENSRNG is the average range (an input variable) at which a sensor will pick up an armored vehicle. Each sensor actually has a range that is chosen at random between SENSRNG - 1/2 VARI and SENSRNG + 1/2 VARI. VARI is another input variable indicating the variation.

The following input variables are used to approximate the geography of the area of interest.

POPMIN is the minimum population needed to establish a population center as a safe city. By a safe city, we mean an area in which the defense will not use nuclear weapons in a way that causes more than, say, light damage. The cities are assumed to be circular.

NOTOWN is the number of safe cities in the defensive area.

[TOWNX(I), TOWNY(I)] is the set of coordinates for the center of the *Ith* safe city.

TOWNR(I) is the radius of the *Ith* safe city.

NOLOSBL is the number of important ridge lines in the problem area. At present, NOLOSBL = 37.

[X1LOS(I), Y1LOS(I)], and [X2LOS(L), Y2LOS(I)] are the end points of the line segments used to approximate the ridge lines.

REFERENCES

1. Staff Officer's Field Manual Organization, Technical and Logistical Data, Department of the Army Field Manual, FM 101-10-1 (July 1971), pp. 2-11.

2. Surveillance, Target Acquisition and Night Observation (STANO) Operations, Department of the Army Field Manual, FM 31-100 (May 1971), pp. B118-B119.

3. Samuel Glasstone, The Effects of Nuclear Weapons, United States Atomic Energy Commission, April 1962, pp. 137-139, p. 573, pp. 174-175.