Non-Monotonicity, Chaos, and Combat Models

J. A. Dewar, J. J. Gillogly, M. L. Juncosa

RAND

This research was supported by RAND using its own research funds.

ISBN: 0-8330-1140-5

Published 1991 by RAND
1700 Main Street, P.O. Box 2138, Santa Monica, CA 90407-2138

R-3995-RC

Non-Monotonicity, Chaos, and Combat Models

J. A. Dewar, J. J. Gillogly, M. L. Juncosa

RAND

PREFACE

This research was inspired by general readings about mathematical chaos and the recognition that certain classes of combat models behave much like forced-damped systems in which chaos can be found. The authors discovered chaotic-like behavior in the outcomes of a very simple computerized combat model. Recognizing that behavior of this type was relatively common to combat models and that there was nothing in the combat modeling literature that tried to connect such behavior to chaos, the authors presented their preliminary results to the RAND analytic community.

The report attempts to describe, in a slightly different way than usual, what goes on in combat models, thus offering promise for a better understanding of why computerized combat models sometimes produce nonintuitive results. It also holds promise for developing an understanding of how to improve the behavior of combat models.

This study was supported by RAND, using its own funds. It builds on preliminary research conducted in RAND's three federally funded research and development centers: Project AIR FORCE, the National Defense Research Institute, and the Arroyo Center. The report should be of interest to those responsible for the development of combat models throughout the armed services, as well as to those who use the results of such models to test doctrines, tactics, and strategies, or to make policy.

SUMMARY

Computer modeling of combat has become an important element in the analysis of military strategy, tactics, policy, and training. Some of these models are rigorously testable and validatable. However, many combat models (particularly those that attempt to simulate human decisionmaking processes) fall into the general set of ad hoc models whose rigorous validation, because of their size and complexity, is impossible and whose underlying dynamics are not well understood. These models are generally "validated" against well-honed human experience and intuition, yet many subsequently produce the occasional disturbing nonintuitive result.

Our task here is to examine the potential causes of these occasional nonintuitive results. Often these nonintuitive results take the form of non-monotonicities in which a capability added to the side of one combatant leads to a less-favorable result for that side. In models used to compare alternatives, this type of non-monotonic behavior is particularly undesirable.

On careful inspection, after the fact, non-monotonic behavior is usually understandable. Often it is the result of a coding error, which is then easily fixed. Sometimes it is the result of numerical or dynamical problems such as roundoff, time-step granularity, or delayed feedback. Problems of this type are well known in the literature and there are a variety of methods for dealing with them.

At times, however, non-monotonic behavior is caused by a modeled decisionmaking process and it is this behavior that interests us. Decisions based on the state of the forces in the battle introduce nonlinearities into a model and if that model has both attrition (damping) and reinforcements (forcing) this can lead to chaos. Chaos is the name given to a new branch of the theory of nonlinear dynamical systems that holds promise for furthering our understanding of dynamical systems. Its most striking characteristic is that chaotic systems are unpredictable even though they are deterministic, and this characteristic is found even in some very simple deterministic dynamical systems.

If combat models with nonlinear decisionmaking processes are chaotic, then nonintuitive behavior may be endemic to the model rather than occasional and anomalous. Further, since the appearance of nonintuitive behavior is unpredictable, eliminating it from the model may require drastic measures.

Chaos in combat models has not been widely reported in the literature both because of the relatively recent development of the tools of mathematical chaos and because combat models tend to be very large and complex, making them difficult to investigate over a broad range of outputs. Our work concentrated on an investigation of chaos in a very simple combat model.

We have shown that in a simple combat model containing reinforcement decisions based on the state of the battle, the resultant nonlinearities lead to chaotic behavior in the underlying dynamics of the model. This is one piece of the puzzle.

The other piece has to do with non-monotonic behavior in the model's outcomes. In our simple combat model we "corrected" for all known potential causes of non-monotonicities including precision/roundoff, time-step granularity, and delayed reinforcement feedback. Nonetheless, non-monotonicities remained in the model's outcomes. Putting these pieces

together, we have shown that these remaining non-monotonicities are the result of chaos in the model's underlying behavior.

An investigation of several dimensions of the input parameter space revealed that the non-monotonicities associated with chaos are widespread in some regions and absent in others. This suggests that if such regions could be accurately mapped, they might be avoided in running the model (unless they are the tip of a more generally chaotic iceberg). On the other hand, if the reinforcement decision is based on time only (referred to as "scripting" the reinforcements), the nonlinearities and chaotic behavior are eliminated from this source, as are the non-monotonicities. The primary disadvantage in scripting the reinforcements is, of course, the reduction in verisimilitude in the model, but in instances where this reduction is acceptable, scripting does eliminate the chaotic non-monotonicities.

In our simple model, then, we have shown that the nonlinear functions of the decisionmaking processes can lead to chaos in the dynamics of the model and non-monotonicities in its outcomes. Also, we can show that adding another decision based on some state of the battle can worsen any observed non-monotonicities, and that the area in which these non-monotonicities can occur does not necessarily shrink when another such decision is added. Although these results are very preliminary, they suggest that larger models are not immune to the behavior we have seen in our simple model.

In conclusion, **in any combat model whose utility depends on monotonic behavior in its outcomes, modeling combat decisions based on the state of the battle must be done very carefully. Such modeled decisions can lead to non-monotonic and chaotic behavior and the only known ways of dealing with that behavior are either to remove the modeled decisions (decreasing the verisimilitude of the model) or to validate that the model is monotonic in the region of interest.**

That conclusion is carefully worded not to overstate the strict implications of the research that was performed. In so doing, however, it understates two other implications of this line of reasoning. In the first case, **this research raises the question of whether the conclusions and applications of past modeling might have been tainted by the unsuspected presence of mathematical chaos and it casts a shadow over the claimed reliability of combat (and other) models even when they do not (or have not been proved to) exhibit blatantly non-monotonic behavior.** Second, historic battles have been known to hinge on very subtle effects of decisionmaking and have been described as "chaotic." This research holds out the promise that mathematical chaos and the chaos of war might be related. As a strong caveat here, it is too easy to presume that they are necessarily connected. Whether or not the behavior in our simple model is akin to behavior in a real battle is an interesting question but one that requires serious thought and research. It is one thing to say that both modeled decisions and real battles produce chaotic behavior and unexpected reversals in fortune, but quite another to say that the one faithfully models the other. The latter is a matter for military science and is amenable to both analytic investigation and historical research.

Because of the preliminary yet important nature of this work we recommend further research in this area. The specific areas of greatest payoff (in order of importance) are:

- Chaotic behavior in larger, working combat models.
- The relationship between chaos and the stopping conditions in a finite combat model.

- The possibility that chaotic underlying dynamics exacerbate non-monotonicities from other sources.
- Mathematical research on the definition and testable criteria of chaos for piecewise continuous maps.

ACKNOWLEDGMENTS

In its infancy, this work was a fatherless child. Carl Builder fought for it and protected it through its difficult birth. Dean Wilkening and James Thomson nurtured it with fiscal support and encouragement through its early growth. Since then, it has benefited greatly from the collegial support of Julian Palmore of the University of Illinois, Vladimir Protopopescu of Oak Ridge National Laboratory, Gottfried Mayer-Kress and Gary Doolen of the Center for Nonlinear Studies at Los Alamos National Laboratory, Stavros Busenberg of Harvey Mudd College, and RAND colleagues too numerous to mention. The final product has improved immeasurably thanks to the careful and insightful reviews of Richard Hillestad and Michael Miller. The final typography would have suffered brutally without the care and attention of Laurie Rennie.

CONTENTS

Section

Appendix

FIGURES

TABLES

I. INTRODUCTION

COMBAT MODELS AND NON-MONOTONICITIES

Computerized combat models are common tools in the analysis of military strategy, tactics, policy, and training (Hughes, 1989). They are widely used to explore alternative force structures and force employment schemes and are a vital part of the process of choosing which weapon systems to develop and purchase. They belong to a general set of ad hoc models whose underlying dynamics are typically nonlinear and have not been subjected to rigorous analysis and testing under controlled conditions. To be sure, some portions of combat models (e.g., projectile trajectories, firing rates, and mean-times-between-failure) may be modeled accurately and tested rigorously under controlled conditions. However, other parts (e.g., "leadership" and the "fog of war") are beyond our ability to represent and test analytically.

The typical model simulates combat between opposing forces at some level of abstraction. No combat model is seriously expected to be absolutely predictive of actual combat outcomes. It is common, however, to expect models to be relatively predictive. That is, if a capability is added to one side and the battle is refought, the difference in battle outcomes is expected to reflect the contribution of the added capability.

Models used for comparative purposes typically carry the implicit assumption that combat is what we will call (borrowing the mathematical term) *monotonic*, in that adding more capabilities to (only) one side will lead to at least as favorable a combat outcome for that side.[1] The model's outcomes are then interpreted in light of that implicit assumption and are generally considered suspect if they exhibit non-monotonicities. Although non-monotonic behavior is not uncommon in combat models, modelers treat it as anomalous and either "fix" the model until the non-monotonicity disappears or ignore the non-monotonic outcomes. By their nature, however, combat models tend to be large and complex and can take several hours of computer time to produce the results of a simulated battle. Because of this, the "fixes" tend to be for a specific observed non-monotonicity, and little is done to explore the model in general for non-monotonicities.

On the other hand, non-monotonicities are known to be caused by a wide variety of mechanisms. Many combat models, for example, contain stochastic variables, and random variations from battle to battle can clearly cause non-monotonicities to creep in. Even in deterministic models with no random components or in one whose random variables are replaced by their mean values, non-monotonicities can arise from several well-known causes. Some of these causes are described in Sec. II. If properly dealt with, these causes and their non-monotonicities can typically be eliminated. In this report, we investigate a different source of non-monotonicities—one whose elimination is more problematic and which is related to mathematical chaos.

[1] In fairness, not all models used for comparative purposes are expected to predict comparative advantages. The best uses of such models are to provide insight into complex relationships and effects that occur in battle. Many such models, however, *are* used to argue the relative merits of competing systems, plans, strategy, etc., and it is primarily to those uses that this work is addressed.

CHAOS

The advent of video-display microcomputers has greatly increased the visibility and understanding of a class of physical and mathematical processes identified with chaotic behavior or chaos.[2] Chaos has now been recognized and investigated in a wide variety of disciplines including weather forecasting (Lorenz, 1963; Palmer, 1989; Pool, 1989e), chemical reaction kinetics (Rehmus and Ross, 1985; Scott, 1989), population dynamics (May, 1976, 1989), planetary orbits (Murray, 1989), the arms race (Saperstein, 1984; Grossmann and Mayer-Kress, 1989), epidemiology (Pool, 1989c), the oscillations of atomic particles (Hoffnagle et al., 1988), economic prices (Jensen and Urban, 1984; Nash and McBride, 1988), and neural networks (Derrida and Meir, 1988).

Although no simple, universally accepted definition of chaotic behavior exists, chaos is characterized by unpredictable, apparently random behavior over long periods and extreme sensitivity to perturbations in current or initial conditions. Chaotic behavior is not a product of random impulses but is implicit in the deterministic equations modeling the process and can be observed in the behavior of the process. The significance of chaos in mathematical simulations (and even in some of the physical processes being simulated) is that the outcomes do not settle out to some steady state or even a predictable cycle; they must be fully calculated through all of their iterations; and they are so sensitive to small variations in the initial conditions as to make each simulation a unique path with little or no long-term relationship to its neighbor currently only slightly removed.

The observation of chaotic behavior in atmospheric simulations by Lorenz in 1963 has now been generally accepted as foreclosing the possibility of weather predictions beyond 10 days to two weeks that might be attempted through larger, more detailed simulations supported by finer, more precise instrumentation nets and larger, faster computers. It is now apparent that chaotic behavior, both in the actual atmospheric processes and in the simulations, implies that the slightest disturbance (e.g., from the fluttering of the wings of a butterfly in Beijing) could propagate in unpredictable ways to alter the weather a continent and a month away. Thus, the long-standing search for bigger and better simulations, computers, or input data for long-range weather prediction has been effectively terminated in the new understandings of chaotic behavior and its implications.

The possibility that chaotic behavior might be lurking in other simulations is an obvious inference. Combat simulations, particularly those for ground combat, seem to be candidates for chaotic behavior because they often involve nonlinear equations that are iterated many times over the course of a battle or war. Also, with reinforcements, combat simulations contain both forcing and damping behavior common to chaotic systems. If they do exhibit chaotic behavior, there is the possibility that some aspects of the simulation (and perhaps of the war being simulated) are simply not predictable or are extremely sensitive to the initial conditions or intervening events.

CHAOS AND COMBAT MODELS

Chaos shows up in a variety of forced-damped dynamical systems; combat models are, at their core, basically forced-damped dynamical systems[3]; and combat models have shown

[2] For good, readable introductions to chaos see Gleick (1987) or Stewart (1989b).

[3] A dynamical system can be thought of as a process whose state can be described mathematically as a function of time. Damping in a dynamical system will drive the system toward a steady-state condition and forcing will drive it away from a steady state. The attrition in a combat model damps it and the reinforcements force it.

non-monotonicities indicative of a sensitivity to small changes. These facts suggest that non-monotonicities occasionally seen in combat models might be related to chaos. If they are, then they may be more widespread than generally believed, and they may represent *inherent* rather than anomalous behavior in such models.

Some work has been done on relating chaos and combat models. Some analysts, for example, believe that they have observed chaotic behavior in large combat simulations, specifically in VIC (Sandmeyer, 1988) and RSAS.[4] Proving that the observed phenomena are indeed evidence of chaos rather than simple sensitivity, noise from rounding errors, or some other cause is very difficult. Because these simulations are very complex, they are not transparent; because of the cost to run them, they are not easily mapped over a wide range of conditions.

P. Rydell and R. Stanton[5] have looked for chaos in a simpler model. Their interest, however, was the general stability of the model and not the chaotic attractors per se. As such, their work is suggestive of chaos in simple combat models but it does not delve rigorously into its mathematical aspects.

Ongoing work at Oak Ridge National Laboratory is trying to model combat through partial differential equations (Protopopescu et al., 1989). In the course of that work, researchers have studied chaos in combat models of that type. In a related field, Saperstein, Grossmann, and Mayer-Kress have written on chaos and the arms race (Saperstein, 1984; Grossmann and Mayer-Kress, 1989).

In general, there are several challenges in investigating chaos and combat models. First, there are several definitions of chaos to be found in the literature. The definition we chose for this report is detailed in Sec. III. Second, chaos is a long-term behavior of dynamical systems and typical combat models are run for a relatively small number of time steps. Behavior that *appears* chaotic in a given combat model run requires careful analysis of the underlying equations to demonstrate that they satisfy the requirements for mathematical chaos. Finally, if chaos is present in the underlying dynamical system, one needs to make clear the relationship between that chaos and misbehavior in (finite) realizations of the system that have stopping conditions.

In this report we investigate a specific type of chaos: one caused by nonlinearities in the model resulting from modeled decisions that are functions of the state of the battle. Section II introduces the simple combat model we used in our investigations and the concept of non-monotonicities. It also deals with some of the common causes of such non-monotonicities and discusses how we eliminated them from our simple model to concentrate on non-monotonicities caused by the chaos we were investigating. Section III discusses what we mean, specifically, by chaos in this context and what is causing it. We are particularly interested in chaos caused by nonlinearities related to a commander's decisionmaking processes and its relationship to non-monotonicities in the outcomes of finite battles. Section IV discusses how to deal with non-monotonicities of this type.

Throughout the first four sections, we concentrate on a very simple combat model. Section V discusses what that simple model can tell us about the behavior of larger, more complex combat models and suggests why our discussions should be pertinent to any combat model that contains decisions based on the emerging state of the battle. The final section summarizes our views on chaos and non-monotonicities of this type and discusses important areas of further research.

[4] Private communication Patrick Allen, RAND, November 1988.

[5] Private communication, RAND, October 1988.

II. NON-MONOTONICITY IN A SIMPLE COMBAT MODEL

Monotonicity in combat models has not been seriously investigated primarily because typical combat models are large and complex and require substantial computer time to run individual battles. To investigate monotonicity, we created a very simple combat model that exhibits at least some of the major features of typical combat models. That such a model can be generated and will exhibit non-monotonicities is not a great surprise; such non-monotonicities are not uncommon in combat modeling. They can occur for a variety of reasons including computational roundoff and precision problems and sensitivities to time-step size. In our simple model we worked to eliminate all such "artificial" causes of non-monotonicities. Perhaps what is surprising is that after the model has been scrubbed for such artificial sources of non-monotonicities, there remain widespread non-monotonicities in our model that would be largely unobservable in a larger, much slower-running model. But that is the concern of later sections. In this section we introduce the simple model we used in our investigations and explore some classical sources of non-monotonicities in such a model.

A SIMPLE COMBAT MODEL WITH NON-MONOTONICITIES

An exemplar of our simple model is shown in Table 1.[1] It describes the parameters for a simulated battle between Blue and Red "troops," with attrition characterized by fixed, linear attrition rates.[2] This model is at the operational level of warfare in that it simulates the human decision processes of when to call for reinforcements and when to withdraw from the battle. It also simulates the potential for delays between when the reinforcements are requested and when they arrive.

B_n and R_n represent troop strengths of Blue and Red at time n. For each battle, each side starts with a fixed number of troops, B_0 and R_0. All these troops are presumed to be in contact and fighting continuously. The dynamics of the battle are described by the attrition equations in Table 1 modified by the incremental reinforcements whenever the reinforcement thresholds are crossed. The attrition coefficients were chosen as powers of 2 to aid in computational precision. The time step of the model is inherent in the selection of the attrition coefficients (and reinforcement delays) and in this case it has been chosen to represent about half an hour of simulated battle per step.

In the example of Table 1, the Blue "commander" calls for reinforcements whenever the Red-to-Blue force ratio exceeds four *or* whenever his force drops below 80 percent of his initial troop strength. In this model, he may call for only one reinforcement block at a time and after he has called for reinforcements, he may not call for more until those he just called for arrive. All reinforcements are delayed by the number of time steps specified by "Reinforcement delay." The 70 time-step delay in Table 1 represents about 35 hours in the simulated

[1] The exemplar has been specifically chosen to illustrate the concept of non-monotonicity. In the actual model, all of the numerical values shown in Table 1 are variables. The operations are performed in the order: reinforcement calls, reinforcements, withdrawals, and attrition. These and other details are spelled out in the code as catalogued in Appendix A.

[2] In combat modeling parlance, this is the so-called "Lanchester square law" formulation with constant coefficients. For mathematical purposes it is important to notice that the equations are linear in all variables and that the attrition coefficients are constant.

Table 1

SIMPLE COMBAT MODEL

Parameter	Blue	Red
Initial troop strength	B_0	R_0
Combat attrition calculation	$B_{n+1} = B_n - \frac{R_n}{2048}$	$R_{n+1} = R_n - \frac{B_n}{512}$
Reinforcement threshold	$\frac{R_n}{B_n} \geq 4$ *or* $B_n < 0.8\ B_0$	$\frac{R_n}{B_n} \leq 2.5$ *or* $R_n < 0.8\ R_0$
Reinforcement block size	300	300
Maximum allowable reinforcement blocks	5	5
Reinforcement delay (time steps)	70	70
Withdrawal threshold	$\frac{R_n}{B_n} \geq 10$ *or* $B_n < 0.7\ B_0$	$\frac{R_n}{B_n} \leq 1.5$ *or* $R_n < 0.7\ R_0$

battle. The reinforcements come in blocks and the commander has a maximum number he can call for.

Not all combat models produce a loser or winner but all have stopping criteria. In this model, the stopping criteria always cause one or both sides to withdraw, thus ending the battle. In Table 1, the Blue commander will withdraw (thus "losing") if the Red-to-Blue force ratio exceeds 10 or if his force is below 70 percent of his initial troop strength. In what follows, Red will be declared the "winner" of this battle unless he, too, withdraws on the same time step. In that case, the battle is a "draw."

Figure 1 represents 2001 battles using the parameters in Table 1, with Blue's initial troop strength fixed at 839 troops and Red's ranging from 1500 to 3500. The outcomes represent monotonic behavior in that once Red wins (as he does when starting with 2696 troops), adding more Red troops at the start of the battle does not change the outcome. However, if Blue's initial troop strength is fixed at 500 troops and another series of battles is run, the outcomes are as shown in Fig. 2. Here the battles range over Red initial troop strengths from 700 to 1800. Outcomes in this region exhibit seriously non-monotonic behavior in that Red wins when starting with as few as 884 troops, loses when starting with as many as 1623 troops, and suffers a surprising number of reversals of fortune in between. In this region, the outcome of an individual battle is not very predictive of the outcome of a "nearby" battle (i.e., one starting with nearby initial troop strengths). Said another way, the outcome of a battle in that region, from 700 to 1800 starting troops, is sensitive to small variations in Red's initial strength.

Figures 1 and 2 are one-dimensional in the sense that they illustrate the outcomes of battles as one varies the number of Red troops. More generally we will be interested in two-dimensional pictures in which the outcomes are plotted along both Blue and Red starting troop dimensions. Figure 3 is an example of such a plot (again using the values in Table 1) with black points representing Red wins and white points representing Blue wins. With few exceptions, in this report plots of this type will have granularity of 10 in each dimension. That is, the points plotted are the outcomes of battles with starting troops that are multiples of 10. Some detail is lost in so doing, but by this compromise, we get to see more of the outcome space without seriously affecting the indications of non-monotonicity. Figures 1 and 2 represent horizontal slices in Fig. 3, as shown. This figure emphasizes the surprising extent over which non-monotonicities may be found.

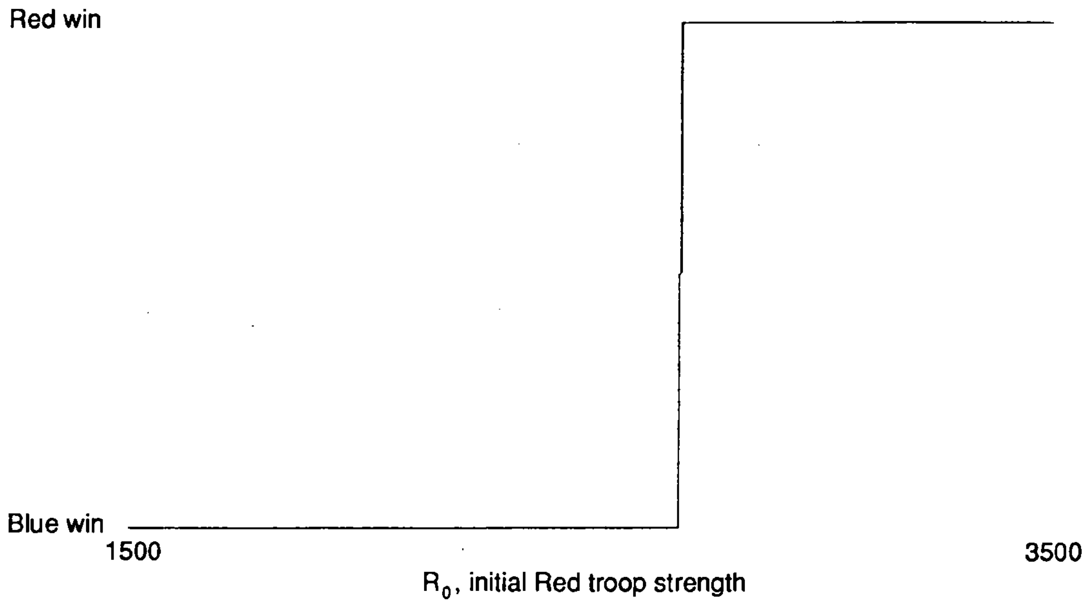


Fig. 1—A range of battle outcomes from the model in Table 1 with B_0 = 839

As simple as this model is, there are 18 different parameters that may be varied making the input space of this model 18-dimensional. Understanding what happens to this model requires understanding what happens across all 18 dimensions. Because we were interested in understanding the non-monotonic behavior and because exploring 18 dimensions is a formidable computational task, we also looked at two subsets of this model. If one deletes the four thresholds dealing with force ratios in Table 1, one has a smaller model that we refer to as the "attrition-only" model. Deleting, instead, the four attrition thresholds yields a "force-ratio-only" model. Each has input spaces of "only" 14 dimensions. Not only did this reduce (somewhat) the computational formidability of the task, but it allowed us to ask and (partially) answer some questions about what happens to non-monotonicity when one combines thresholds in a simple combat model.

CAUSES OF NON-MONOTONICITY

In an investigation of chaos and non-monotonicity involving numerical computations, one must ensure that the conclusions are not vitiated by artifacts introduced by the computational process. Digital computers require that modeled processes be described in discrete terms. In our model, for example, time is represented by a series of discrete intervals, rather than as a continuous stream. It is well known to modelers (and others) that working with discrete representations of continuous variables can cause unexpected artifacts that are due solely to having made the discrete approximation. Particularly in models with thresholds, discrete approximations can cause thresholds to be "erroneously" triggered, leading to "erroneous" results.

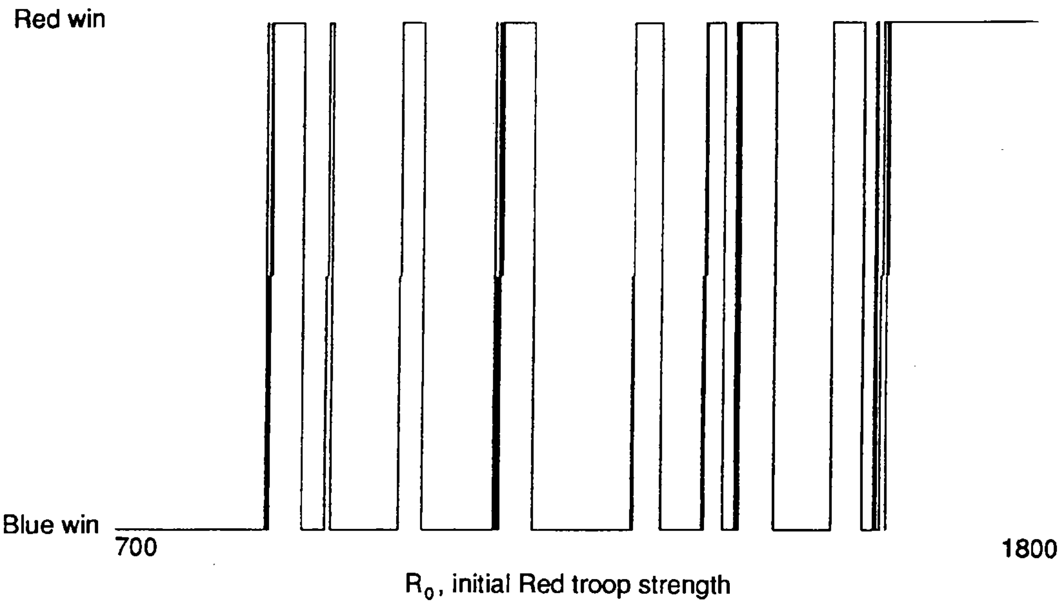


Fig. 2—A range of battle outcomes from the model in Table 1 with $B_0 = 500$

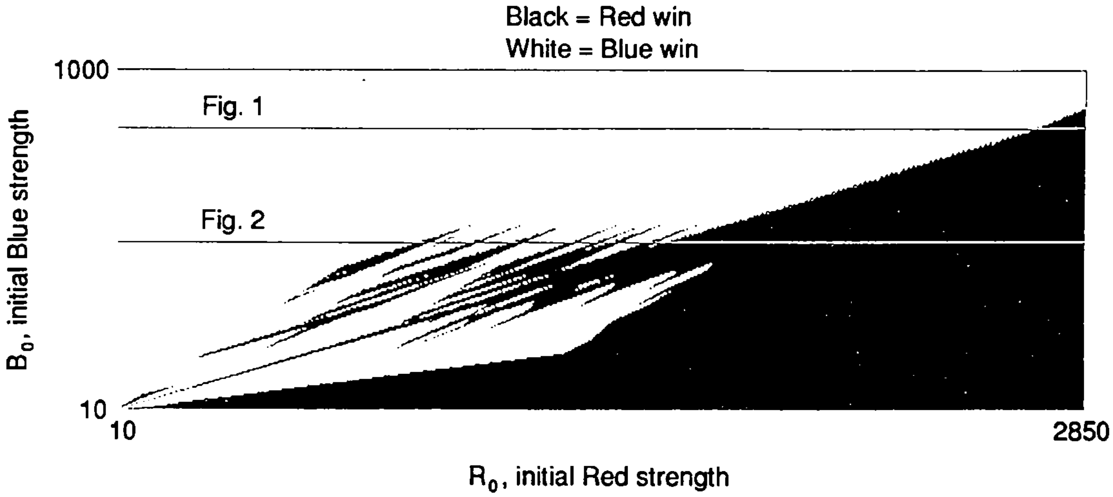


Fig. 3—An example of non-monotonicities over a range of Blue and Red starting values

In combat models there can also be the reverse problem in that discrete events are approximated by "continuous" variables. A good example of this in our simple model is in the attrition equations. The number of Blue troops lost in a given time step is a fraction of the number of Red troops in contact during that time step. This generally results in a fractional number of Blue troops lost—an arguable result since combat is typically fought by integral numbers of troops and systems. Either proceeding with a fractional number of troops or turning that number back into an integer may produce artifacts in the results of the model.

A variety of such computational considerations goes into the building and verifying of a combat model. An important goal of a combat modeler is to minimize any artifactual effects of such computational limitations that may invalidate or distort conclusions. In our model we are particularly interested in minimizing these effects because we do not want our speculations about the role of nonlinear decision heuristics to be contaminated by effects that are due to more common causes. To this end, in the subsections below we will specifically discuss two particular types of computational considerations in our simple model—roundoff/precision and time-step granularity. We will also mention a few others.

Roundoff and Precision

Computer calculations are subject to errors because of how computations are rounded off by the machine and how many significant digits the machine carries in the computations. To check for roundoff and precision problems, Palmore and Herring (1990) suggest running a combination of both roundoff and precision options. They recommend four roundoff options: round off to the nearest even binary digit, truncate, round up, and round down. The three precision options recommended are single, double, and extended precision arithmetic. If all 12 combinations of these cases are run with a given model and there are few differences, one can be fairly certain that roundoff problems are not affecting conclusions drawn from the model's computations.

We mitigated precision problems by choosing our attrition coefficients to be powers of 2. This has the advantage of making the computations (on a binary machine) both quicker and more accurate without seriously affecting the realism of the model.

Actually testing for roundoff and precision problems was a different question. Since the Sun 4/110 we were running on did not support these calculation options easily, we were forced to improvise our check. The only option we tested directly was to run the model in both single and double precision around an area of significant reversals and to note that there were no significant changes in the outcomes.

On the other hand, we have indirect evidence that there are no precision problems by looking at the attractor in phase space (see Sec. III for details). Phase space in this case is the Red-Blue plane of remaining troops. For our purposes here, it is sufficient to know that the model in Table 1—without restrictions on the number of reinforcements—will generate an "attracting" region (i.e., one to which other points in phase space are drawn and one from which points cannot escape after entering). It is also possible to describe an attractor analytically from a continuous version of the model. If there were significant problems with roundoff or precision in the computations, the computational attractor would be different from the analytic one. The fact they are very close is sound evidence that computational roundoff and precision are not playing a large role in our simple combat model.

Time-Step Granularity

The primary equations in our simple model are the computations of the number of troops remaining at the end of each time step. Although we have not chosen a specific length for our time step, it is inherent in our selection of the attrition coefficients and considerations of real combat. That is, in real combat, attrition occurs, it can be measured for historical battles, and "average" attrition rates can be computed. If our combat model is to approximate real combat, our attrition coefficients should reflect nominal values of attrition rates. Therefore, the coefficients we choose are "realistic" for a range of time periods and unrealistic for others. The attrition coefficients in Table 1 represent reasonable attrition rates for about half-hour time periods. It is in this sense that the time step for our model is about half an hour.

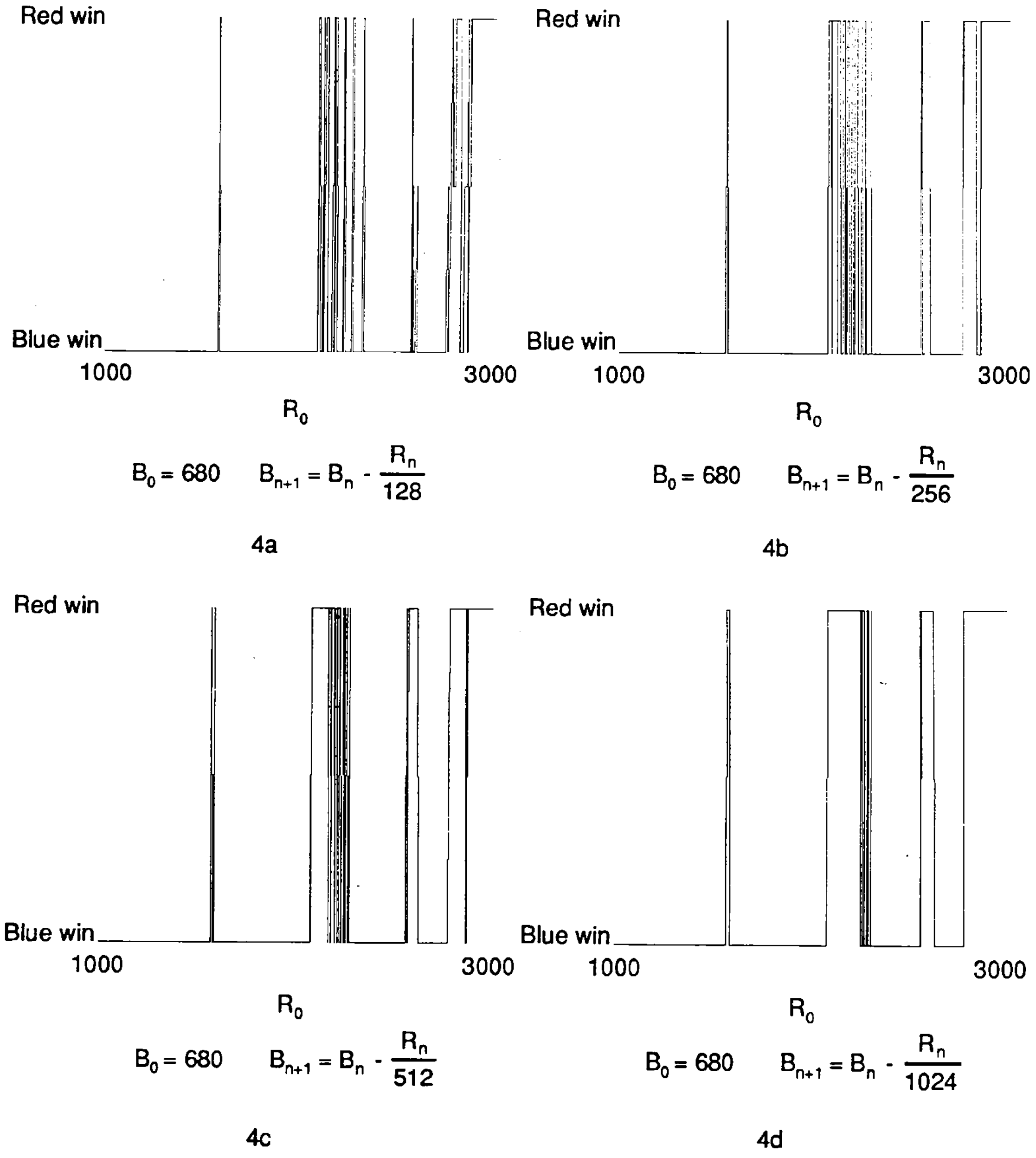
More important, however, the discrete time steps over which the attrition is computed correspond to a discrete approximation of a continuous "average" attrition. It is in this approximation sense that time-step granularity is a consideration. If attrition coefficients 10 times as large as in Table 1 are chosen, they are "realistic" in the sense that they represent average attrition over a span about 10 times as long. However, it is also well known that the goodness of the approximation to integrated continuous attrition depends on the size of the time step chosen, and the goodness of that approximation can also have serious implications for the outcome of our simple combat model. For example, if the attrition coefficients in Table 1 are taken to be 10 times as large (and the delay times are multiplied by one-tenth), the model represents an approximation to an underlying set of differential equations with time steps 10 times as large. With these coefficients, the plot of outcomes as a function of Red and Blue starting troops would differ significantly from those in Fig. 3.

To control for these artifacts, one needs to take smaller effective time steps.[3] But there is a penalty to be paid in terms of how long it takes the model to run. For our purposes, we selected the tradeoff by looking at a variety of time steps in the area of a particularly non-monotonic region of our model. Figure 4 shows the same one-dimensional slice of outcome space from Fig. 3 for a variety of time-step sizes. Although there continues to be a slight change in the number of reversals at time steps smaller than those used in Table 1, the reversals evident in the plots seem clearly to be due to something other than time-step granularity.

Another, less precise measure of time-step sensitivity is related to the number of draws produced by the model. The withdrawal thresholds are sharp enough that, with the different attrition rates and initial troop strengths of the combatants, there should be very few draws in a model with properly chosen time-step granularity. If the granularity is too coarse, attrition at each step can be large enough to cause both sides to cross even a sharp threshold at the same time a disproportionate number of times. In our model, there were several areas of draws when the time-step granularity was 10 times as coarse as in Table 1. At granularities at least as small as those in Table 1 there were rarely draws.

Although further research may reveal additional time-step granularity problems in our model, the above reasoning was compelling enough for us to choose the granularity in Table 1 with a reasonable expectation that any non-monotonicities found in the behavior of our model would not be primarily due to time-step granularity problems.

[3] In cases for which there is a closed form solution of the attrition equations (such as the Lanchester square law formulation used in our simple model), there is another approach to time-step granularity problems. One would use the closed form solution and root finding techniques to determine the exact time at which the forces or force ratios drop below a threshold value. We chose the time-step integration method for our work because of its greater generality.



$B_0 = 680 \qquad B_{n+1} = B_n - \frac{R_n}{128}$

4a

$B_0 = 680 \qquad B_{n+1} = B_n - \frac{R_n}{256}$

4b

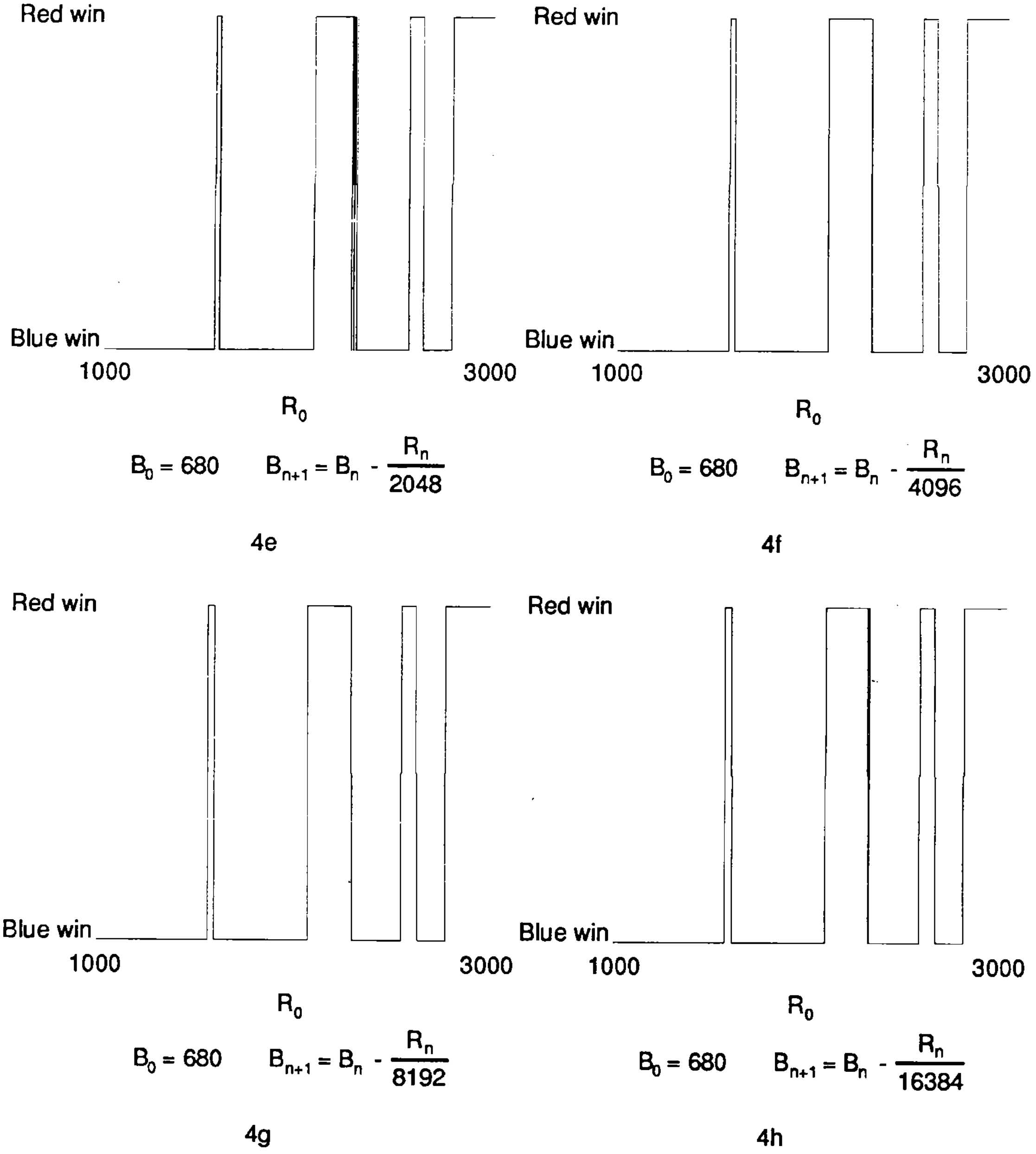
$B_0 = 680 \qquad B_{n+1} = B_n - \frac{R_n}{512}$

4c

$B_0 = 680 \qquad B_{n+1} = B_n - \frac{R_n}{1024}$

4d

Fig. 4—Time-step granularity test for eight different granularities



$B_0 = 680 \qquad B_{n+1} = B_n - \frac{R_n}{2048}$

4e

$B_0 = 680 \qquad B_{n+1} = B_n - \frac{R_n}{4096}$

4f

$B_0 = 680 \qquad B_{n+1} = B_n - \frac{R_n}{8192}$

4g

$B_0 = 680 \qquad B_{n+1} = B_n - \frac{R_n}{16384}$

4h

Fig. 4—Time-step granularity test for eight different granularities (cont.)

Other Causes

Roundoff/precision and time-step granularity are probably the most common causes of non-monotonicities encountered in developing a combat model. Again, with proper care, they can generally be eliminated as sources of non-monotonicity. However, there are other known causes of non-monotonicities in combat models as well. Although we do not attempt to completely catalog all such causes, there are a couple others worth mentioning.

One cause is related to reinforcement delays. Although such delays are common in real battle, in the model they set up what is known in control theory as lagged feedback in the dynamical system being modeled. Such lagged feedback can be stabilizing or destabilizing and, thus, *can* cause strange behavior—including non-monotonicities. In our computational investigations we explored regions with nonzero reinforcement delay, but, to ensure that lagged feedback was not the cause of the chaos or non-monotonicities, all of our mathematical work was done with zero delay (for which there still exist non-monotonicities).

Another subtle cause has showed up in large simulations and has to do with the sequential nature of digital computers. Events that would happen simultaneously in the real world must be presented sequentially to a sequential computer and this can cause problems. In both the VIC model (used widely by the Army for ground combat analysis)[4] and RAND's RSAS model (used for strategic analysis)[5] there have been instances in which changing the order of the presentation of computations to the processor has resulted in differing battle outcomes. Our simple model is insensitive to the order in which the computations are done within a given time step.

There are other potential sources of non-monotonicities in combat models—such as smoothing or time-averaging of results or the rounding off of casualties to the nearest whole troop (actually another form of roundoff)—to which our simple combat model is immune. To the extent possible, then, we have eliminated all known artifactual sources of non-monotonicities in our simple model so the resulting non-monotonic behavior must be related to the nonlinearities we wish to study.

AN ADJUSTED SIMPLE MODEL WITH NON-MONOTONICITIES

The model used to produce Fig. 3 had parameter values chosen to minimize roundoff/precision problems and time-step granularity problems. In addition, it was analyzed for all the other causes of non-monotonicities we mentioned above, *except* those caused by delayed reinforcements. There was a 70 time-step reinforcement delay in the battles represented in Fig. 3. To the extent that such a delay is realistic, Fig. 3 represents the behavior of that model. To the extent that the delay could be causing non-monotonicities apart from the non-monotonicity sources we are interested in, we would like to get rid of them. This is easily done by setting the delay to zero. This may not be realistic in a combat modeling sense, but we are now more interested in the behavior of the model without reinforcement delay as a potential source of non-monotonicities. Figure 5[6] represents an example of a simple combat model that has controlled for all well-known sources of non-monotonicities *including* those caused by delayed reinforcements. It is the non-monotonicities in this figure that we are specifically interested in investigating.

[4] From the work of one of the authors (J. J. Gillogly).

[5] Private communication, W. Perry.

[6] See Appendix B for the specific parameters used to generate this figure and all remaining figures in this report.

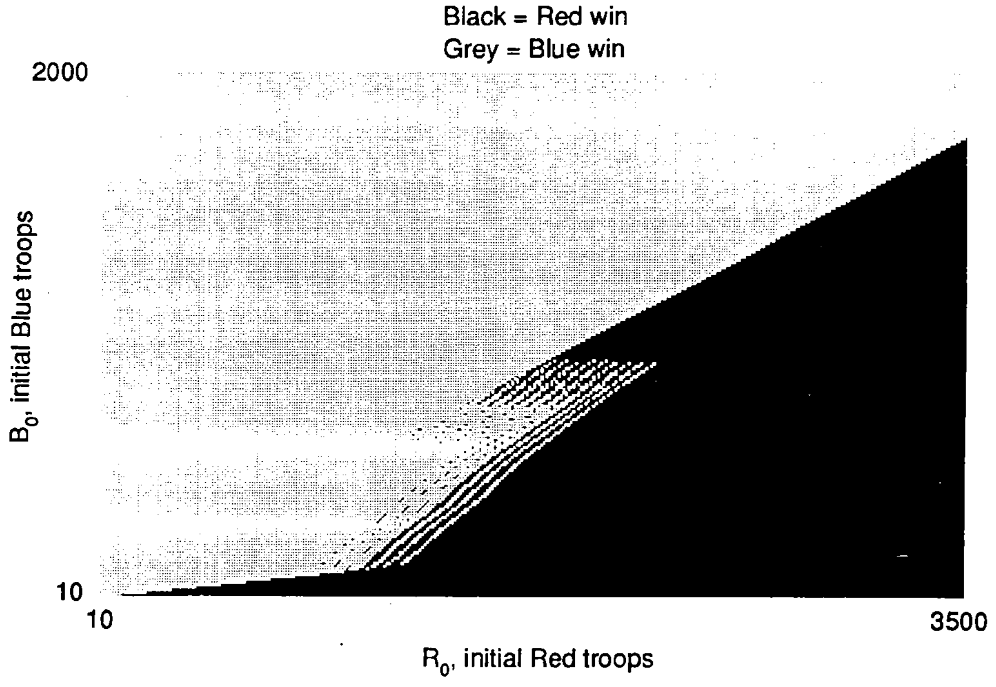


Fig. 5—Non-monotonicities in a simple combat model controlling for all well-known non-monotonicity sources including delayed reinforcements

III. CHAOS IN A SIMPLE COMBAT MODEL

Whether or not the non-monotonicities observed in the behavior of simple combat models are related to chaos, they are matters of concern to modelers. If they are caused by or related to chaos, however, they are likely to be both more difficult to predict and more difficult to eliminate without eliminating the nonlinearities that produce them. It is of more than passing interest, then, to know if and how the non-monotonicities might be related to chaos.

Much of the behavior of the non-monotonicities observed in our simple model has the flavor of mathematical chaos in that there is apparent sensitivity to small variations in initial conditions. Further, combat models with their attrition and reinforcements are akin to systems with state-dependent forcing and damping in which chaos is often found. Actually proving that our simple model is chaotic and that the non-monotonicities are related to that chaos is not a simple matter, however.

One problem in relating the two is that chaos tends to be a long-term phenomenon, but combat models are typically run only over limited duration battles and it is unclear whether or not chaotic behavior can be detected in this amount of time if, indeed, it is present. Another difficulty is in defining what is meant by a chaotic mapping. Several definitions are available in the literature and none of them is strictly applicable to our model. A third difficulty is in actually relating the non-monotonicities to the chaotic behavior. In this section we will discuss what we mean by chaos, demonstrate that chaos, in our sense, is present in our simple model, and show how the non-monotonicities are related to that chaos.

DEFINITION OF CHAOS

There is still no generally accepted definition of chaos.[1] Even for continuous flows and maps,[2] the literature has several definitions. Our case is complicated in that, even though the equations for our model are linear where they are continuous, they are only piecewise continuous with the discontinuities at the times of reinforcement. There are discontinuous maps which satisfy published definitions of chaos (the baker transformation is the classical example),[3] but ours does not in the strict sense. What we have done about that requires a brief description of how we went about looking for chaos.

Typical published definitions of chaos deal with the asymptotic (as time approaches infinity) behavior of dynamical systems. There is no asymptotic behavior in our simple model. The battle proceeds for a finite number of steps and then one side or the other withdraws, ending the battle. If we remove the withdrawal criteria, the battle would still be finite because of the finite number of reinforcements available (and the inexorable attrition that eventually leads to annihilation of one side). If we allow for an unlimited number of

[1] Lichtenberg and Lieberman (1983). See p. 213 for a list of six characteristics with references that various authors have used. For a classical definition, see Li and Yorke (1975), pp. 985–992.

[2] In a dynamical system time may be treated either as continuous or as discrete. If continuous, the process is usually described by a system of differential equations and this is referred to as a *flow*. In cases where discrete time is used (typically either for computational feasibility or because the system is sampled that way) the resulting system is called a mapping or *map*.

[3] See, for example, Devaney (1989).

reinforcements, the battle will continue indefinitely with no winner or loser and we can now *almost* study the asymptotic behavior of this dynamical system that "underlies" our simple model. In fact, this revised model is still too complex to check for chaotic behavior, so we made use of the submodels discussed briefly in Sec. II. The submodel we used for exploring chaos was the force-ratio-only submodel. The attrition-only submodel was studied by researchers at the Center for Nonlinear Studies at Los Alamos National Laboratory.[4] Looking at a submodel of our original model is not a crucial concession for our purposes. Our primary interest is in investigating the behavior of any model that exhibits chaos and nonmonotonicities. In fact, however, a secondary interest is in the behavior of a model when thresholds are added (such as adding the attrition thresholds to the force-ratio-only submodel). The latter subject is taken up in the next section.

To understand our definition of chaos, we begin with the definition of chaos found in Devaney (1989).

Let V be a set. f: V → V is said to be chaotic on V if

1. f has sensitive dependence on initial conditions.
2. f is topologically transitive.
3. periodic points are dense in V.

Devaney states:

> To summarize, a chaotic map possesses three ingredients: unpredictability,[5] indecomposability, and an element of regularity. A chaotic system is unpredictable because of the sensitive dependence on initial conditions. It cannot be broken down or decomposed into two subsystems (two invariant open subsets) which do not interact under f because of topological transitivity.[6] And, in the midst of this random behavior, we nevertheless have an element of regularity, namely the periodic points which are dense.[7]

A property similar to topological transitivity has to do with invariant densities or measures.[8] Although our modified model does not strictly meet these chaos criteria it does have the following properties:

1. Sensitive dependence on initial conditions.

[4] This work is described in Sholl et al. (forthcoming).

[5] A good indication of the general confusion in defining chaos has to do with unpredictability. For example, the baker transformation, which is chaotic by this definition, is completely predictable. The n-th iterate $f^n(x) = (2^n x - k)/2^n$, $k < x < k + 1$, $k = 0, 1, \ldots, 2^n - 1$, $f^n(1) = 1$ and its 2^n fixed points are given by $x = k/(2^n - 1)$, $k = 0, 1, \ldots, 2^n - 1$, these being periodic points of f(x). On the other hand, the baker transformation is quite sensitive to initial conditions.

[6] An orbit of a dynamical system is said to be *topologically transitive* if, for all pairs of regions in the phase space (no matter how small), the orbit visits each region of the pair at some point. In other words, in a topologically transitive system it is always possible to get from an area around any state to the area around any other state by following the orbit you are on.

[7] A set of points X is *dense* in a set of points Y if in any area around a point in Y, no matter how small, one can find a point from the set X. In our case, dense periodic orbits imply that there is a periodic orbit arbitrarily close to any orbit in the phase space.

[8] Under this formulation a mapping is called chaotic if it admits a nontrivial invariant density which is absolutely continuous relative to the Lebesque measure. See Lasota and Mackey (1985). An *invariant density* or invariant measure can be thought to measure the fraction of time an orbit spends in each small area of the phase space.

2. Topological transitivity.
3. An infinite number of periodic points.
4. An apparently invariant density (with gaps).

Although some of these properties are satisfied in a restricted sense, our model clearly satisfies the *spirit* of the definition of chaos in the descriptive sense that Devaney uses above. Furthermore, although satisfaction of these four criteria is sufficient to lead to non-monotonicities, it is not clear that all four are necessary. It is inappropriate at this point to introduce yet another definition of chaos (for discontinuous maps), but we will consider a map with discontinuities to be chaotic if it satisfies the four conditions above. The exact sense in which our modified model satisfies these conditions is the subject of the following subsection.

CHAOS IN THE UNDERLYING MODEL

The actual "model" we investigated for chaos is shown in Table 2. Putting it in the format of our original simple model emphasizes just how simple it is. This is slightly different from the model in Table 1 (e.g., the attrition coefficients are not powers of 2), but the differences are not important because the model in Table 2 will be investigated analytically rather than numerically. Since this "model" does not have any stopping conditions, we will refer to it as the "force-ratio-only mapping." In its functional form it can be written as:

$$\begin{bmatrix} B_{n+1} \\ R_{n+1} \end{bmatrix} = \begin{bmatrix} 1 & -0.005 \\ -0.02 & 1 \end{bmatrix} \left\{ \begin{bmatrix} B_n \\ R_n \end{bmatrix} + \begin{bmatrix} \delta_B(n) \\ \delta_R(n) \end{bmatrix} \right\}$$

where

$$\delta_B(n) = \begin{cases} 10 \text{ if } R_n \geq 4B_n \\ 0 \text{ otherwise} \end{cases}$$

and

$$\delta_R(n) = \begin{cases} 10 \text{ if } R_n \leq 2.5B_n \\ 0 \text{ otherwise} \end{cases}$$

This force-ratio-only mapping has an attracting region in the (phase) space defined by remaining Blue and Red troops as shown in Fig. 6. That is, any orbit[9] of the mapping that intersects the boundary of this "attractor" stays inside that region forever. It is the mapping above, defined on the attracting region in Fig. 6, that we investigated for chaos.

The demonstration that the force-ratio-only mapping satisfies the four conditions above is often long and grueling. Dr. Mario Juncosa, in forthcoming work, furnishes the details, but it is useful to present the general idea of the proof here. Each of the four chaos criteria mentioned above can be treated separately:

[9] For our purposes, the *phase space* of a dynamical system can be thought of as the set of all points or values that the system can take. In this case the dynamical system can take on any values in the two-dimensional space defined by remaining Blue and Red troops. An *orbit* or trajectory in that space is the time evolution of a given set of starting conditions.

Table 2

MODIFIED MODEL FOR INVESTIGATING CHAOS

Parameter	Blue	Red
Initial troop strength	40	120
Combat attrition calculation	$B_{n+1} = B_n - 0.005\ R_n$	$R_{n+1} = R_n - 0.020\ B_n$
Reinforcement threshold	$\frac{R_n}{B_n} \geq 4$	$\frac{R_n}{B_n} \leq 2.5$
Reinforcement block size	10	10
Maximum allowable reinforcement blocks	Unlimited	Unlimited
Reinforcement delay (time steps)	0	0
Withdrawal threshold	None	None

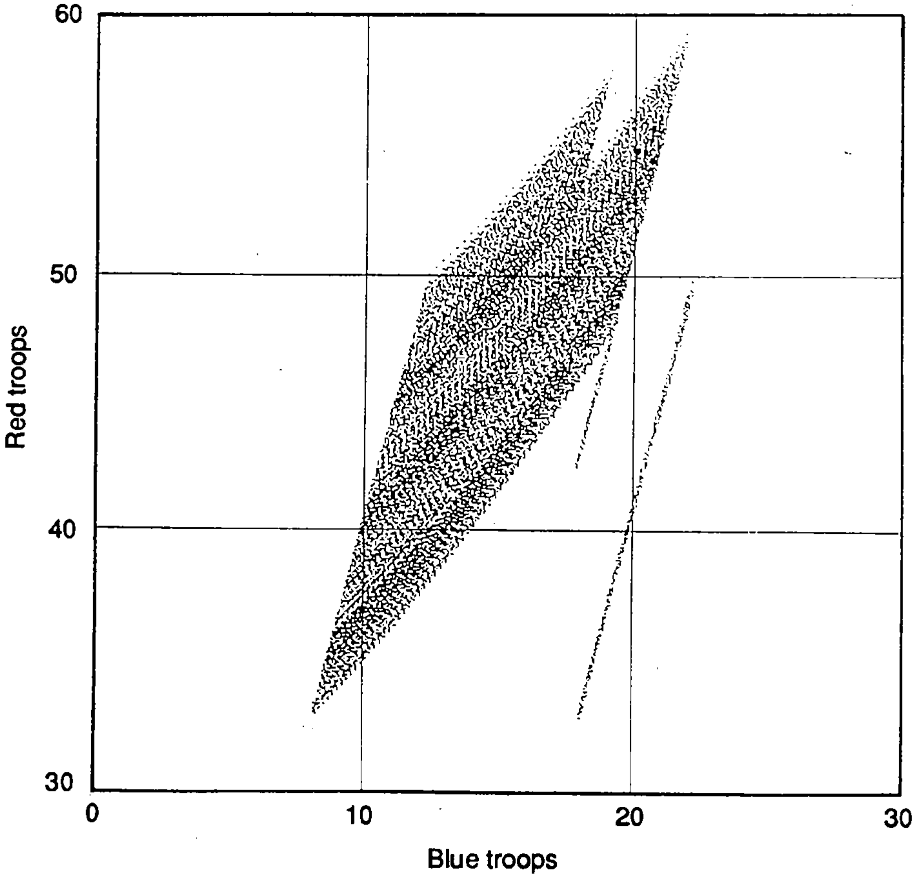


Fig. 6—Phase space attractor for the modified model of Table 2

1. **The force-ratio-only mapping has sensitive dependence on initial conditions.**

The usual means of establishing sensitive dependence on initial conditions is to show that the mapping has a positive Lyapunov exponent.[10] Our discontinuous mapping has Lyapunov exponents only in the regions where it is continuous. In those regions, the mapping has the form:

$$\begin{bmatrix} B_{n+1} \\ R_{n+1} \end{bmatrix} = \begin{bmatrix} 1 & -0.005 \\ -0.02 & 1 \end{bmatrix} \begin{bmatrix} B_n \\ R_n \end{bmatrix} \equiv A \begin{bmatrix} B_n \\ R_n \end{bmatrix} = A^{n+1} \begin{bmatrix} B_0 \\ R_0 \end{bmatrix}$$

and the matrix A has eigenvalues $\lambda = 1 \pm \sqrt{0.0001} = 1 \pm 0.01$. In the linear case, having an eigenvalue greater than 1 is the same as having a positive Lyapunov exponent (the Lyapunov exponents here being the natural logarithms of the eigenvalues), so that the sensitive dependence on initial conditions is established for the regions in which our mapping is continuous. This means that in one "direction" of the two-dimensional attractor, two points starting out close together will be driven apart, and this is sufficient for sensitive dependence on initial conditions.[11]

The fourth criterion:

4. **The force-ratio-only mapping has an invariant measure (with gaps)**

can be demonstrated with Figs. 7 and 8. Figure 7 plots the density of the Blue troop value as it crosses the R = 4B line of the attractor in Fig. 6 plotted for one million "crossings" (from an arbitrary starting point). Figure 8 plots the same density for 10 million crossings. The similarity of these two figures indicates an invariant measure. This does not constitute a proof of the existence of an invariant measure (actually proving invariant measures or densities is quite difficult), but is another correlate with chaotic behavior.

To settle criteria two and three:

2. **The force-ratio-only mapping has topological transitivity.**
3. **The force-ratio-only mapping has an infinite number of periodic points.**

we need to look at the attractor shown in Fig. 6. The proofs rest on two simplifications. First, notice how time flows in Fig. 6. Any point in the figure is a state of the modeled combat. The next state in that combat trajectory is then down and to the left of it, because the attrition diminishes each side. The system continues moving down and to the left in time until it crosses the R = 4B line. At that point, Blue reinforces and the next point jumps 10 units to the right. At that point, Red reinforces (a jump 10 units upward) twice, once, or not at all (depending on where he is with respect to his reinforcement threshold), and the battle continues. The very top line of dots in the figure is a good example of one segment of the battle represented by this figure.

[10] The Lyapunov exponents of a dynamical system are a measure of the stability of the system. In a linear system (or in the linear pieces of a piecewise linear system such as ours) the Lyapunov exponent is related directly to the eigenvalues of the solution of the system and these determine whether small perturbations of an orbit can produce wide divergences at later times.

[11] On a speculative note, a more general definition of chaos might use the fact that positive Lyapunov exponents, except on a set of measure zero, would still satisfy one's intuitive impression of sensitive dependence on initial conditions.

In the regions between reinforcements the model can be approximated in the limit (as the time increments tend toward zero) by a continuous model. This is tricky, however, because the continuous approximation of a discrete system does not necessarily have to have the same dynamic properties as does the discrete system. In this case, however, they do,[12] so this can be done without loss of generality. This piecewise continuous approximation is easier to analyze. In this "continuized" version, the second simplification is to look only at the R = 4B line and watch where the successive segments of the battle intersect it (this is called a Poincaré section). Establishing criteria (2) and (3) for the force-ratio-only mapping is equivalent to establishing them for the Poincaré section on the R = 4B line. The details are left for the forthcoming reference noted above, where it is shown that the returns to the R = 4B line have topological transitivity and that there are an infinite number of periodic points.

The modified model in Table 2, then, is chaotic in the sense we are considering. What remains is to determine how the chaos in this "infinite" model is related to non-monotonicities in finite models that derive from them.

CHAOS AND NON-MONOTONICITIES

It is necessary to take some care when discussing the relation between chaos in the modified infinite model and non-monotonicities in the simple, finite combat model. In what sense can we say that the chaos in the infinite model is "causing" non-monotonicities in the finite model? If the chaos is removed from the infinite model, and non-monotonicities then disappear from the finite model, this is evidence that the one is causing the other. That this is the case here can be seen in Figs. 9 and 10. Figure 9 shows the force-ratio-only model with nominal stopping criteria. Figure 10 is the same model except that the reinforcements are not brought in according to the force ratio threshold. In this case they are "scripted," or brought in at a specific set of fixed times in every run of the battle, i.e., reinforcements are functions of time alone, not of the state of the battle. As we shall discuss in greater detail in the next section, scripting the reinforcements in this way removes the chaos from the model, and, as in the figure, the non-monotonicities generally disappear as well.

It is important to note that the models used in Figs. 9 and 10 both have stopping criteria based on the state of the battle. It is clear in this case that any nonlinearities that the stopping criteria introduce are not causing or remedying the non-monotonicities, but what effect do the stopping criteria have on the finite model in general?

There are stopping criteria in any finite model. Practically speaking, as soon as either side has fewer than, say, one remaining combatant, it must stop fighting and the attrition equations generally ensure that eventually this must happen. More commonly, however, battles (both real and simulated) are stopped long before annihilation of one side. In some cases, the battle will be stopped independently of its progress (e.g., after a certain amount of simulated or computer time has elapsed). In other cases, stopping criteria will be established that are a function of the state of the battle.

Whatever the stopping criteria, though, non-monotonicities are associated most generally with a further evaluative mapping from the final state of the battle to an ordered set of states.[13] A popular such set is the binary set "win" and "lose," but there are a variety of other

[12] The reinforcements then become delta functions with their mass concentrations (reinforcements) occurring at the appropriate points in the Blue-Red phase space.

[13] In models stopped after a certain amount of time has elapsed the evaluation differs qualitatively. In this case it is common to compare a set of measures at several points along the dynamic trajectory. Since non-monotonicities are measured for separate runs, the equivalent in this case would be the behavior of the sets of measures at equivalent time points for separate entire runs started with different initial conditions.

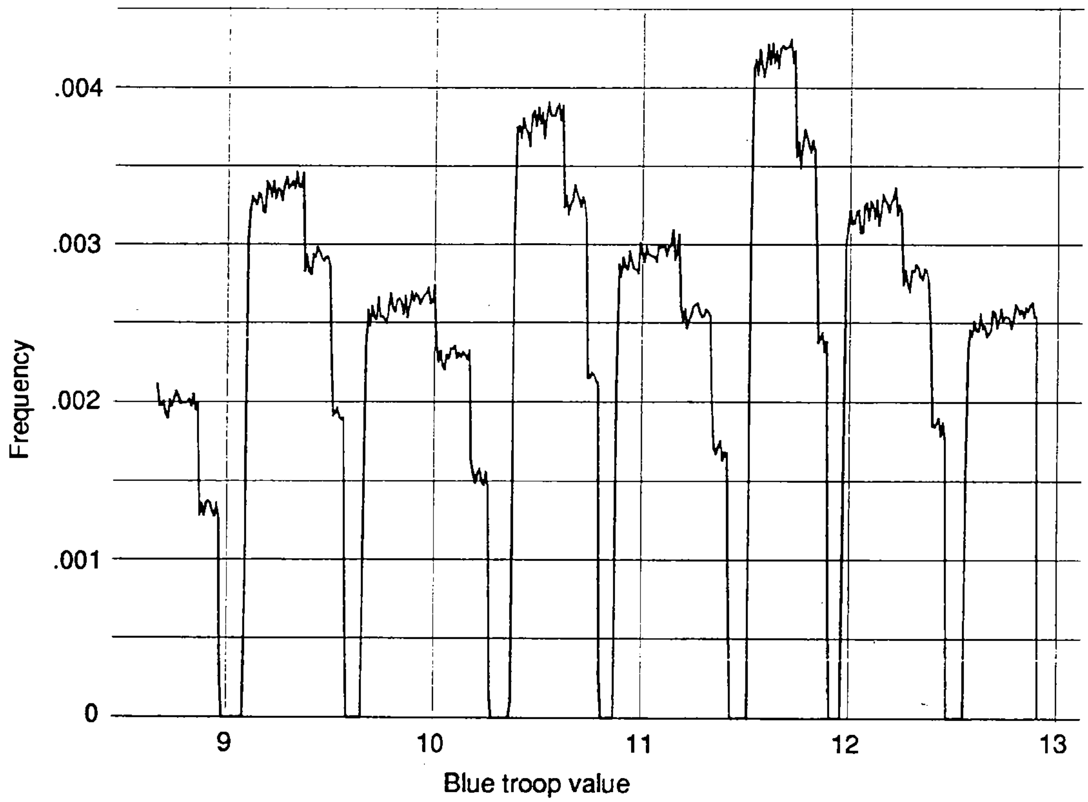


Fig. 7—Density plot of Blue troop values for Poincaré return to the R = 4B line of the modified model attractor (1 million returns)

such sets ranging from territory won or lost to multidimensional measures of materiel used, etc. As long as the evaluative mapping is ordered, then a non-monotonicity is any unexpected reversal in outcome associated with a given change in inputs.

In our case, the non-monotonicity is a change from one side winning a battle to that side losing given that the only change in inputs is an increase in that side's initial troop strength. To see most clearly that it is the nonlinearity of the function describing the reinforcement decision that is necessary to cause non-monotonicities to appear in our finite battles, we will look at what constitutes a finite model in our case. There are two basic differences between the modified model in Table 2 and the finite force-ratio-only submodel of the original model Table 1 (with zero delay): (1) There are stopping criteria in the finite model and (2) only a finite number of reinforcements can be called upon in the finite model. To eliminate the stopping criteria as a potential source of non-monotonicities, we will get rid of them and stop the battle only when one side is annihilated. To ensure finite battles, we will restrict the number of reinforcements available to each side. If there are non-monotonicities in this model, then they must be related to the fact that the reinforcement schedules are potentially different from battle to battle because of the reinforcement heuristic.

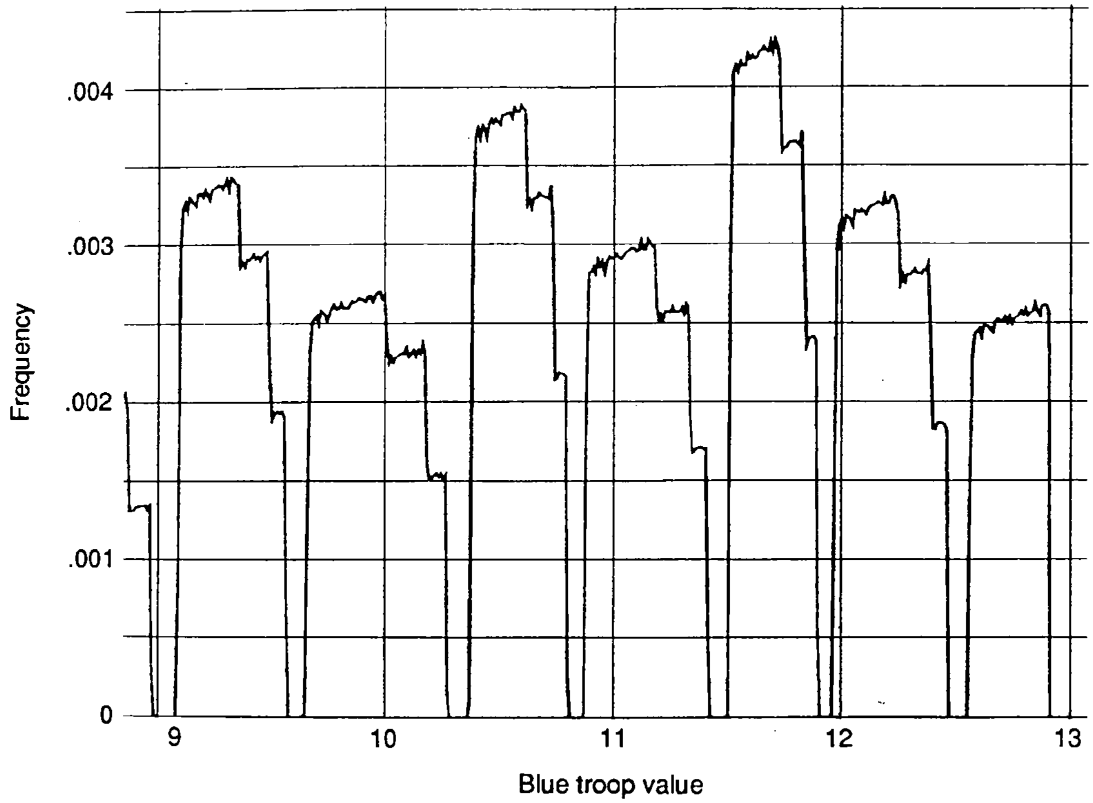


Fig. 8—Density plot of Blue troop values for Poincaré return to the R = 4B line of the modified model attractor (10 million returns)

Figures 11 through 17 show actual battle traces for seven battles of this type of model. The first three show outcome reversals given a Red reinforcement at the first step of the battle. The remaining four show further reversals in who is annihilated without that initial Red reinforcement. For a change in Red initial troop strength of 10 troops (since the total reinforcements do not change, this is the total difference in troop strengths committed throughout the battle), there is one swing of almost 300 troops (Fig. 13 and Fig. 14) in final troop strengths for the two sides.

There can be little doubt from these pictures that, if the timing of the reinforcements is changed, the progress of the battle can be changed significantly enough to alter which side will predominate. That is, the (chaotic) nonlinearities introduced by the reinforcement heuristic can produce unexpected changes in the fundamental progress of the battle. In that sense it must be said that the chaotic nonlinearities have caused "non-monotonic" behavior in the model. Non-monotonicities in model outcomes are thus a necessary corollary.

Before leaving this point, it should be noted that the *size* of the reinforcement blocks appears to play an important role in the magnitude of the non-monotonicities. In Figs. 11 through 17 the Blue reinforcement blocks are 250 troops and the Red blocks are 125 troops

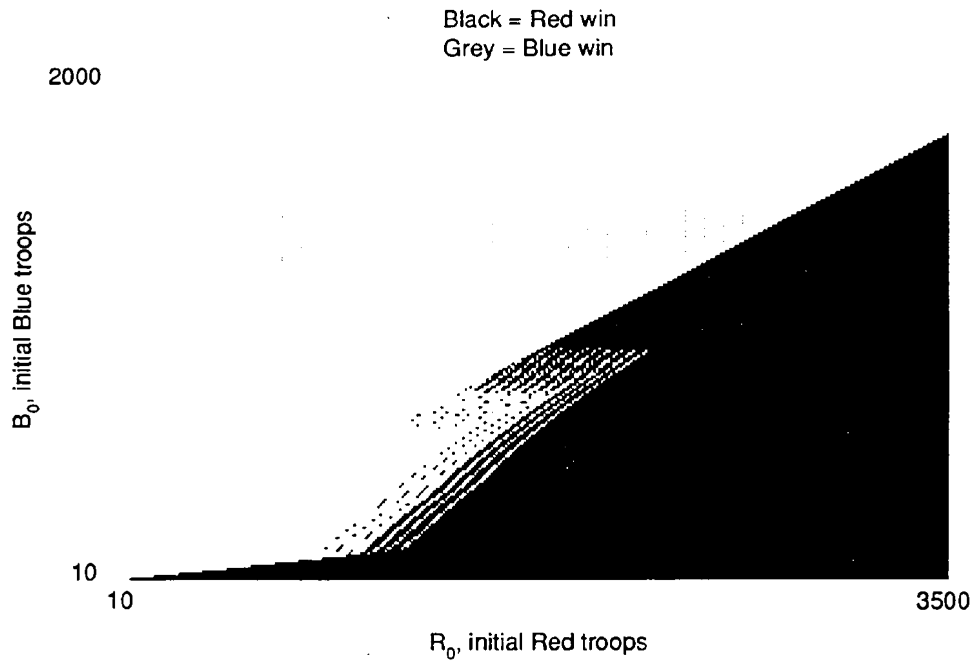


Fig. 9—Outcomes for the force-ratio-only model (with no reinforcement delay)

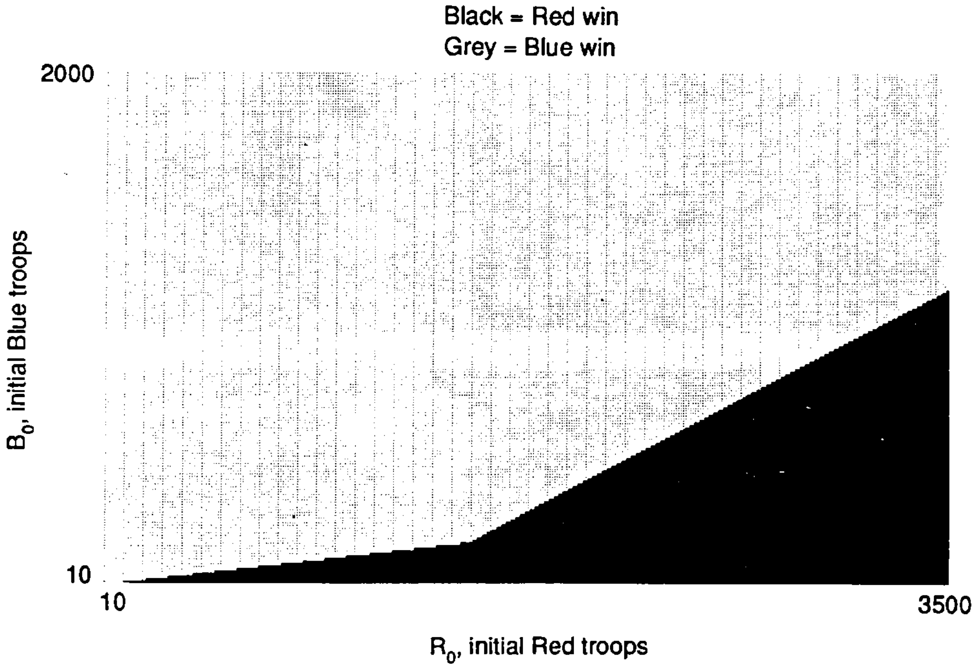


Fig. 10—Outcomes for the force-ratio-only model (with scripted reinforcements and no delay)

(although in some cases Red reinforces on two consecutive time steps). We have run cases with very small reinforcement block sizes in which the non-monotonicities all but disappear. Conversely, with larger reinforcement block sizes the magnitudes of the reversals in battle fortune are larger. The size of the reinforcement blocks does not affect whether or not the underlying equations are chaotic; however, it does affect the nature of the non-monotonicities.

In the foregoing we have attacked head-on the question of the relationship between chaos and non-monotonicities. There is another way to look at this question that relates more to the relationship between the stopping criteria and the underlying dynamics of the model. In our case, if the lines defining the stopping criteria for both combatants intersect the attractor, it can be shown[14] that, if the attractor is chaotic, there will be non-monotonicities in the outcomes of the finite model containing those criteria. That is, without a restriction on the total amount of reinforcements allowable,[15] there can be both finite battles and non-monotonicities in the outcomes.

SOME NAGGING QUESTIONS

We have shown that our force-ratio-only mapping is chaotic in the sense referred to above and that that chaos produces unexpected divergences in nearby orbits which result in non-monotonicities in finite realizations of the mapping. Put more succinctly, there is chaos in our underlying model and it leads to non-monotonicities in our finite model. That said, a variety of interesting and important questions still lack answers:

1. **Is it necessary to have a chaotic attractor to produce non-monotonicities?**

From the work of Perkers et al. (1990), if one plots a *periodic* orbit of a force-ratio-only model, one can define win-lose criteria that will produce non-monotonicities as well. Periodic non-monotonicities of this type, however, ought to be quite predictable. On the other hand, the chaotic reversals are probably going to be unpredictable, and that may be the crucial difference. From a military science standpoint, the arrival of reinforcements is unlikely to be nicely periodic and equally unlikely to produce periodic fluctuations in the progress of the battle.

2. **How chaotic is the attractor for the force-ratio-only mapping?**

There is ample evidence that some aspects of the chaotic behavior in our attractor can take time to become apparent. The fact that the positive Lyapunov exponent is close to 1 says that nearby orbits will diverge, but not very quickly. Figure 18 is a trace of one chaotic orbit as it passes several times through the attractor. Each pass is numbered and it can be seen that parts of the orbit seven passes apart are relatively close together. The fact that the attractor is chaotic means that they must eventually get arbitrarily far apart, but they clearly do it somewhat slowly. This also agrees with the fact that there are orbits of period seven in the attractor. Yet another indication that the force-ratio-only mapping is only mildly chaotic is shown in Fig. 19. The battles in Figs. 11 through 17 were taken from this figure. Although the changes in those figures are dramatic, it is clear that they are reasonably well confined in extent.

[14] See Perkers et al. (1990).

[15] Note here that if a stopping criterion intersects a chaotic attractor, then topological transitivity of the attractor guarantees that a chaotic orbit will eventually enter the region beyond the stopping criterion, stopping the battle.

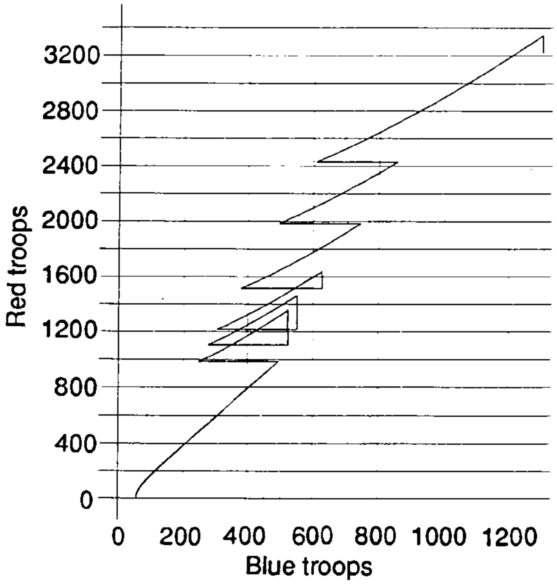


Fig. 11—Trace of battle with $B_0 = 1300$, $R_0 = 3220$

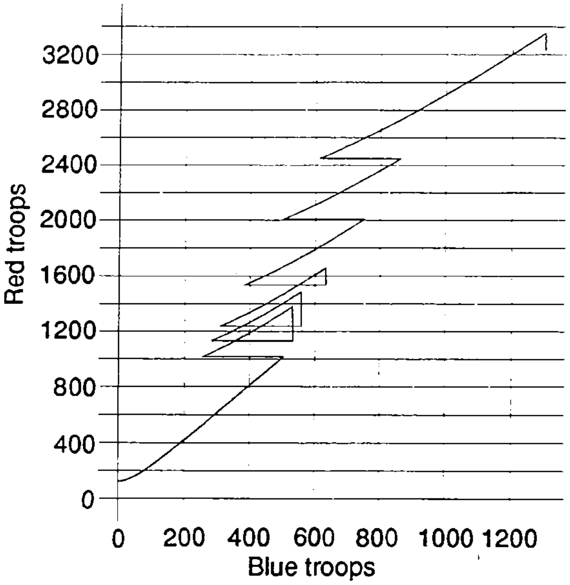


Fig. 12—Trace of battle with $B_0 = 1300$, $R_0 = 3230$

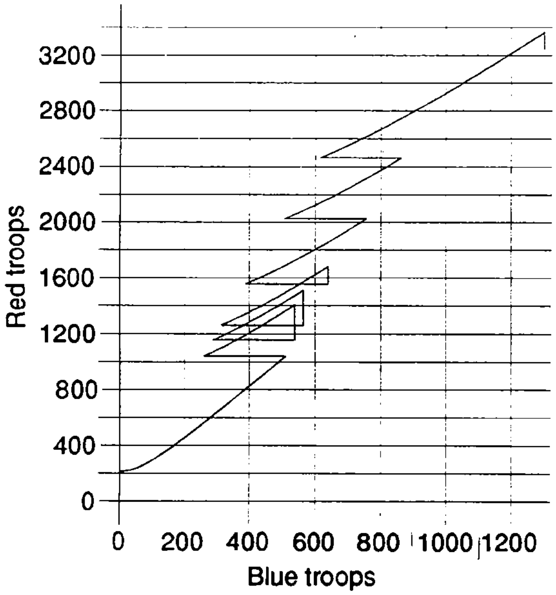


Fig. 13—Trace of battle with $B_0 = 1300$, $R_0 = 3240$

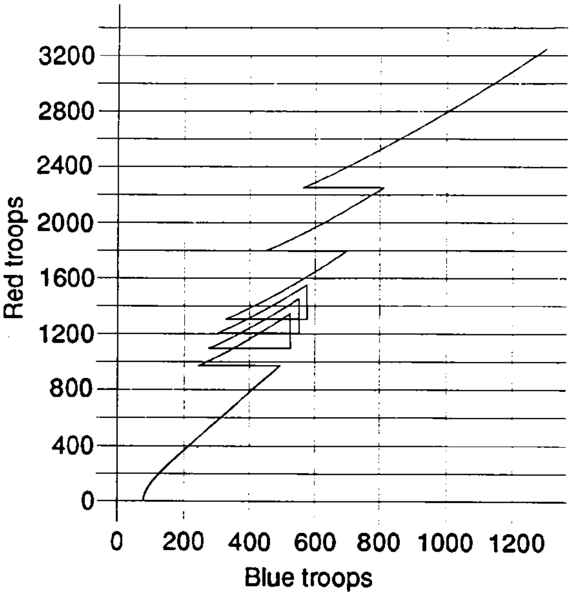


Fig. 14—Trace of battle with $B_0 = 1300$, $R_0 = 3250$

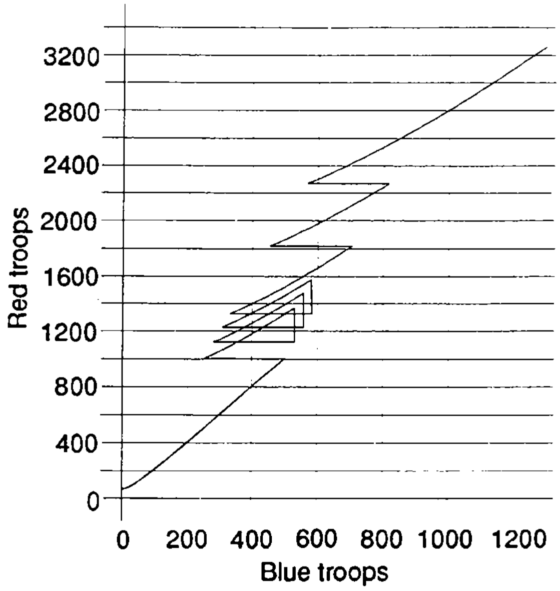


Fig. 15—Trace of battle with $B_0 = 1300$, $R_0 = 3260$

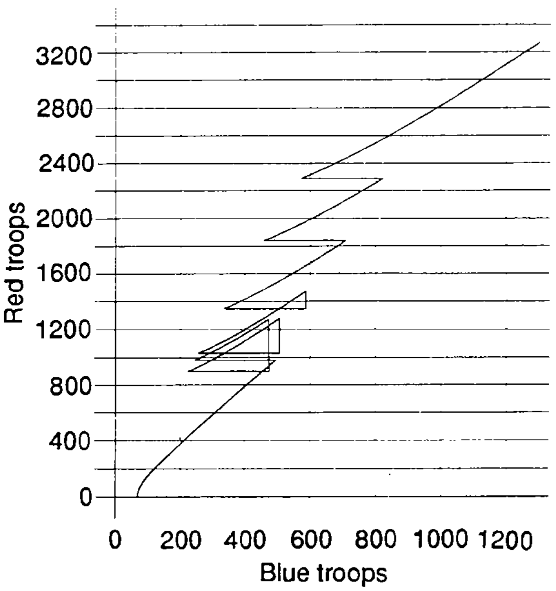


Fig. 16—Trace of battle with $B_0 = 1300$, $R_0 = 3270$

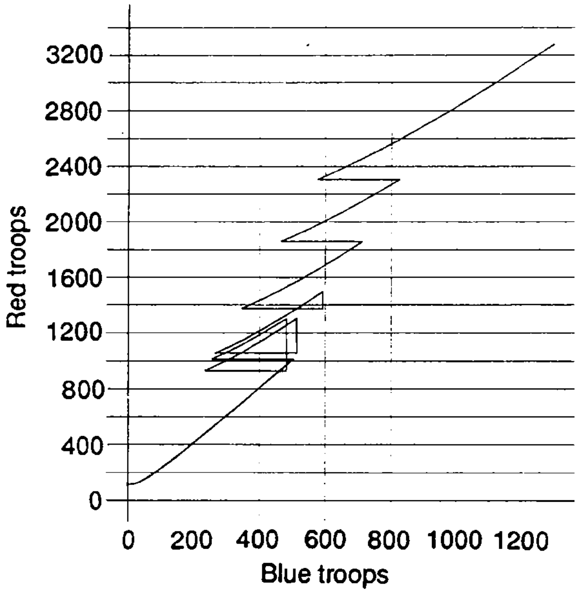


Fig. 17—Trace of battle with $B_0 = 1300$, $R_0 = 3280$

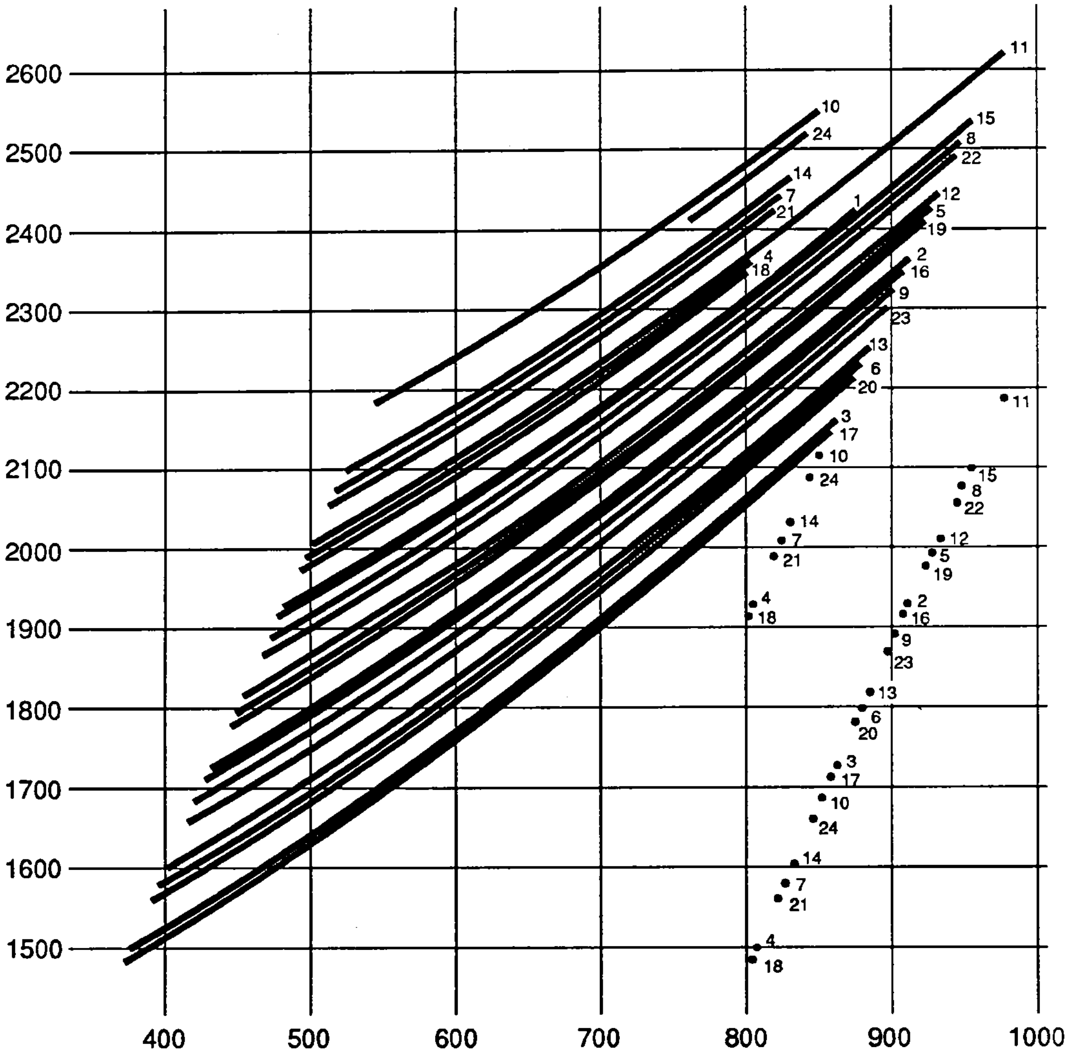


Fig. 18—A partial orbit (24 passes through the attractor) of the force-ratio-only model showing quasi-periodicity

3. What role (if any) does chaos play in other non-monotonicities?

This question stems from the fact that non-monotonicities caused by other problems seem to be exacerbated by the chaos in the underlying model. Non-monotonicities caused by reinforcement delays or by adding thresholds seem to be much worse with the underlying chaos (Fig. 3) than without. Direct comparisons are difficult because it is not clear which non-chaotic model should be compared with the chaotic one. There is some evidence, however, that non-chaotic underlying models do not particularly complicate the phase plots of the simple model having reinforcement delay or two types of reinforcement thresholds.

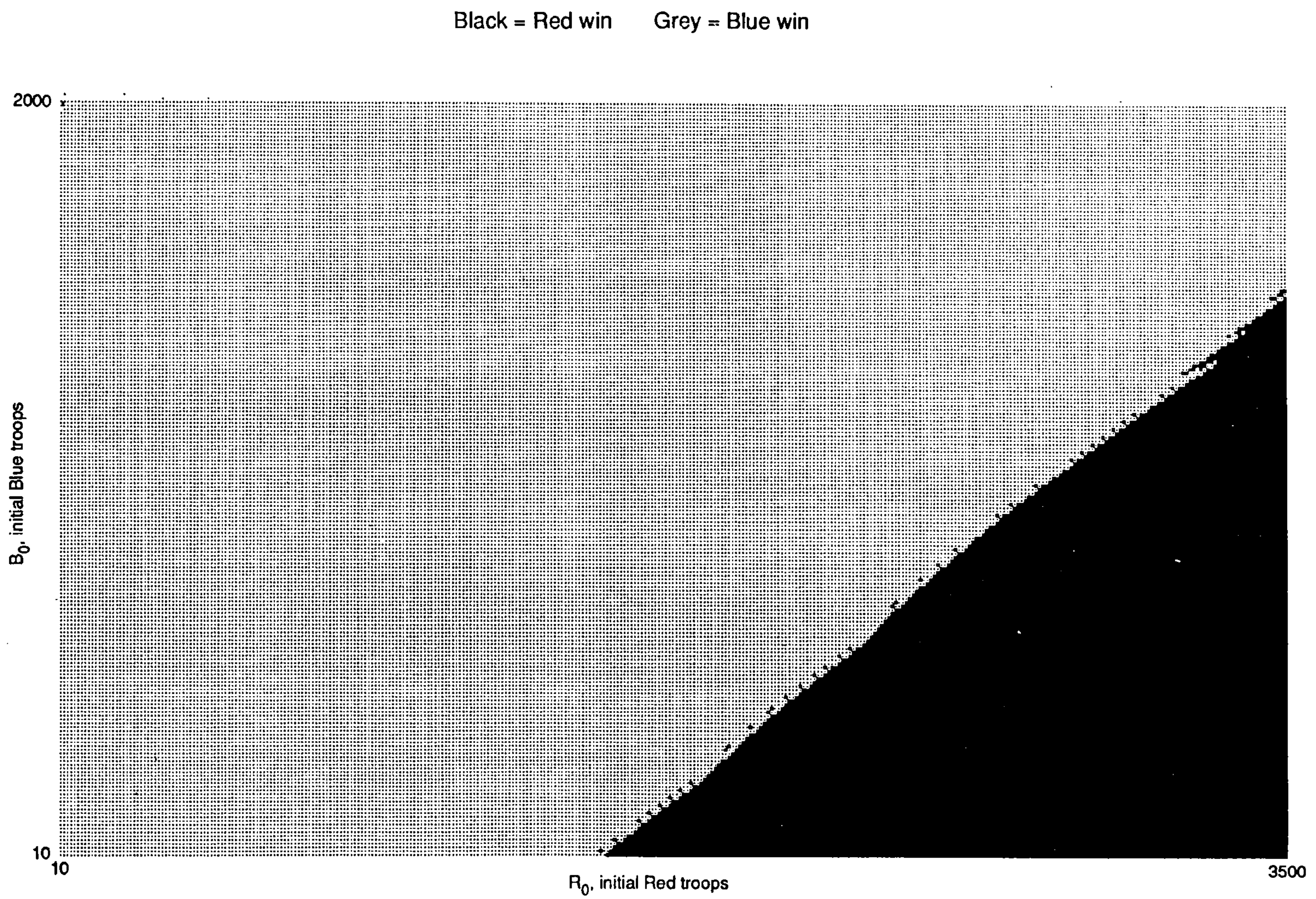


Fig. 19—Outcomes of the force-ratio-only model with annihilation as the stopping criteria (for both sides)

4. What role (if any) do the stopping criteria play in the non-monotonicities?

Another question of some interest is the role of the stopping criteria in the non-monotonicities. As seen above, they played no role in getting rid of the non-monotonicities by scripting the reinforcements. On the other hand, Perkers et al. (1990), working with first crossings, indicate that the stopping criteria can be chosen to make the non-monotonicity problem better or worse. If they do not intersect the attractor, then the battle will halt only after at least one side has used up all its reinforcements. In some cases this appears to lead to fewer non-monotonicities than when the stopping criteria do intersect the attractor. It might even be possible to identify stopping criteria that dampen the non-monotonicities caused by chaotic dynamics in the underlying model. Then, too, there is the question of whether stopping criteria that depend on the state of the battle are better or worse than stopping criteria that do not. Preliminary indications are that the former might be worse than the latter, but we have been surprised before.

With respect to chaos and non-monotonicities, these are the major unanswered questions at this time. However, questions of a different nature remain that appeared more pressing to us and to which we turned our attention. Namely, given that chaotic dynamics can cause non-monotonicities in simple combat models, how widespread are the non-monotonicities thus created and what can be done about them? These questions are addressed in the next section.

IV. DEALING WITH CHAOS

There are two major approaches to dealing with chaos in our combat model: determine the boundaries of the induced regions of non-monotonicity with the thought of avoiding those areas in running the model, or eliminate the chaos from the model altogether. Each approach is addressed in turn in this section. Again, to isolate the non-monotonicities most directly related to the chaotic dynamics, we will be dealing with the force-ratio-only model with zero delay in the arrival of reinforcements.

BOUNDING THE NON-MONOTONIC REGIONS

Non-monotonicities occur for some input parameter values and not for others. One aspect of bounding the non-monotonic regions is to have some understanding of how extensive the non-monotonicities are in a given model. This is a difficult task in general because of the size of the input parameter space. Even in our force-ratio-only model with zero reinforcement delay, the input parameter space is 12-dimensional (for each side there are: initial troop counts, attrition coefficients, reinforcement thresholds, reinforcement block sizes, maximum limits on the number of reinforcements, and withdrawal thresholds). The following subsection gives some empirical feel for the extent of the non-monotonicities in our simple model.

Another aspect of bounding the non-monotonicities is to have analytic bounds on the maximum possible size of the non-monotonic regions. Because of the nature of our simple model, we were able to compute the maximal possible regions in which non-monotonicities could appear. This work is described in the subsection on ambiguity regions.

The Extent of Non-Monotonicities

In looking across a 12-dimensional space it would be nice to have some way of picturing as many of the dimensions at once as possible. Figure 20 shows the force-ratio-only model's behavior across two of its input dimensions (Red and Blue initial troop strengths) simultaneously. What we wanted was a quick measure of whether or not such a figure was going to demonstrate significant non-monotonicity.

We settled on two characteristics of figures such as Fig. 20 to be our indicators. The first is the maximum "span" or largest distance (measured in numbers of troops) along either the Red or Blue dimension between the first time there is a reversal in the outcomes to the last time. If the picture is monotonic, the maximum span is zero (since there is at most one reversal in the outcomes along any Red or Blue dimension). In Fig. 20 the maximum span in the Red direction is 1090 troops at 420 initial Blue troops.

On occasion there are long but "uninteresting" spans in pictures such as Fig. 20. That is, they are uninteresting because there are very few reversals in the outcomes in that span. Because we are interested in unpredictability, we are relatively more interested in pictures with both long maximum spans and many reversals throughout those spans. Our other characteristic indicator, then, is the maximum number of reversals. In Fig. 20 the maximum number of reversals in the Red direction—13—occurs at 370 Blue initial troops. These reversals were measured strictly as a transition from a Blue win to a Red win.

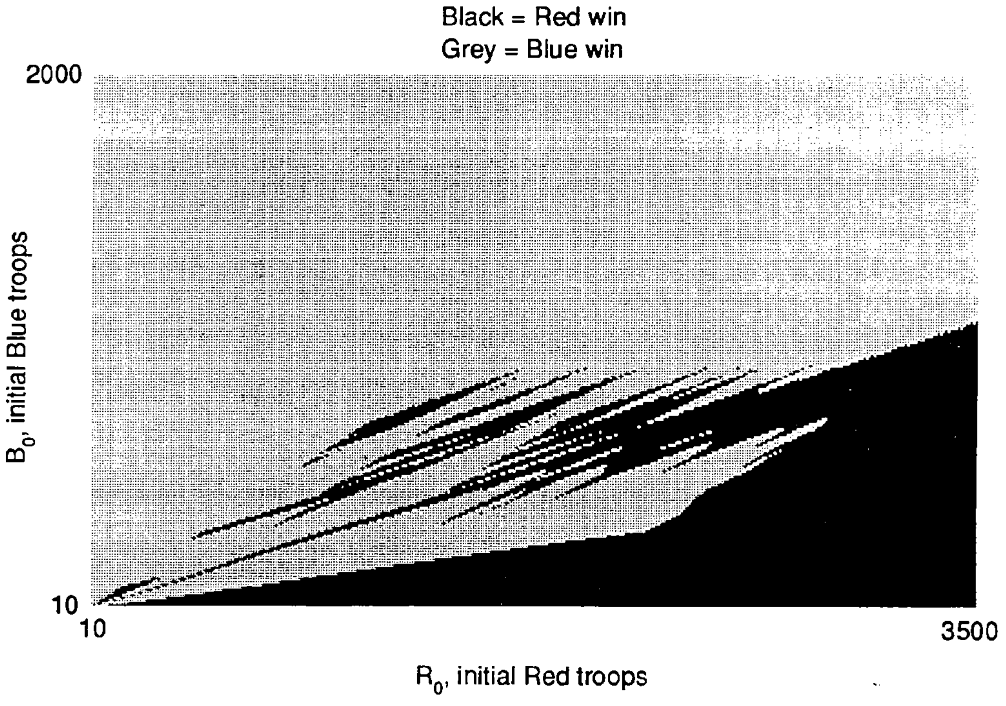


Fig. 20—Simple model: force ratio and attrition thresholds

These two (rather arbitrary) measures then allowed us to "collapse" a picture like Fig. 20 into two numbers. If we further collapse them into a single number (by multiplying them together) and ask which of these numbers exceeds, for example, 1000, we have collapsed a picture like Fig. 20 into the single measure of "serious" non-monotonicity. We can now look across other dimensions of the input space and ask for what values figures, such as those in Fig. 20, are seriously non-monotonic.

From experimentation we concluded that the input parameters figuring most importantly in the behavior of the non-monotonicities were the size of the reinforcement blocks and the total number of reinforcements available to each side. To get at the effects of these parameters we computed our measure of non-monotonicity for a fixed set of total reinforcements, but a range of block sizes, or equivalently, maximum allowable numbers of reinforcements. Table 3 is an example of one such set of matrices for both the maximum span and the maximum number of reversals in the Red direction.

Across the top of the matrices, the total number of reinforcement blocks for each side is allowed to vary. For a given total reinforcement value (1500 in this case), the number of reinforcement blocks fixes the number in each block as well (total reinforcements divided by total number of blocks (fractional troops allowed)). Along the side of the matrix, the reinforcement delay (in time steps) varies. Each entry in the top matrix, then, characterizes a picture similar to that in Fig. 20. The picture for zero reinforcement delay and 10 maximum

Table 3

SPAN AND REVERSAL CHARACTERISTICS FOR THE SIMPLE COMBAT MODEL WITH 1500 TOTAL REINFORCEMENTS AVAILABLE TO EACH SIDE

	Maximum Number of Reinforcement Blocks									
Delay	1	2	3	4	5	6	7	8	9	10
	Maximum span in Red direction									
0	1510	750	500	340	260	250	170	170	170	90
10	1530	770	510	380	300	240	160	160	160	150
20	1550	770	510	380	310	220	210	150	150	150
30	1570	780	520	390	290	260	210	150	150	150
40	1580	800	520	390	310	260	200	190	140	140
50	1610	810	530	390	320	260	230	190	130	130
60	1640	810	540	400	360	370	420	440	480	490
70	1840	1660	1380	1240	1090	980	890	830	800	780
80	2040	1640	1310	1130	1050	940	870	810	770	720
90	2140	1680	1320	1160	1020	920	870	810	760	710
10	2180	1680	1320	1160	1000	900	830	760	710	660
	Maximum win reversals in Red direction									
0	2	2	2	2	3	2	2	4	2	2
10	2	2	2	2	3	2	2	4	2	2
20	2	2	2	2	3	2	2	3	2	2
30	2	2	2	2	3	2	2	3	2	2
40	2	2	2	2	3	2	2	3	2	2
50	2	2	2	2	2	2	2	2	2	2
60	2	2	2	3	3	4	4	5	6	5
70	3	6	9	12	13	13	15	15	16	14
80	4	5	6	7	9	11	12	13	13	13
90	3	5	5	7	10	11	14	16	17	18
100	2	3	5	6	6	6	6	8	9	8

allowable reinforcement blocks, for example, would be a dull one. With a maximum span of 90 and (from the bottom matrix) a maximum of two reversals, the picture would look much like a solid wedge of Blue wins abutting a solid wedge of Red wins. On the other hand, the picture for a 70-step delay and five maximum reinforcement blocks is represented by Fig. 20.

In Table 3 the bottom matrix has been shaded for those matrix entries "seriously" nonmonotonic in that the product of the corresponding entries in the two matrices exceeded 1000. For 70 delay and five maximum reinforcement blocks the product is 1090 × 13 = 14170. By shading one of the matrices in this way, we can collapse the information in these matrices into a single shaded matrix. This then allows us to look across two more dimensions of output space as shown in Fig. 21. In this case we have examined three different values each for the total number of reinforcements available to each protagonist. Note that each "entry" in this matrix represents 121 pictures, as in Fig. 20, and each of those figures represents the results of 70,000 battles so the total number of battles represented in Fig. 21 is close to 85,000,000.

From Fig. 21 it is clear that there are regions in input parameter space in which the non-monotonicities are widespread. Continuing to explore the extent of non-monotonicities in this vein, however, is computer-intensive and time-consuming. Then, too, if exploring the

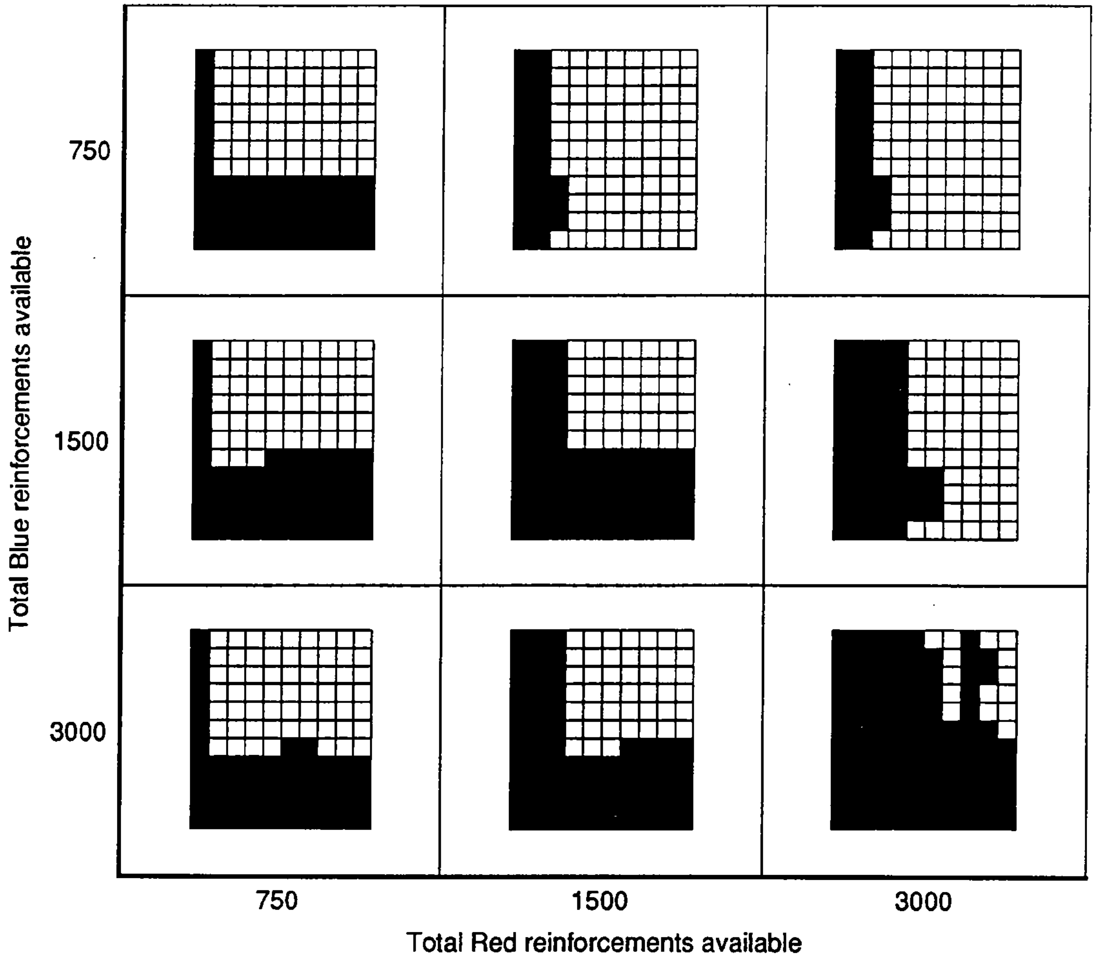


Fig. 21—The prevalence of "serious" non-monotonic behavior across six dimensions of the simple model's input space

extent of non-monotonicities through computer runs is so daunting for our simple models, the problem increases exponentially as the dimension of the input parameter space increases. That is, this is not a particularly viable way of exploring the extent of non-monotonicities in large models. Because of the particular nature of our model, there is another way of addressing this question—by looking at the regions of input space in which non-monotonicities *can* and *cannot* occur.

Ambiguity Regions

It is possible to describe, analytically, the regions outside of which non-monotonicities *cannot* occur for our simple models. For each of the force-ratio-only and attrition-only models, there is, under certain circumstances at least, an optimal reinforcement strategy for both Red and Blue. Further, because it is the same strategy for both submodels, there is an optimal reinforcement strategy under those conditions for the combined model.

It can be shown[1] that, *if one side's reinforcements are scripted*, the optimal reinforcement strategy for the other side—for both the force-ratio-only and the attrition-only models—is to have all its reinforcements immediately at the beginning of the battle. This is enough to find the bounds on the regions of Blue wins and Red wins for a given total reinforcement value. If one side (say Blue) has all its reinforcements immediately and the other side never calls for reinforcements (this is a trivial case of scripted reinforcements), then the outcome for Blue for any starting conditions will be as good as Blue can do. Note that because there are no reinforcement calls, there is no chance for non-monotonicity and the resulting picture over a range of starting conditions will have a clean demarcation line between Blue and Red wins. That line represents the edge of the area outside of which non-monotonicities cannot occur. Specifically, the area of Red wins represents an area of assured Red wins no matter what reinforcement strategies are employed by the two commanders. By reversing the situation (Red has all its reinforcements immediately and Blue never has any) another line can be generated outside of which there can be no non-monotonicities in the Blue direction. Combining the lines on the same picture defines the limits of the area, called the *ambiguity region*, within which non-monotonicities *can* occur.

Figure 22 shows the ambiguity regions drawn onto a plot that contains non-monotonicities. From the figure, the non-monotonicities for the associated reinforcement strategies cover a significant area within the ambiguity region but do not fill the region by any means. We examined several cases like this and found that the situation is typical—the non-monotonicities fill somewhat less than half of the total possible ambiguity region.

To see that the ambiguity regions are "tight" limits, we need only display a set of reinforcement strategies—based on the state of the battle—which produce non-monotonicities throughout the ambiguity region. Such a set of strategies is shown in Fig. 23. The Blue commander decides that if each side begins with an even number or an odd number, he will bring all of his reinforcements into the battle at once; otherwise he will never reinforce. The Red commander decides just the opposite. The resulting battles produce Red and Blue wins in a checkerboard pattern throughout the ambiguity region. Reversing these reinforcement heuristics will reverse the checkerboard region and the two checkerboard regions combined show that the boundary of the ambiguity region is exactly the boundary of the guaranteed-win region.

In logical terms, these are "legal" reinforcement strategies in that they are predicated on some characteristic of the evolving battle and they prescribe when reinforcements are to be introduced into the battle. In terms of military theory these reinforcement strategies are "crazy." As such, however, they emphasize the difficulty of defining what one might mean by a "realistic" reinforcement strategy. This, in turn, emphasizes the difficulty of restricting one's search for non-monotonicities to just those produced by "realistic" reinforcement strategies.

[1] See Appendix C. It is important to note that for the attrition-only model there is also a counterexample to this result for the case in which the other side's reinforcements are *not* scripted.

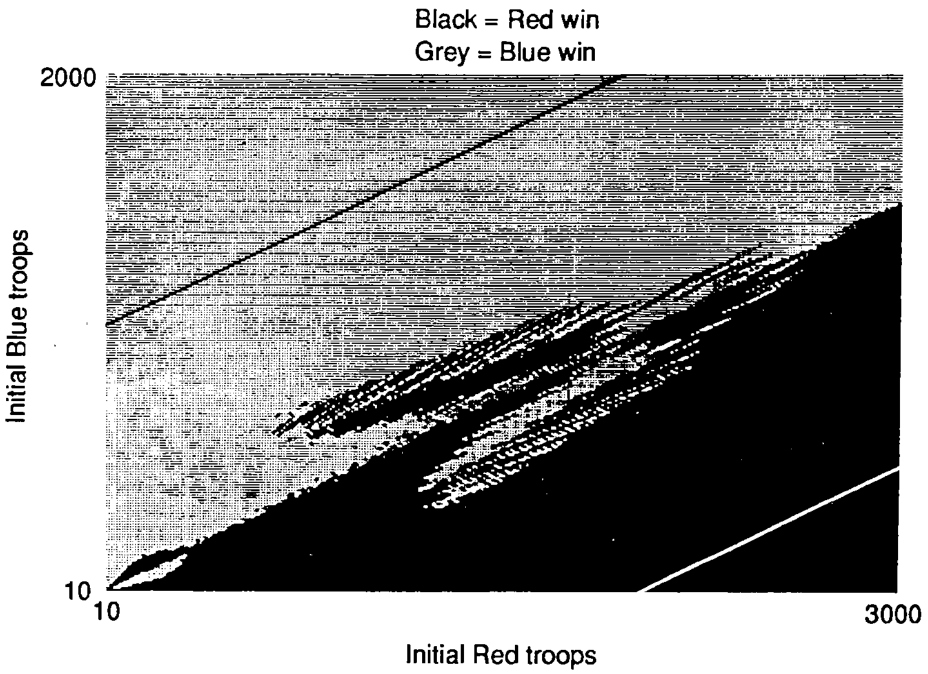


Fig. 22—Ambiguity region for one realization of the simple combat model

B_0 = Initial blue

R_0 = Initial red

	B_0 even	B_0 odd
R_0 even	Blue: All reinforcements Red: None	Blue: None Red: All reinforcements
R_0 odd	Blue: None Red: All reinforcements	Blue: All reinforcements Red: None

Fig. 23—Reinforcement heuristics for maximum non-monotonicity

For our simple models, then, the regions in which non-monotonicities can occur can be bounded. In our case, these ambiguity regions will be useful (see Sec. V) when discussing what can happen in larger models. In general, however, it will be difficult to obtain bounds on the regions of potential non-monotonicities unless optimal strategies for reinforcement can be generated and computed. In many working models this is virtually impossible.

One further note of caution is appropriate here. In regions outside the ambiguity region the model will be well behaved and monotonic. Inside the ambiguity regions, however, the absence of non-monotonicities does not necessarily imply that the model is well behaved. If the model's only measure of merit is battle outcomes, regions of monotonic behavior connote good behavior by the model. However, if internal measures of the modeled combat are also important to the analysis, regions of monotonic behavior in the battle outcome are not guarantors of non-chaotic behavior by the internal measures. In other words, monotonicity in the outcomes of a model for some subset of the ambiguity region does not guarantee stable or monotonic behavior of other modeled parameters in that region.

ELIMINATING THE NON-MONOTONIC REGIONS

Scripting the Reinforcements

If bounding the regions in which non-monotonicities might appear is a daunting task, what might be done to eliminate the non-monotonicities altogether? In Sec. III we mentioned briefly that scripting the reinforcements eliminated the non-monotonicities. It is useful in this context to understand *how* scripting the reinforcements clears up the non-monotonicities. If there are m Blue reinforcement blocks and M Red reinforcement blocks and they are scripted to come in at times $t_{B1}, t_{B2}, \ldots, t_{Bm}$ and $t_{R1}, t_{R2}, \ldots, t_{RM}$, respectively, then the force-ratio mapping can be written as a system of linear difference equations:

$$\begin{bmatrix} B_{n+1} \\ R_{n+1} \end{bmatrix} = A \left\{ \begin{bmatrix} B_n \\ R_n \end{bmatrix} + \begin{bmatrix} \delta_B(n) \\ \delta_R(n) \end{bmatrix} \right\}$$

where

$$A = \begin{bmatrix} 1 & -a_B \\ -a_R & 1 \end{bmatrix} \text{ with } a_B, a_R \text{ the attrition coefficients}$$

and

$$\delta_B(n) = \begin{cases} \text{Blue reinforcement if } n = t_{Bi} \text{ for } i = 1, \ldots, m \\ 0 \text{ otherwise} \end{cases}$$

$$\delta_R(n) = \begin{cases} \text{Red reinforcement if } n = t_{Rj} \text{ for } j = 1, \ldots, M \\ 0 \text{ otherwise} \end{cases}$$

so that

$$\begin{bmatrix} B_{n+1} \\ R_{n+1} \end{bmatrix} = A^{n+1} \begin{bmatrix} B_0 \\ R_0 \end{bmatrix} + \sum_{k=1}^{n} A^{n+1-k} \begin{bmatrix} \delta_B(k) \\ \delta_R(k) \end{bmatrix}$$

and the summation on the right side is independent of the state of the battle (that is, is dependent on time alone). But this is a linear mapping and will not exhibit non-monotonic behavior. To see this, look at two starting conditions for Blue, B_0 and $B_0' = B_0 + \Delta_B$. Then the orbit for B′ will dominate that for B in the following sense:

$$\text{for } \Delta_B > 0$$

$$B'_n - B_n = \text{first line of } \left\{ A^n \begin{bmatrix} \Delta_B \\ 0 \end{bmatrix} \right\} > 0$$

and is not dependent on the nature of the reinforcements. Scripting the reinforcements, then, turns the mapping back into a linear one. Unfortunately, this does not guarantee that the outcomes of every finite model based on this mapping are monotonic. It is still possible to devise stopping conditions that will produce non-monotonicities. For example, a stopping condition, admittedly unrealistic, that says Blue wins if it has an even number of troops on the 80th time step of the battle and will lose otherwise will produce non-monotonicities. These non-monotonicities, however, are clearly induced by the stopping conditions, not by the underlying dynamics.

Although a characterization of all stopping conditions that will produce monotonic finite models is beyond the scope of this report, it is interesting to note that in Figs. 9 and 10 the stopping conditions were based on the state of the battle. That is, in Fig. 9, a finite model with both reinforcement and stopping decisions based on the state of the battle, there were non-monotonicities. In Fig. 10, the same model with reinforcement decisions based only on the time and the stopping decisions based on the state of the battle, the model was monotonic.

Other Approaches

In general, it is clear that getting rid of the chaos and making the underlying mapping linear will seriously reduce the chances for mischief in a model. What about other means of ameliorating the effects of a chaotic underlying mapping? As long as decisions remain based on the state of the battle, there is the chance that battles from nearby starting conditions can evolve quite differently in time. Not all stopping criteria or evaluation criteria, however, produce the same amount of non-monotonicities. Table 4, for example, shows what happens in our simple force-ratio-only model for different values of the withdrawal, or stopping, force ratios. The shading represents combinations of withdrawal thresholds for Red and Blue. A shaded square means that with those withdrawal thresholds (and a given set of other input parameters), model outcomes satisfied the "serious" non-monotonicity criterion (non-monotonicity parameter > 1000, as described above). For a slightly different set of input parameters in Table 4 the shading is slightly different, but there is still no serious non-monotonicity at a Red withdrawal threshold of 1.5.

Table 4 generally reinforces the notion that the closer the withdrawal thresholds are to the reinforcement thresholds, the more non-monotonicity there is likely to be. From a military science standpoint, there may not be much leeway in the choice of some parameters

Table 4

NON-MONOTONICITY VS. WITHDRAWAL THRESHOLD FOR
TWO DIFFERENT TOTAL REINFORCEMENT VALUES
(Shading indicates "serious" non-monotonicity)

Total Blue reinforcements = Total Red reinforcements = 500

		Blue withdrawal threshold					
		5	6	7	8	9	10
Red	1.5						
withdrawal	1.75	▨					
threshold	2.0	▨	▨	▨	▨	▨	▨
	2.25	▨	▨	▨	▨	▨	▨

Total Blue reinforcements = 500, Total Red reinforcements = 1500

		Blue withdrawal threshold					
		5	6	7	8	9	10
Red	1.5						
withdrawal	1.75			▨	▨	▨	▨
threshold	2.0	▨	▨	▨	▨	▨	▨
	2.25	▨	▨	▨	▨	▨	▨

(e.g., doctrine may prescribe strict limits in some cases), but it is clear that, where possible, it may be feasible to choose input parameters that are both consistent with "reality" and ameliorate potential chaotic effects. In this case, however, finding the parameters with minimal non-monotonicities was, again, a computer-intensive process and this may not be possible with large models.

Another rule of thumb that was discussed in a different context in Sec. III is that if the size of the reinforcement blocks is very small, then the non-monotonicities appear small. Whether or not the system is chaotic, then, if the reinforcements come often and in small amounts, any observed non-monotonicities of the system are likely to be mild and well contained. Again, it may not be realistic from a military science perspective to "dribble" in the reinforcements in this way. Further, the question of how small the reinforcements need to be to ameliorate the effects of underlying chaos is again (at least at this point) a computational question rather than an analytic one. In some cases reinforcement blocks as large as 25 percent of the forces in place do not cause appreciable non-monotonicities and in others the blocks need to be much smaller. Nonetheless, making the reinforcement blocks "small enough" is a candidate amelioration for the non-monotonic effects produced by chaos in the underlying dynamics of a combat model.

We have described two heuristic solutions to the problem of non-monotonicities, apart from scripting the reinforcements, which is an analytic solution to the problem. There are undoubtedly others that depend on the particular characteristics of a combat model. One hope is that research can point to general guidelines for combat modeling that allow for "realistic" but necessarily risky nonlinear processes while minimizing those processes' potential for causing mischief in the model outcomes.

V. SOME IMPLICATIONS FOR LARGER MODELS

Thus far, our concern has been with the simple models we created to explore non-monotonicity and chaos. Although much of what we have uncovered can be generalized to other models, there is also much that is subtle and that depends on the specifics of our simple models. In Sec. VI we outline some specific research issues aimed at better understanding the connection between this work and larger, working combat models. However, our work points out two sobering generalizations that can be made about larger, more complex models and we take those up in this section. Both of them deal with adding thresholds to a model. The first deals with the behavior of the model itself and the second with the ambiguity region of a model with added thresholds.

ADDING THRESHOLDS TO A MODEL

Figure 24 shows the results of our attrition-only model for a given set of parameters. Its behavior is reasonably monotonic. The same set of parameters, but using force ratios instead of attrition for both reinforcement and withdrawal thresholds, produces Fig. 25. It, too, is reasonably monotonic. But these are just the attrition-only and force-ratio-only submodels of the model described in Sec. II. That is, if we added force ratio thresholds to the attrition-only model of Fig. 24 or attrition thresholds to the force-ratio-only model of Fig. 25, we get Fig. 20 (which is the same as Fig. 3).

In other words, here is an example where either submodel is reasonably well behaved. However, if we add the two submodels together the result is seriously non-monotonic behavior. This is a counterexample to a suggestion we have heard from several modelers that having many thresholds might "wash out" the undesired effects of any given threshold. Sometimes adding a threshold can, perhaps, improve the behavior of a model (possibly even by *moving* the undesirable behavior out of a given region), but it is clear from this example that adding a threshold can also, demonstrably, worsen the behavior of the model.

THE AMBIGUITY REGION OF COMBINED THRESHOLDS

Another hypothesis about adding thresholds is that adding a threshold might worsen a model's behavior for a given set of parameter values, but that, overall, it would be shrinking the area in which non-monotonicities might occur. In other words, perhaps it is worsening the situation in a local region, but adding the threshold shrinks the ambiguity region in which non-monotonicities might occur, and adding enough thresholds would shrink the ambiguity region to a manageable or insignificant size.

This hypothesis can also be tested using our two submodels. Fig. 26 shows the separate ambiguity regions of attrition-only and force-ratio-only submodels for a given set of input parameters. Shown also in Fig. 26 is the ambiguity region of the combined model with both force-ratio and attrition thresholds. The significance of this figure is not the exact size of the ambiguity region of the combined model, but rather the fact that it is *not* the union or the intersection of the ambiguity regions of the submodels. Many of the examples we ran in this part of the research had the union of the two constituent regions as the ambiguity region of

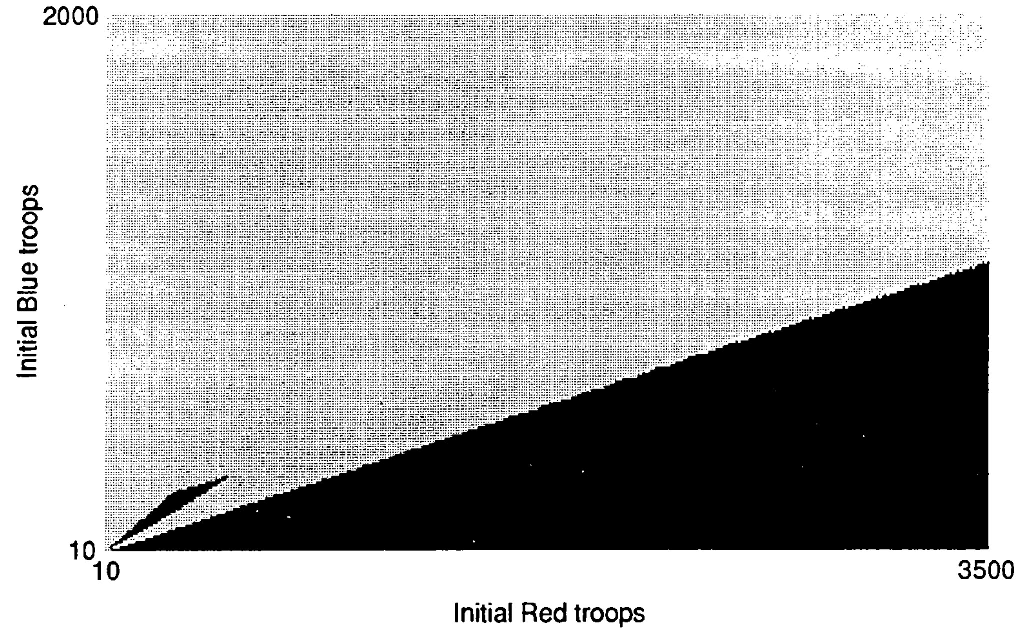


Fig. 24—Attrition thresholds only in a simple model

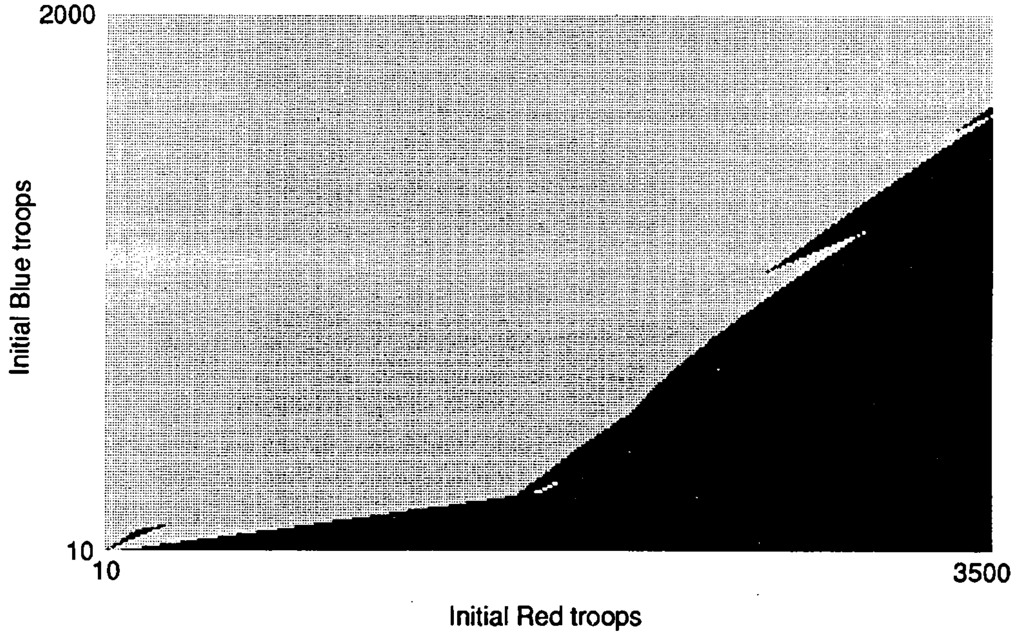


Fig. 25—Force-ratio thresholds only in a simple model

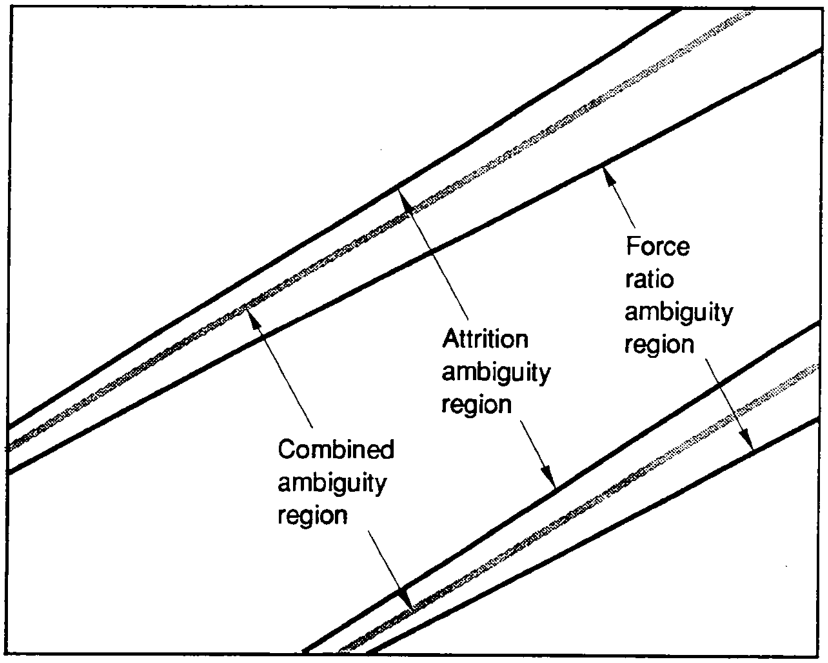


Fig. 26—Combining ambiguity regions

the combined model. However, since there are times when it is neither the union nor the intersection, the general behavior of ambiguity regions in large, complex models is likely to be complex and difficult to gauge.

IMPLICATIONS

Some care is needed in interpreting these results correctly. In logical terms, these two cases are counterexamples to the respective hypotheses that adding thresholds will *always* decrease the non-monotonicities or will *always* decrease the ambiguity region of a combat model. That is, the counterexamples incontrovertibly establish the logical falsity of the two hypotheses.

In combat modeling terms, the situation is more equivocal. If the battle in our simple model were the only battle in the war, each side would want to go all out to win that battle. If either side knew that the other side was going to throw all of its reserves into the battle at the beginning, as before, the optimal strategy would be for it, too, to throw all its reserves into the battle at the beginning. This would lead, in our simple model, to a pure attrition battle with a computable outcome and no chance for non-monotonicities. It is because it is not the only battle of the war that the commanders are marshaling their reinforcements and trying to win efficiently. The heuristics for calling up those reinforcements, however, do not (necessarily) reflect this larger reality. There can be no doubt that reinforcement heuristics based on the state of the battle can lead to non-monotonic behavior in the outcomes of that battle. It is conceivable, however, in the larger context of a two-or-more-battle war, that the overall effect of the reinforcement heuristic might produce monotonic results at the total-war level. Although the reinforcement heuristic in our simple model *implies* a two-or-more-battle war, we have concentrated on the behavior of one of the battles to the exclusion of the behavior of the total war. This suggests that an early effort in the investigation of more complex models should include looking at the results of our simple model in its implied larger context of a two-or-more-battle war.

Although neither result is very helpful in settling the case for larger, more complex models, they suggest both that it will not be surprising to find similar behavior in larger models and that it will be more difficult to analyze the exact effects. Not to put too fine a point on it, they also argue eloquently for caution both in extrapolating the results from the simple model work and in jumping to conclusions about the behavior of larger models.

VI. CONCLUSIONS AND FURTHER RESEARCH

CONCLUSIONS

For an important class of realistic combat phenomena—decisions based on the state of the battle—we have shown that modeling this behavior can introduce nonlinearities that lead to chaotic behavior in the dynamics of computerized combat models. Specifically, in a simple combat model with unlimited reinforcements, we have shown for a specific decision (when to call in battle reinforcements) based on the state of the battle (specifically, on the ratio of the opposing forces strengths) that the underlying dynamics of the model satisfy four mathematical conditions characteristic of chaotic systems. **Further, we have shown that, for a variety of battle stopping conditions, the chaotic dynamics of the underlying system give rise to non-monotonicities in model outcomes.**

We discussed a variety of sources of non-monotonicities in combat models, including: roundoff/precision problems, time-step granularity problems, and feedback effects from delayed reinforcements. We eliminated these as sources in our simple model, thus isolating the source of our observed non-monotonicities to the nonlinear functions introduced by the decisions based on the state of the battle. This cause of non-monotonicity and its chaotic nature have not been previously reported in the literature on combat modeling. Further, it is of a generally different type than the other non-monotonicity sources. Computational sources such as precision/roundoff and time-step granularity are generally well understood and correctable (albeit generally with adverse effects on the turnaround time of a model). Even feedback effects from delayed reinforcements can be ameliorated by look-ahead schemes that call for the reinforcements to arrive when the force ratio is projected to cross a threshold.

Because of the chaotic underlying dynamics, the sensitivity to initial conditions associated with the nonlinear reinforcement heuristics will appear, for example, even if the dynamical system is solved exactly. The "misbehavior" of this model is structural rather than computational, it is in the nature of the phenomenon being modeled—decisions based on the state of the battle. We have shown, further, that this structural misbehavior can lead to non-monotonicities in the outcomes of the model, and that the non-monotonic behavior can be spread over wide areas of the input parameter space.

Now, it is a separate argument to suggest that non-monotonicities in a model's outcomes are undesirable. If monotonic behavior is important for the uses of a given model, non-monotonic behavior is bad. This almost tautological statement nonetheless appears to be seriously underappreciated in the modeling community.

Consider a model that is to be used for comparative purposes, that is, to compare two or more systems, force structures, doctrines, or other alternatives. Valid comparisons require that the model outcomes accurately reflect at least the relative contributions to the battle of the competing alternatives. Suppose, instead, that the model outcomes are reflecting a combination of the contributions of the competing alternatives *and* non-monotonicities from the underlying dynamics of the combat model. For the comparisons to be valid in this case, it must be that the observed non-monotonicities reflect what would happen in real battle. This is a crucial point: Although it is conceivable that they might, unless it has been validated that they *do*, the comparisons are invalid. Since no combat model has ever been (or, we will assert, *could ever be*) validated in this sense, comparisons among competing alternatives are invalid to the extent that the model being used exhibits non-monotonic behavior in the region of the comparisons.

Models used for comparative purposes, then, must be monotonic for the comparisons to be arguably valid. How might the non-monotonicities observed in our model be eliminated? We have shown that if the reinforcement heuristic is not a function of the state of the battle (e.g., if it is "scripted" as a function of time), the nonlinearities, the chaos, and the non-monotonicities generally disappear. However, this also removes the verisimilitude of having the decision made based on the progress of the battle—as it is done in real life. However, for comparisons that do not critically depend on such verisimilitude, scripting the reinforcements in the model is a viable mitigation measure.

It is possible that there are solutions to the non-monotonicity problem that retain the verisimilitude of decisionmaking based on the state of the battle, yet do not require exhaustive validation of a model's behavior. We have shown cases in which judicious selection of model parameter values has led to generally monotonic behavior in the model. On the other hand, this required both a deeper understanding of the chaotic behavior of the underlying dynamical system and confirmation that the selected parameter values were "realistic." This is an area for further research, but it does not seem likely that such a solution, if it exists, will come easily or cheaply.

Put in other words, we have shown that a combat model with a single decision based on the state of the battle, no matter how precisely computed, can produce non-monotonic behavior in the outcomes of the model and chaotic behavior in its underlying dynamics. Working models, however, have not a single such decision, but numbers of such decisions ranging from dozens to thousands. What can be said of their behavior in this regard? We have shown that adding another decision based on some state of the model (not necessarily the same state as any existing decisions) can worsen any observed non-monotonicities, and that the area in which these non-monotonicities can occur does not necessarily shrink when another such decision is added.

In conclusion, then, in any combat model that depends for its usefulness on monotonic behavior in its outcomes, modeling combat decisions based on the state of the battle must be done very carefully. Such modeled decisions can lead to non-monotonic and chaotic behavior and the only sure ways (to date) to deal with that behavior are either to remove state dependence of the modeled decisions or to validate that the model is monotonic in the region of interest.

The importance of this conclusion is that, although carefully stated to conform with the limitations of the research, it raises the question of the extent to which the conclusions and applications of past modeling might have been tainted by the unsuspected presence of mathematical chaos and it casts a shadow over the claimed reliability of combat (and other) models even when they do not (or have not been proved to) exhibit blatantly non-monotonic behavior.

AFTERTHOUGHT

We have been discussing non-monotonic and chaotic behavior in combat models as though it were an unmitigated evil. This is clearly not necessarily the case. Historical battles have been known to hinge on very subtle effects of decisionmaking and have often been described as "chaotic" by military historians. Perhaps there is a connection between the two and it might now be possible to model such important, subtle, and nonintuitive effects. We have avoided discussing this possibility before now, because it is too easy to presume that such effects are necessarily connected.[1] Whether or not the behavior in our simple model is akin to behavior in real battle is an interesting question, but one that requires serious thought and research. It is one thing to say that both modeled decisions and real battles produce chaotic behavior and unexpected reversals in fortune, but quite another to say that one models the other. The latter is a matter for military science and is amenable to both analytic investigation and historical research.

FURTHER RESEARCH

To the question "What effect do decisions based on the state of the battle have on computerized combat models?" we have supplied only a partial and preliminary answer. We have shown, however, that such decisions *can* have unintuitive and undesirable effects on any model. We have done this by showing that they do have such effects on a simple combat model and, by counterexamples, that larger models are not immune from similar effects. In doing so, however, we have left several unanswered questions. The four described below are of greatest interest and are in order of research importance.

What Happens in Larger Models?

Our simple model showed subtle effects from the nonlinearities and the chaos it exhibited was provable because of the particularly simple form of the combat equations. We showed (but only indirectly) that such effects may also haunt larger models. What, exactly, happens in larger models? Although this may seem like a hopeless question to attempt, there are reasons to believe it is tractable. First, the chaotic and non-monotonic effects exhibited in our simple model come about because of the forced-damped nature of the underlying dynamics. The damping comes about from the attrition calculations and these tend to be simple even in large, complex models. In the TACSAGE model used for the Air Force, for example, three simple equations describe attrition. This makes them both amenable to abstraction in a simpler submodel for quick-turnaround computational investigation and to mathematical analysis. The forcing comes from reinforcements and the decisions as to when to provide them. From our research here, we have seen that we can take a simple reinforcement scheme and build on that to understand the behavior of larger models.

A rough research plan, then, would be to identify a complex, working combat model that can be abstracted, work both computationally and mathematically with the abstracted part, and then put the knowledge gained back into understanding the behavior of the larger model.

[1] It is clear that, in current models, the burden of proof of any such specific connection is on the modeler.

How Do Chaos and Stopping Conditions Relate?

Chaos is a long-term behavior of dynamical systems. Given that the dynamical system underlying a finite combat model is chaotic, we have seen that chaos is manifested in a finite realization of the model through the particular stopping conditions chosen. The relationship between the chaotic dynamics and the form of the stopping conditions that leads to non-monotonicities could stand greater elucidation, however, because, as we have shown, certain forms of stopping conditions can lead to non-monotonicities without chaos in the underlying dynamics. Moreover, chaotic underlying dynamics can lead to non-monotonicities through "no" stopping conditions (i.e., run the battle to the annihilation of one side). Understanding the exact relationship between chaotic underlying behavior and the form of the stopping conditions could be done by investigating both computationally and mathematically a variety of stopping conditions and the chaotic force-ratio mapping we investigated in this research.

Does Underlying Chaos Exacerbate Non-Monotonicities from Other Sources?

There was evidence in our research that underlying chaos magnified the extent of non-monotonicities from other sources. This would be a reasonable expectation in that sensitivity to initial conditions in the underlying equations could easily magnify a sensitivity from other sources. If this could be proved it would imply that extreme care would need to be taken to deal appropriately with all types of non-monotonicity sources in a model with underlying chaos. Again, research in this area could be done with the force-ratio mapping (with both scripted and non-scripted reinforcements) and a variety of other sources of non-monotonicity reintroduced into the model.

What Do We Mean by Chaotic?

Remember that our definition of chaos was similar to, but not the same as, definitions of chaos found elsewhere in the literature. The differences arose primarily because our simple combat model was piecewise continuous, not continuous. Characterizing chaos in such systems would be an important contribution both to understanding chaos in combat models and to the mathematics of chaos. A general definition of chaos in piecewise continuous functions could involve the generation of a mean-value theorem for piecewise continuous functions and this would be of general interest to the mathematical community. Work in this area would be pure research and would build on the research done to prove that our simple model satisfied "chaos-like" mathematical conditions.

Appendix A

COMPUTER CODE FOR THE SIMPLE COMBAT MODEL

The code presented here is for a single battle in Fig. 2 with $B_0 = 500$ and $R_0 = 1000$. It represents the fundamental unit of our investigations. All other computations used code such as this for the calculation of the outcome of an individual battle.

```
/* delayrb_r: Lanchester model with reinforcement and withdrawal
 *
 * Dewar and Gillogly:  1988-1990.
 */

char *hdr = "$Id$";

#include <stdio.h>
#include <math.h>

typedef double real;

real RI = 1000;                 /* Initial number of Red troops */
real BI = 500;                  /* Initial number of Blue troops */

real c1 = 1. / 2048.;           /* Defender's attrition rate */
real c2 = 1. / 512.;            /* Attacker's    "        "    */

real rBA = 4;                   /* Blue calls for reinforcements at 1:4 */
real rRA = 2.5;                 /* Red calls for reinforcements at 2.5:1 */

real rBW = 10;                  /* Blue withdraws at 1:10 */
real rRW = 1.5;                 /* Red withdraws at 1.5:1 */

real rBAA = .80;                /* Blue calls for more if attrition is high*/
real rRAA = .80;                /* Red  " */

real rBAW = .70;                /* Blue withdraws when down to this */
real rRAW = .70;                /* Red    "       */

int B_delay = 70;               /* Arrival time delay of reinforcements */
int R_delay = 70;               /* Arrival time delay of reinforcements */

int B_maxchunks = 5;            /* How many chunks may they use? */
int R_maxchunks = 5;            /* How many chunks may they use? */
int b_tot_reinf = 1500;         /* Total reinforcements */
int r_tot_reinf = 1500;

real a1;                        /* Reinforcement chunk size for Blue */
real a2;                        /*       "          "    "    "  Red */

int bchunks, rchunks;           /* How many chunks have Blue and Red ordered? */

real B_reinf, R_reinf;          /* Total reinforcements tossed in */

int B_step, R_step;             /* Arrival time step of reinforcements */
int B_ordered, R_ordered;       /* How many reinforcements ordered? */

real B_with_thresh, R_with_thresh, B_reinf_thresh, R_reinf_thresh;

real rc;                        /* Force ratio: Red/Blue */
real R, B;                      /* Current number of Red/Blue troops */

#define YES 1
#define NO 0
#define STOP 1
#define GO 0

#define MAXITER 200000


main()
{
      long i;

      B = BI;
      R = RI;

      B_reinf = R_reinf = 0;          /* Total reinforcements so far */
      B_ordered = R_ordered = 0;

      B_with_thresh = BI * rBAW;      /* Thresholds */
      B_reinf_thresh = BI * rBAA;

      R_with_thresh = RI * rRAW;
      R_reinf_thresh = RI * rRAA;

      B_step = R_step = 0;
      bchunks = rchunks = 0;

      a1 = (real) b_tot_reinf / B_maxchunks; /* Size of Blue blocks */
      a2 = (real) r_tot_reinf / R_maxchunks; /* Size of Red blocks */
      for (i = 0; i < MAXITER; i++)  /* ~ 1/2-hour intervals */
      {
          if (B <= 0) B = .0001;  /* Don't divide by 0 */
          rc = R/B;               /* Force ratio */

          call_reinforcements(i); /* Call for reinforcements */
          reinforce(i);           /* Do the reinforcement */
          if (withdraw(i) == STOP) break; /* See if one side quits */
          attrit();               /* Blue and Red attrition */

      }
      exit(0);
}

call_reinforcements(i)  /* see if we need to call for reinforcements */
int i;                  /* Current time */
{
      /* Blue reinforcement based on attrition */
      if (B_ordered == 0 && bchunks < B_maxchunks &&
          B < B_reinf_thresh)
      {
            B_step = i + B_delay;
            B_ordered = a1;
            bchunks++;
            printf("Blue calls for reinforcements/A  at %d.\n", i);
      }
      /* Blue reinforcement based on force ratio */
      if (B_ordered == 0 && rc > rBA && bchunks < B_maxchunks)
      {
            B_step = i + B_delay;
            B_ordered = a1;
            bchunks++;
            printf("Blue calls for reinforcements/FR at %d.\n", i);
      }
      /* Red reinforcement based on attrition */
      if (R_ordered == 0 && rchunks < R_maxchunks &&
          R < R_reinf_thresh)
      {
            R_step = i + R_delay;
            R_ordered = a2;
            rchunks++;
            printf("Red  calls for reinforcements/A  at %d.\n", i);
      }
      /* Red reinforcement based on force ratio */
      if (R_ordered == 0 && rc < rRA && rchunks < R_maxchunks)
      {
            R_step = i + R_delay;
            R_ordered = a2;
            rchunks++;
            printf(“Red  calls for reinforcements/FR at %d.\n", i);
      }
}


reinforce(i)    /* Do the reinforcement */
int i;
{
            if (B_ordered > 0 && B_step <= i)  /* B reinf arrive */
            {     B += B_ordered;
                  B_reinf += B_ordered;
                  B_ordered = 0;
                  printf(“Blue reinforcements arrive at %d.\n", i);
            }
            if (R_ordered > 0 && R_step <= i)   /* R reinf arrive */
            {     R += R_ordered;
                  R_reinf += R_ordered;
                  R_ordered = 0;
                  printf(“Red  reinforcements arrive at %d.\n", i);
            }

}


withdraw(i)     /* See if one side wants to withdraw */
int i;
{
      /* Check for withdrawal based on force ratio or attr */

      if (B < B_with_thresh || rc > rBW)
      {
            printf(“Blue withdraws at %d.\n", i);
            return STOP;
      }
      if (R < R_with_thresh || rc < rRW)
      {
            printf(“Red  withdraws at %d.\n", i);
            return STOP;
      }

      return GO;
}
```

```
attrit()        /* Throw stones at frogs in sport */
{
      real b, t;

      b = B;                     /* B and R troop strength */
      t = c1 * R;                /* Blue attrition */
      B -= t;
      t = c2 * b;                /* frogs die in earnest */
      R -= t;
}
```

Appendix B

MODEL PARAMETERS FOR TABLES AND FIGURES

The following model parameters were used to produce the figures in this report. Refer to Table 1 for an explanation of the variables.

Table B.1

MODEL PARAMETERS FOR TABLES AND FIGURES

Table	initial	final	initial	final	attrition		reinf.		reinf.		block		num of				withdraw		withdraw	
					coefficients		attrition		force/ratio		size		blks		delay		attrition		force/ratio	
or Fig.	B	B	R	R	B	R	B	R	B	R	B	R	B	R	B	R	B	R	B	R
1	839	839	1000	3500	1/2048	1/512	0.8	0.8	4.0	2.5	300	300	5	5	70	70	0.7	0.7	10.0	1.5
2	500	500	1	2000	1/2048	1/512	0.8	0.8	4.0	2.5	300	300	5	5	70	70	0.7	0.7	10.0	1.5
3	10	1000	10	2850	1/2048	1/512	0.8	0.8	4.0	2.5	300	300	5	5	70	70	0.7	0.7	10.0	1.5
4a	680	680	1001	3000	1/128	1/32	0.8	0.8	4.0	2.5	500	500	5	5	70	70	0.7	0.7	10.0	1.5
4b	680	680	1001	3000	1/256	1/64	0.8	0.8	4.0	2.5	500	500	5	5	70	70	0.7	0.7	10.0	1.5
4c	680	680	1001	3000	1/512	1/128	0.8	0.8	4.0	2.5	500	500	5	5	70	70	0.7	0.7	10.0	1.5
4d	680	680	1001	3000	1/1024	1/256	0.8	0.8	4.0	2.5	500	500	5	5	70	70	0.7	0.7	10.0	1.5
4e	680	680	1001	3000	1/2048	1/512	0.8	0.8	4.0	2.5	500	500	5	5	70	70	0.7	0.7	10.0	1.5
4f	680	680	1001	3000	1/4096	1/1024	0.8	0.8	4.0	2.5	500	500	5	5	70	70	0.7	0.7	10.0	1.5
4g	680	680	1001	3000	1/8192	1/2048	0.8	0.8	4.0	2.5	500	500	5	5	70	70	0.7	0.7	10.0	1.5
4h	680	680	1001	3000	1/16384	1/4096	0.8	0.8	4.0	2.5	500	500	5	5	70	70	0.7	0.7	10.0	1.5
5	10	2000	10	3500	1/2048	1/512	–	–	4.0	2.5	150	54	5	14	0	0	–	–	10.0	2.0
6	40	–	80	–	.005	.02	–	–	4.0	2.5	10	10	(a)	(a)	0	0	–	–	–	–
9	10	2000	10	3500	1/2048	1/512	–	–	4.0	2.5	750	750	5	14	0	0	–	–	10.0	2.0
10	10	2000	10	3500	1/2048	1/512	–	–	(b)	(b)	750	750	5	14	0	0	–	–	10.0	2.0
11	1300	–	3210	–	1/2048	1/512	–	–	4.0	2.5	1500	750	6	6	0	0	–	–	10.0	1.5
12	1300	–	3220	–	1/2048	1/512	–	–	4.0	2.5	1500	750	6	6	0	0	–	–	10.0	1.5
13	1300	–	3230	–	1/2048	1/512	–	–	4.0	2.5	1500	750	6	6	0	0	–	–	10.0	1.5
14	1300	–	3240	–	1/2048	1/512	–	–	4.0	2.5	1500	750	6	6	0	0	–	–	10.0	1.5
15	1300	–	3250	–	1/2048	1/512	–	–	4.0	2.5	1500	750	6	6	0	0	–	–	10.0	1.5
16	1300	–	3260	–	1/2048	1/512	–	–	4.0	2.5	1500	750	6	6	0	0	–	–	10.0	1.5
17	1300	–	3270	–	1/2048	1/512	–	–	4.0	2.5	1500	750	6	6	0	0	–	–	10.0	1.5
18	40	–	80	–	1/2048	1/512	–	–	4.0	2.5	428	571	–	–	0	0	–	–	–	–
19	10	2000	10	3500	1/2048	1/512	–	–	4.0	2.5	1500	750	6	6	0	0	–	–	10.0	1.5
20	10	1000	10	2850	1/2048	1/512	0.8	0.8	4.0	2.5	300	300	5	5	70	70	0.7	0.7	10.0	1.5
t3	10	1500	10	3500	1/2048	1/512	0.8	0.8	4.0	2.5	(c)	(c)	(c)	(c)	(c)	(c)	0.7	0.7	10.0	1.5
21	10	1500	10	3500	1/2048	1/512	0.8	0.8	4.0	2.5	(d)	(d)	(d)	(d)	(d)	(d)	0.7	0.7	10.0	1.5
22	10	2000	10	3000	.005	.02	0.8	0.8	4.0	2.5	200	400	5	5	6	6	0.7	0.7	10.0	1.5
t4	10	2000	10	4000	1/2048	1/512	–	–	4.0	2.5	100	100	5	5	0	0	–	–	(e)	(e)
24	10	1000	10	2850	1/2048	1/512	0.8	0.8	–	–	300	300	5	5	70	70	0.7	0.7	–	–
25	10	1000	10	2850	1/2048	1/512	–	–	4.0	2.5	300	300	5	5	70	70	–	–	10.0	1.5
26(f)	10	3500	10	4500	.005	.02	0.8	0.8	4.0	2.5	200	400	5	5	6	6	0.5	0.5	10.0	1.5

[a] No reinforcement limit.

[b] Scripted reinforcements.

[c] Total reinforcements are 1500 for Blue and Red. Blocks vary from 1 to 10. Delay 0–100 by 10s.

[d] Total reinforcements are 750, 1500, and 3000. Blocks vary from 1 to 10. Delay 0–100 by 10s.

[e] Coarse step non-monotonicities. Blue withdrawal force ratio 5–10 by 1; Red withdrawal ratio 1.5–2.25 by 0.25.

[f] Attrition-only model, force-ratio-only model, combined.

Appendix C

OPTIMAL REINFORCEMENT STRATEGIES FOR RED AND BLUE

Let B_0 and R_0 be the initial Blue and Red troops, respectively, and let $\delta_B(n)$ and $\delta_R(n)$ be the reinforcements introduced after the battle takes place at integer time step $n \geq 0$ and before the decision on withdrawal from the battle has been made. For arbitrary Blue and Red reinforcement strategies, the progress of the battle is defined by:

$$\begin{bmatrix} B_{n+1} \\ R_{n+1} \end{bmatrix} = (I - C) \left\{ \begin{bmatrix} B_n \\ R_n \end{bmatrix} + \begin{bmatrix} \delta_B(n) \\ \delta_R(n) \end{bmatrix} \right\}$$

for $n \geq 0$, where

$$I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

and

$$C = \begin{bmatrix} 0 & -a_B \\ -a_R & 0 \end{bmatrix}$$

with $0 < a_B, a_R < 1$ being the attrition coefficients for Blue and Red, respectively. By combat modeling convention if $B_{n+1} \leq 0$ we set $B_{n+1} = 0$ and similarly for R_{n+1}.

Let Δ_B be the total number of Blue reserves available so that

$$0 \leq \sum_{k=1}^{\infty} \delta_B(k) \leq \Delta_B$$

With this notation if Blue were to introduce all its reserves at the start of the battle, then $\delta_B(0) = \Delta_B$ and $\delta_B(k) = 0$ for $k > 0$.

If Red's reinforcements are scripted then one can define an optimal Blue reinforcement strategy in the following way: For the attrition-only model the optimal reinforcement strategy $\delta_B^0(k)$, $k \geq 0$ with starting conditions B_0, R_0, is one with the following properties:

$$\frac{B_n^0}{B_0 + \sum_{k=0}^{n} \delta_B^0(k)} \geq \frac{B_n}{B_0 + \sum_{k=0}^{n} \delta_B(k)}, \; n \geq 0$$

for any other reinforcement strategy $\delta_B(k)$, $k \geq 0$ and

$$\frac{R_n^0}{R_0 + \sum_{k=0}^{n} \delta_R^0(k)} \leq \frac{R_n}{R_0 + \sum_{k=0}^{n} \delta_R(k)}, \; n \geq 0$$

and since Red's reinforcements are scripted, this second condition reduces to $R_n^0 \le R_n$. For the force-ratio-only model the optimal reinforcement strategy has the property:

$$\frac{B_n^0}{R_n^0} \ge \frac{B_n}{R_n}, n \ge 0$$

for any other reinforcement strategy $\delta_B(k)$, $k \ge 0$. With these definitions we have the following two theorems:

Theorem 1. In the attrition-only model, if one side's reinforcements are scripted, the optimal reinforcement strategy for the other side is to introduce all its reserves at the start of the battle.

Proof. Without loss of generality, assume Red's reinforcements are scripted. The theorem then says that, for $n \ge 0$

$$\frac{B_n^0}{B_0 + \Delta_B} \ge \frac{B_n}{B_0 + \sum_{k=0}^{n} \delta_B(k)}$$

and $R_n^0 \le R_n$ for any reinforcement strategy $\delta_B(k)$, $k \ge 0$.

For any reinforcement strategy:

$$\begin{bmatrix} B_n \\ R_n \end{bmatrix} = (I - C) \left\{ \begin{bmatrix} B_{n-1} \\ R_{n-1} \end{bmatrix} + \begin{bmatrix} \delta_B(n-1) \\ \delta_R(n-1) \end{bmatrix} \right\}$$

$$= (I - C)^n \begin{bmatrix} B_0 \\ R_0 \end{bmatrix} + \sum_{k=1}^{n} (I - C)^k \begin{bmatrix} \delta_B(n-k) \\ \delta_R(n-k) \end{bmatrix}$$

and since I and C commute

$$= \sum_{i=0}^{n} \binom{n}{i} (-1)^i C^i \begin{bmatrix} B_0 \\ R_0 \end{bmatrix} + \sum_{k=1}^{n} \sum_{i=0}^{k} \binom{k}{i} (-1)^i C^i \begin{bmatrix} \delta_B(n-k) \\ \delta_R(n-k) \end{bmatrix}$$

but now

$$C^{2p} = \begin{bmatrix} (a_B a_R)^p & 0 \\ 0 & (a_B a_R)^p \end{bmatrix} = (a_B a_R)^p I$$

and

$$C^{2p+1} = (a_B a_R)^p C$$

so that

$$\begin{bmatrix} B_n \\ R_n \end{bmatrix} = \left\{ \sum_{k \text{ even}} \binom{n}{k} \left[\sqrt{a_B a_R}\right]^k I - \sum_{k \text{ odd}} \binom{n}{k} \left[\sqrt{a_B a_R}\right]^k \begin{bmatrix} 0 & \sqrt{\frac{a_B}{a_R}} \\ \sqrt{\frac{a_R}{a_B}} & 0 \end{bmatrix} \right\} \begin{bmatrix} B_0 \\ R_0 \end{bmatrix}$$

$$+ \sum_{k=1}^{n} \left\{ \sum_{i \text{ even}} \binom{k}{i} \left[\sqrt{a_B a_R}\right]^i I - \sum_{i \text{ odd}} \binom{k}{i} \left[\sqrt{a_B a_R}\right]^i \begin{bmatrix} 0 & \sqrt{\frac{a_B}{a_R}} \\ \sqrt{\frac{a_R}{a_B}} & 0 \end{bmatrix} \right\} \begin{bmatrix} \delta_B(n-k) \\ \delta_R(n-k) \end{bmatrix}$$

and

$$B_n = \sum_{k \text{ even}} \binom{n}{k} \left[\sqrt{a_B a_R}\right]^k B_0 - \sum_{k \text{ odd}} \binom{n}{k} \left[\sqrt{a_B a_R}\right]^k \sqrt{\frac{a_B}{a_R}}\, R_0$$

$$+ \sum_{k=1}^{n} \left\{ \sum_{i \text{ even}} \binom{k}{i} \left[\sqrt{a_B a_R}\right]^i \delta_B(n-k) - \sum_{i \text{ odd}} \binom{k}{i} \left[\sqrt{a_B a_R}\right]^i \sqrt{\frac{a_B}{a_R}}\, \delta_R(n-k) \right\}$$

For convenience, define

$$F_k = \sum_{i \text{ even}} \binom{k}{i} \left[\sqrt{a_B a_R}\right]^i$$

and

$$G_k = \sum_{i \text{ odd}} \binom{k}{i} \left[\sqrt{a_B a_R}\right]^i \sqrt{\frac{a_B}{a_R}}$$

so that

$$B_n = F_n B_0 - G_n R_0 + \sum_{k=1}^{n} F_k \delta_B(n-k) - \sum_{k=1}^{n} G_k \delta_R(n-k)$$

Note that

$$1 \leq F_k = 1 + \sum_{i>1,\text{even}} \binom{k}{i} \left[\sqrt{a_B a_R}\right]^i,$$

that

$$F_{k+1} = \sum_{\text{i even}} \binom{k+1}{i} \left[\sqrt{a_B a_R}\right]^i \geq F_k \text{ for } k \geq 0$$

and that

$$G_k = \sum_{\text{k odd}} \binom{k}{i} \left[\sqrt{a_B a_R}\right]^i \sqrt{\frac{a_B}{a_R}} \geq 0 \text{ for } k \geq 0$$

Now the strategy we want to prove optimal has $\delta_B^0(0) = \Delta_B$ and $\delta_B^0(k) = 0$ for $k \geq 1$ and its time series is:

$$\begin{bmatrix} B_n^0 \\ R_n^0 \end{bmatrix} = (I - C)^n \begin{bmatrix} B_0 + \Delta_B \\ R_0 \end{bmatrix} + \sum_{k=1}^{n} (I - C)^k \begin{bmatrix} 0 \\ \delta_R(n - k) \end{bmatrix}$$

and by calculations similar to those above:

$$B_n^0 = F_n(B^0 + \Delta_B) - G_n R_0 - \sum_{k=1}^{n} G_k \delta_R(n - k)$$

Since the Red reinforcements are scripted the terms containing δ_R $(n - k)$ are the same in the expressions for B_n and B_n^0. For convenience, again, define

$$R = G_n R_0 + \sum_{k=1}^{n} G_k \delta_R(n - k)$$

and note $R \geq 0$ so that

$$B_n = F_n B_0 + \sum_{k=1}^{n} F_k \delta_B(n - k) - R$$

and

$$B_n^0 = F_n(B_0 + \Delta_B) - R$$

We will first establish that

$$\frac{B_n^0}{B_0 + \Delta_B} \geq \frac{B_n}{B_0 + \sum_{k=1}^{n} \delta_B(n - k)} \text{ for all } n \geq 0 \tag{1}$$

This is the same as showing that

$$B_n^0 \left[B_0 + \sum_{k=1}^{n} \delta_B(n - k) \right] \geq B_n(B_0 + \Delta_B)$$

From above, this is the same as

$$(F_nB_0 + F_n\Delta_B - R)\left[B_0 + \sum_{k=1}^{n} \delta_B(n-k)\right] \geq \left[F_nB_0 + \sum_{k=1}^{n} F_k\delta_B(n-k) - R\right](B_0 + \Delta_B) \qquad (2)$$

Multiplying out and cancelling terms, this implies that

$$B_n^0 \sum_{k=1}^{n} \delta_B(n-k) \geq (B_0 + \Delta_B) \sum_{k=1}^{n} F_k\delta_B(n-k) - R\Delta_B$$

Now look at

$$\sum_{k=1}^{n} F_k\delta_B(n-k)$$

Since $F_k \geq 1$ and $\delta_B(n-k) \geq 0$,

$$\sum_{k=1}^{n} F_k\delta_B(n-k) \geq 0$$

and if the equality holds, then

$$\sum_{k=1}^{n} \delta_B(n-k) = 0$$

Further, since $F_{k+1} \geq F_k$ for $k \geq 0$,

$$\sum_{k=1}^{n} F_k\delta_B(n-k) \leq F_n\sum_{k=1}^{n} \delta_B(n-k) \leq F_n\Delta_B$$

Now, establishing the inequality in (2) at the maximum and minimum of

$$\sum_{k=1}^{n} F_k\delta_B(n-k)$$

suffices to establish the inequality (1). But if

$$\sum_{k=1}^{n} F_k\delta_B(n-k) = 0$$

then (2) becomes $0 \geq -R\Delta_B$ which is true. Also, if

$$\sum_{k=1}^{n} F_k\delta_B(n-k) = F_n\Delta_B$$

then

$$\sum_{k=1}^{n} \delta_B(n-k) = \Delta_B$$

and (2) becomes

$$(F_n(B_0 + \Delta_B) - R)\Delta_B \geq (B_0 + \Delta_B)F_n\Delta_B - R\Delta_B$$

which is also true so that (1) is established.

It remains to show that $R_n^0 \leq R_n$ for $n \geq 0$.

$$R_n = \sum_{k\ \text{even}} \binom{n}{k} \left[\sqrt{a_B a_R}\right]^k R_0 - \sum_{k\ \text{odd}} \binom{n}{k} \left[\sqrt{a_B a_R}\right]^k \sqrt{\frac{a_R}{a_B}}\, B_0$$

$$+ \sum_{k=1}^{n} \left\{ \sum_{i\ \text{even}} \binom{k}{i} \left[\sqrt{a_B a_R}\right]^i \delta_R(n-k) - \sum_{i\ \text{odd}} \binom{k}{i} \left[\sqrt{a_B a_R}\right]^i \sqrt{\frac{a_R}{a_B}}\, \delta_B(n-k) \right\}$$

and

$$R_n^0 = \sum_{k\ \text{even}} \binom{n}{k} \left[\sqrt{a_B a_R}\right]^k R_0 - \sum_{k\ \text{odd}} \binom{n}{k} \left[\sqrt{a_B a_R}\right]^k \sqrt{\frac{a_R}{a_B}}\, (B_0 + \Delta_B)$$

$$+ \sum_{k=1}^{n} \sum_{i\ \text{even}} \binom{k}{i} \left[\sqrt{a_B a_R}\right]^i \delta_R(n-k)$$

so that if

$$\sum_{k\ \text{odd}} \binom{n}{k} \left[\sqrt{a_B a_R}\right]^k \sqrt{\frac{a_B}{a_R}}\, \Delta_B \geq \sum_{k=1}^{n} \sum_{i\ \text{odd}} \binom{k}{i} \left[\sqrt{a_B a_R}\right]^i \sqrt{\frac{a_R}{a_B}}\, \delta_B(n-k)$$

the theorem is established.

That is, we need to show

$$G_n\Delta_B \geq \sum_{k=1}^{n} G_k\, \delta_B(n-k)$$

but $G_{k+1} \geq G_k$ for $k \geq 0$ from earlier, so that

$$\sum_{k=1}^{n} G_k\delta_B(n-k) \leq G_n \sum_{k=1}^{n} \delta_B(n-k) \leq G_n\Delta_B$$

and the theorem is established. q.e.d.

A stronger theorem can be established. Namely, if B uses its reserves faster than B′ in the sense that

$$\sum_{k=1}^{n} \delta_B(k) \geq \sum_{k=1}^{n} \delta_{B'}(k)$$

for every n then $B_n \geq B'_n$ and $R_n \leq R'_n$ for all n.

To see that the theorem is *not* true for unscripted Red reinforcements consider a simple counterexample with two time steps in two battles—one with an arbitrary reinforcement strategy for Blue and the other with all Blue reserves thrown in at the outset. For simplicity, assume in the former that all reserves are used up. That is,

$$\delta_B(1) + \delta_B(2) = \Delta_B$$

Assuming neither side has withdrawn:

$$\begin{bmatrix} B_2 \\ R_2 \end{bmatrix} = A^2 \begin{bmatrix} B^0 \\ R^0 \end{bmatrix} + A \begin{bmatrix} \delta_B(1) \\ \delta_R(1) \end{bmatrix} + \begin{bmatrix} \delta_B(2) \\ \delta_R(2) \end{bmatrix} \text{ with } A = \begin{bmatrix} 1 & -a_B \\ -a_R & 1 \end{bmatrix}$$

$$= \begin{bmatrix} (1 + a_B a_R)B_0 - 2a_B R_0 + \delta_B(1) - a_B \delta_R(1) + \delta_B(2) \\ -2a_R B_0 + (1 + a_B a_R)R_0 - a_R \delta_B(1) + \delta_R(1) + \delta_R(2) \end{bmatrix}$$

Similarly

$$\begin{bmatrix} B_2^0 \\ R_2^0 \end{bmatrix} = \begin{bmatrix} (1 + a_B a_R)(B_0 + \Delta_B) - 2a_B R_0 - a_B \delta_{R^0}(1) \\ -2a_R(B_0 + \Delta_B) + (1 + a_B a_R)R_0 + \delta_{R^0}(1) + \delta_{R^0}(2) \end{bmatrix}$$

so that

$$\frac{B_2^0}{B_0 + \Delta_B} - \frac{B_2}{B_0 + \delta_B(1) + \delta_B(2)} = \frac{a_B a_R \Delta_B - a_B(\delta_{R^0}(1) - \delta_R(1))}{B_0 + \Delta_B}$$

$$= \frac{a_B}{B_0 + \Delta_B} \left[a_R \Delta_B - (\delta_{R^0}(1) - \delta_R(1)) \right]$$

but this is ≥ 0 only if $a_R \Delta_B \geq (\delta_{R^0}(1) - \delta_R(1))$. For any reasonable reinforcement strategy $\delta_{R^0}(1) \geq \delta_R(1)$ since $R_1^0 \leq R_1$ so it will be possible to choose values of a_R, Δ_B, $\delta_{R^0}(1)$, and $\delta_R(1)$ to violate the inequality. This very simple example demonstrates the subtlety of the interplay of the timings on reinforcements. A large reinforcement by Blue can "trigger" a large one by Red where a smaller one might not.

Theorem 2. In the force-ratio-only model, if one side's reinforcements are scripted, the optimal reinforcement strategy for the other side is to introduce all its reserves at the start of the battle.

Proof. Assuming Red's reinforcements are scripted and $R_n, R_n^0 > 0$ for all n, it will suffice to show that

$$\frac{B_n^0}{R_n^0} \geq \frac{B_n}{R_n} \text{ or } B_n^0 R_n \geq B_n R_n^0 \text{ for all n}$$

Using the terminology in the proof of Theorem 1,

$$B_n^0 = F_n(B_0 + \Delta_B) - G_nR_0 - \sum_{k=1}^{n} G_k\delta_R(n-k)$$

and

$$B_n = F_nB_0 - G_nR_0 + \sum_{k=1}^{n} F_k\delta_B(n-k) - \sum_{k=1}^{n} G_k\delta_R(n-k)$$

Since Red's reinforcements are scripted, the last term in the two equations is the same so

$$B_n^0 - B_n = F_n\Delta_B - \sum_{k=1}^{n} F_k\delta_B(n-k) \geq 0 \text{ for all n}$$

from the proof of Theorem 1. Also from that proof, $R_n^0 \leq R_n$ for all n, and since all four quantities are nonnegative, $B_n^0R_n \geq B_nR_n^0$ for all n. q.e.d.

BIBLIOGRAPHY

Azmy, Y. Y., and V. Protopopescu, *Two Dimensional Maps Generated by Competitive Systems,* ORNL/TM-11026, Oak Ridge National Laboratory, 1989.

Barnsley, Michael, *Fractals Everywhere,* Academic Press, California, 1988.

Benzinger, Harold E., Scott A. Burns, and Julian I. Palmore, "Chaotic Complex Dynamics and Newton's Method," *Physics Letters A,* Vol. 119, No. 9, pp. 441–446, 1987.

Burns, Scott A., and Julian I. Palmore, "The Newton Transform: An Operational Method for Constructing Integrals of Dynamical Systems," *Physica D,* Vol. 37, pp. 83–90, 1989.

Campbell, David K., and Gottfried Mayer-Kress, *Nonlinearity and Chaos: The Challenges and Limitations of Mathematical Modeling of Environmental and Socio-Political Issues,* Center for Nonlinear Studies, Los Alamos National Laboratory, 1989.

Cipra, Barry A., "Computer-Drawn Pictures Stalk the Wild Trajectory," *Science,* Vol. 241, pp. 1162–1163, 1988.

Cipra, Barry A., "Image Capture by Computer," *Science,* Vol. 243, pp. 1288–1289, 1989.

Collet, Pierre, and Jean-Pierre Eckmann, *Iterated Maps on the Interval as Dynamical Systems,* Birkhauser, Basel, 1980.

Crutchfield, James P., "Quantifying Spatial Chaos," *Research and Development,* pp. 13–17, 21 November 1989.

Culioli, J.-C., and V. Protopopescu, *Bifurcating Optimization Algorithms and Their Possible Application,* ORNL/TM-10976, Oak Ridge National Laboratory, 1988.

Cvitanovic, Predrag, *Universality in Chaos,* Adam Hilger, Bristol, 1984.

Derrida, B., and R. Meir, "Chaotic Behavior of a Layered Neural Network," *Physical Review A,* Vol. 38, No. 6, 1988.

Devaney, Robert L., *An Introduction to Chaotic Dyamical Systems,* Addison-Wesley, New York, 1989.

Dockery, J. T., and A.E.R. Woodcock, "Models of Combat with Embedded C^{2}II: Catastrophe Theory and Chaotic Behavior," *International CIS Journal,* Vol. 2, No. 4, pp. 17–51, 1988.

Eckmann, J. P., and D. Ruelle, "Ergodic Theory of Chaos and Strange Attractors," *Reviewers of Modern Physics,* Vol. 57, No. 3, Part 1, pp. 617–656, 1985.

Froehling, Harold, J. P. Crutchfield, Doyne Farmer, N. H. Packard, and Rob Shaw, "On Determining the Dimension of Chaotic Flows," *Physica 3D,* pp. 605–617, 1981.

Gallavotti, Giovanni, and Donald S. Ornstein, "Billiards and Bernoulli Schemes," *Communications in Mathematical Physics,* Vol. 38, pp. 83–101, 1974.

Geweke, John, *Inference and Forecasting for Chaotic Nonlinear Time Series,* Duke University, North Carolina, 1989.

Gleick, James, *Chaos, Making a New Science,* Viking Press, New York, 1987.

Gordon, Theodore J., and David Greenspan, "Chaos and Fractals: New Tools for Technological and Social Forecasting," *Technological Forecasting and Social Change,* Vol. 34, 1988.

Grossman, Siegfried, and Gottfried Mayer-Kress, "The Role of Chaotic Dynamics in Arms Race Models," *Nature,* Vol. 337, pp. 701–704, 1989.

Guckenheimer, John, and Philip Holmes, *Nonlinear Oscillations, Dynamical Systems, and Bifucations of Vector Fields,* Springer-Verlag, New York, 1986.

Hoffnagle, J., et al., "Order-Chaos Transition of Two Trapped Ions," *Physical Review Letters,* Vol. 61, 255, 1988.

Homer-Dixon, Thomas F., "A Common Misapplication of the Lanchester Square Law," *International Security,* Vol. 12, No. 1, pp. 135–139, 1987.

Hughes, Wayne P., Jr., *Military Modeling,* Military Operations Research Society, 1989.

Jensen, Roderick V., "Classical Chaos," *American Scientist,* Vol. 75, pp. 168–181, 1987.

Jensen, R. V., and R. Urban, "Chaotic Price Behavior in a Nonlinear Cobweb Model," *Econ. Lett.,* Vol. 15, 1984.

Kan, Ittai, "Strange Attractors of Uniform Flows," *Transactions of the American Mathematical Society,* Vol. 293, No. 1, pp. 135–159, 1986.

Kloeden, P. E., "Chaotic Difference Equations in R," *Journal Australian Mathematical Society,* Vol. 31, pp. 217–225, 1981.

Lasota, A., and M. C. Mackey, *Probabilistic Properties of Deterministic Systems,* Cambridge University Press, New York, 1985.

Lepingwell, John W. R., "The Laws of Combat?" *International Security,* pp. 89–133.

Li, T.-Y., and J. A. Yorke, "Period Three Implies Chaos," *Amer. Math. Monthly,* Vol. 82, pp. 985–992, 1975.

Lichtenberg, A. J., and M. A. Lieberman, *Regular and Stochastic Motion,* Springer-Verlag, New York, 1982.

Lorenz, E., "Deterministic Nonperiodic Flow," *J. Atmos. Sci.,* Vol. 20, pp. 130–141, 1963.

Marotto, Frederick R., "Snap-Back Repellers Imply Chaos in R," *Journal of Mathematical Analysis and Applications,* Vol. 63, pp. 199–223, 1978.

May, R. M., "Simple Mathematical Models with Very Complicated Dynamics," *Nature,* Vol. 261, 1976.

May, Robert, "The Chaotic Rhythms of Life," *New Scientist,* pp. 37–41, 18 November 1989.

Mayer-Kress, Gottfried, *A Nonlinear Dynamical Systems Approach to International Security,* Proceedings of AAAS Conference, San Francisco, January 1989.

McCauley, Joseph L., Jr., and Julian I. Palmore, "Computable Chaotic Orbits," *Physics Letters A,* Vol. 115, No. 9, 1986.

Moore, Cristopher, "Unpredictability and Undecidability in Dynamical Systems," *Physical Review Letters,* Vol. 64, No. 20, pp. 2354–2357, 1990.

Mullin, Tom, "Turbulent Times for Fluids," *New Scientist,* pp. 52–55, 11 November 1989.

Murray, Carl, "Is the Solar System Stable?" *New Scientist,* pp. 60–63, 25 November 1989.

Nash, David H., and Brian P. McBride, *Fractals, Chaos Theory and Stock Averages,* Eighth International Symposium on Forecasting, Amsterdam, June 1988.

Ornstein, D. S., "Ergodic Theory, Randomness, and 'Chaos'," *Science,* Vol. 243, pp. 182–187, 1989.

Palmer, Tim, "A Weather Eye on Unpredictability," *New Scientist,* pp. 56–59, 11 November 1989.

Palmore, Julian I., and Joseph L. McCauley, "Shadowing by Computable Chaotic Orbits," *Physics Letters A,* Vol. 122, No. 8, pp. 399–402, 1987.

Palmore, Julian I., and C. Herring, "Computer Arithmetic, Chaos and Fractals," *Physica D,* Vol. 42, pp. 99–110, 1990.

Palmore, Julian I., Scott A. Burns, and Harold E. Benzinger, "Ecology Models and Newton Vector Fields," *Mathematical Biosciences,* Vol. 90, pp. 221–232, 1988.

Percival, Ian, "Chaos: A Science for the Real World," *New Scientist,* pp. 42–47, 21 October 1989.

Perkers, Kelly, Dan Burnett, and Stavros Busenberg, *Theory of Chaos Applied to Combat Models,* Mathematics Clinic Report, Harvey Mudd College, May 1990.

Pool, Robert, "Chaos Theory: How Big an Adventure?" *Science,* Vol. 245, pp. 26–28, 1989a.

Pool, Robert, "Ecologists Flirt with Chaos," *Science,* Vol. 243, pp. 310–313, 1989b.

Pool, Robert, "Is It Chaos, or Is It Just Noise?" *Science,* Vol. 243, pp. 25–28, 1989c.

Pool, Robert, "Is It Healthy to Be Chaotic?" *Science,* Vol. 243, pp. 604–607, 1989d.

Pool, Robert, "Is Something Strange About the Weather?" *Science,* Vol. 243, pp. 1290–1293, 1989e.

Pool, Robert, "Seeing Chaos in a Simple System," *Research News,* Vol. 241, pp. 787–788, 1988.

Protopopescu, V., R. T. Santoro, and J. Dockery, "Combat Modeling with Partial Differential Equations," *European Journal of Operational Research,* Vol. 38, pp. 178–183, 1989.

Protopopescu, V., R. T. Santoro, R. Cox, and P. Rusu, *Combat Modeling with Partial Differential Equations: The Bidimensional Case,* ORNL/TM-11343, Oak Ridge National Laboratory, 1990.

Protopopescu, V., R. T. Santoro, and Y. Y. Azmy, *PDE Models for Combat: Potential Applications to LIC,* Oak Ridge National Laboratory, forthcoming.

Rehmus, Paul, and John Ross, "Periodically Perturbed Chemical Systems," *Oscillations and Traveling Waves in Chemical Systems,* John Wiley, New York, pp. 287–332, 1985.

Sandmeyer, R. S., "SIMTECH Project: Application of Supercomputers to Division/Corps Level Combat Simulation," Presentation at VIC Model Users Group meeting, White Sands Missile Range, New Mexico, October 6, 1988.

Saperstein, Alvin M., "Chaos—A Model for the Outbreak of War," *Nature,* Vol. 309, pp. 303–305, 1984.

Saperstein, Alvin M., "SDI a Model for Chaos," *Bulletin of the Atomic Scientists,* pp. 24–27, October 1988.

Saperstein, Alvin M., and Gottfried Mayer-Kress, "A Nonlinear Dynamic Model of the Impact of SDI on the Arms Race," *J. Conflict Resolution,* Vol. 32, pp. 636–670, 1988.

Saperstein, Alvin M., and Gottfried Mayer-Kress, "Chaos Versus Predictability in Formulating National Strategic Security Policy," *Am. J. Phys.,* Vol. 57, No. 3, pp. 217–223, 1989.

Scollan, D. F., Y. Y. Azmy, and V. Protopopescu, *Nonlinear Maps with Competitive Interactions: Fixed-Points, Bifurcations, and Chaotic Attractors,* ORNL/TM-11315, Oak Ridge National Laboratory, 1989.

Scott, Stephen, "Clocks and Chaos in Chemistry," *New Scientist,* pp. 53–59, 2 December 1989.

Sholl, D., G. D. Doolen, and Y. C. Lee, *Analytic Solutions for Lanchester Equations,* Center for Nonlinear Studies, Los Alamos National Laboratory, forthcoming.

Simon, Herbert A., "Prediction and Prescription in Systems Modeling," *Operations Research,* Vol. 38, No. 1, pp. 7–14, 1990.

Stewart, Ian, "Portraits of Chaos," *New Scientist,* pp. 42–47, 4 November 1989.

Stewart, Ian, *Does God Play Dice? The Mathematics of Chaos,* Basil Blackwell Ltd., Oxford, 1989b.

Taylor, James G., *Force-on-Force Attrition Modeling,* Military Applications Section of the Operations Research Society of America, Arlington, Virginia, 1980.

Taylor, James G., *Lanchester Models of Warfare,* Military Applications Section of the Operations Research Society of America, Arlington, Virginia, 1983.

Thompson, J.M.T., and H. B. Stewart, *Nonlinear Dynamics and Chaos,* John Wiley, New York, 1987.

Vivaldi, Franco, “An Experiment with Mathematics,” *New Scientist,* pp. 46–49, 28 October 1989.

Walt, Stephen M., “The Search for a Science of Strategy,” *International Security* Vol. 12, No. 1, pp. 140–165, 1987.

RAND/R-3995-RC