EXPLORATORY MODELING FOR POLICY ANALYSIS

STEVE BANKES
RAND, Santa Monica, California

(Received February 1992; revision received January 1993; accepted March 1993)

Exploratory modeling is using computational experiments to assist in reasoning about systems where there is significant uncertainty. While frequently confused with the use of models to consolidate knowledge into a package that is used to predict system behavior, exploratory modeling is a very different kind of use, requiring a different methodology for model development. This paper distinguishes these two broad classes of model use, describes some of the approaches used in exploratory modeling, and suggests some technological innovations needed to facilitate it.

We are arguably in a golden age of computer modeling for policy analysis. Computational power is plentiful. A plethora of models are in use or under development, from small spreadsheet models developed for individual use to large multipurpose supercomputer-based simulations. Computer modeling has become central to nearly every kind of policy concern, from military procurement, to economic forecasting, to studies of global climate change.

If computer modeling were fulfilling its promise, this would also be a golden age for policy analysis, with the computational power of computer models supporting significantly better decision making than was possible thirty years ago. However, the principal result of the increasing use of computer models seems to be not a marked improvement in the quality of decision making, but rather a growing sensitivity to the shortcomings of models. There is an extensive and often insightful literature discussing and documenting the pitfalls in the uses of models and suggesting ways to avoid them (Quade 1980, 1985, Meadows, Richardson and Bruckmann 1982, Raiffa 1982, Goeller 1984, Meadows and Robinson 1985, Miser and Quade 1988). Nevertheless, confusion over the appropriate uses for models persists, and criticisms of policy studies based on large simulations are becoming increasingly common (Stockfisch 1975, Freedman 1981, Schrage 1989, Hodges 1991).

This paper argues that many of the problems of computer modeling generally, and of the use of large simulations most particularly, result from a fundamental confusion between two very different uses for computer models. I call these two uses consolidative and exploratory modeling.

Building a model by consolidating known facts into a single package and then using it as a surrogate for the actual system, which I call *consolidative modeling*, is in many ways the standard approach to model development and use. Where successful, it is a powerful technique for understanding the behavior of complex systems. Unfortunately, the consolidative approach is not always possible.

When insufficient knowledge or unresolvable uncertainties preclude building a surrogate for the target system, modelers must make guesses at details and mechanisms. While the resulting model cannot be taken as a reliable image of the target system, it does provide a computational experiment that reveals how the world would behave if the various guesses were correct. *Exploratory modeling* is the use of series of such computational experiments to explore the implications of varying assumptions and hypotheses.

Enormous increases in the availability of computational power in the past few years have made aggressive exploratory use of complex computer models possible for the first time. We now live in an era in which computational experiments are commonplace in many of the sciences (Strauss 1974, Campbell et al. 1985, Rose and Dobson 1985, Anderson 1988, Lipton, Marr and Welsh 1989). Exploratory use

Subject classifications: Computers/computer science: system design. Philosophy of modeling: exploratory modeling.
Area of review: OR FORUM.

0030-364X/93/4103-0435 $01.25

involves guessing the details of systems for which there are no data, such as the behavior of subatomic particles at very high energies or the spatiotemporal activation patterns of large numbers of neurons in the brain. The implications of these guesses can be computed, allowing the computer-assisted researcher to look for interesting guesses. As computing becomes easier than performing experiments, this style of research becomes increasingly attractive. The difficulties of modeling once dictated gathering as much data as possible before modeling, but now it may often prove more expedient to do extensive modeling to constrain the number of actual experiments that need to be done to answer a question.

Exploratory approaches to modeling are especially favored whenever critical information is absolutely unavailable. As policy studies must contend with significant unresolvable uncertainties, modeling for policy analysis often must have an exploratory rationale. Unfortunately, the exploratory nature of most model-based policy research is often unrecognized, and the commonplace adoption of methodology drawn from the consolidative use of models frequently results in questionable strategies for model development and use.

In this paper, I will describe the consolidative and exploratory modeling paradigms and argue that they are distinctly different though frequently confused. I will further argue that the confusion between these uses for models is the cause of many, perhaps most, of the problems surrounding computer modeling for policy analysis. I will then describe in greater detail how exploratory modeling can be used to support policy studies. Finally, I will suggest that this approach, together with the growing abundance of computational power, implies significant opportunities for improving the methodology and the technology of computer modeling.

THE PROBLEM: PRETENDING TO DO WHAT CAN'T BE DONE

Consider the following fictional account.

In 199x, the Joint Chiefs of Staff decide they need to develop improved means for making (and defending) procurement decisions, designing force structures, and training officers. (This story would not change much if the problem were to avoid global warming or navigating the world out of an economic recession.) To analyze the differential impact of alternative decisions on potential combat outcomes, they decide to build the ultimate combat simulation. The wisest experts in military science are drawn together to define the model. A crack team of programmers is assembled to implement it on the most advanced computers, using state-of-the-art software tools. All relevant data bases are made available for the effort. In all cases, the best regarded modeling techniques are used, and military experts are consulted to ensure the realism of each submodel. For the ultimate model to be valid for a wide range of contingencies, all phenomena that might potentially be influential on battle outcomes are included, and all details that might prove pivotal are represented. This results in a very detailed model (because of a nail, the shoe was lost, because of the shoe the horse, etc.). The model is seen as realistic because it includes many factors, and, because it represents warfare at a very high level of resolution, lots of hard engineering data can be used, enhancing the model's credibility.

The resulting computer program is, of course, quite large, involving several million lines of code. And, because new studies often suggest needed modifications, the length continues to grow even after the model becomes operational. For the model to execute within a reasonable time it must run on the newest, massively parallel supercomputers. Even on these machines computational complexity limits the number of cases that can be run.

An additional constraint on the number of cases possible for any given analysis is the time required to set up the hundreds of thousands of inputs that specify the initial situation and model boundary conditions (the scenario). The outputs of the model are so voluminous that figuring out what happened is not trivial either. These problems are met by yet more computer programs that automate the process of setting up initial conditions and summarizing the outputs, effectively requiring the use of models to make sense of the ultimate model. The computer software requires highly trained operators, and they are unfortunately in short supply. It is still difficult to understand why certain simulated results occur, but warfare is a complex business, and no one really expected simulation to completely eliminate that complexity. In spite of these problems, the state-of-the-art graphics make for great demonstrations and study outputs are compelling to their sponsors. All the software managers and military action officers get promoted.

After some time, however, the ultimate model begins to develop enemies. Outputs often tend to show little impact for some types of forces or weapon systems, even though proponents may consider them crucial. Upon examination, certain aspects of model output can be demonstrated to be unrealistic. This is ascribed to details that failed to make it into the model. Modifications to the model are made, but this fails to satisfy those who are unhappy with the model's outcomes. Some assumptions about the nature of warfare that were made in designing the model are contested by detractors. Validation exercises designed to adjudicate these issues become politicized.

Although there are far too many inputs to do a thorough sensitivity analysis, occasionally someone discovers a case where a minor change creates a big swing in outcome. This is pointed out to be not unrealistic, but it is distressing to those who are rationalizing multibillion dollar expenditures by comparing simulations runs.

In response to all these problems the model is frequently revised. Unfortunately, its size makes revision nontrivial, and making sure revisions have not created obscure bugs is very time consuming (unfortunately, some of the original programmers have moved on to new jobs).

Eventually the entire enterprise collapses under its own weight, and use of the ultimate model is abandoned. Work immediately begins on its successor.

This story is admittedly dramatized and oriented to a worst case. Nonetheless, there exist failed simulation modeling efforts whose stories are quite similar, and most large simulations for policy analysis have encountered at least some of the problems mentioned here. Analysts that are wiser than the protagonists of this story certainly exist. However, the effort to build the ultimate combat simulation model is completely consistent with much existing conventional wisdom, evidenced both by the way we build models and the way we talk about them.

Computer modeling efforts in policy areas are prone to a variety of problems:

- Computer models often tend to be large and to continue growing throughout their history.
- It is often difficult to verify that a computer program correctly implements the advertised conceptual model. The sheer size and complexity of the models often make it essentially impossible to guarantee that they have been completely debugged and to ascertain whether there are conceptual errors in the model the program implements. These problems apply equally to the computer program and to its outputs, and affect all phases of a model's history—its construction, use, maintenance, and modification for new purposes.
- Because of this opacity, experts not associated with a model must rely on a priesthood of model cognoscenti. One must trust the priesthood to have done a good job and to be portraying the details of the model's internal structure correctly.
- Consequently, computer models are seldom subjected to peer review. Studies based upon large models are never truly replicated by second groups, as such replication would require the reimplementation of the identical conceptual model by other programmers.
- It is extremely difficult to adequately determine how sensitive model outputs are to uncertainties in the inputs.
- There is a corresponding tendency to underestimate (or ignore) the uncertainties (or inaccuracies) of inputs to models. This includes both explicit inputs and assumptions made in the process of building the model.
- There is a strong tendency to model in detail phenomena for which good models can be constructed, and to ignore phenomena that are difficult to model, producing a systematic bias in the results.
- These technical difficulties can interact with psychological or bureaucratic tendencies to produce a host of problems, including using models to rationalize institutional prejudices, poor models driving out careful thinking, and tending to emphasize the aspects of a problem that can best be simulated. The result can often be that models provide an illusion of analytic certainty for problems that are not well understood, or in the worst cases provide scientific costume for points of view that are self-serving.

While these problems have varying technical attributes, and particular problems may be managed through technological improvements, these myriad components are but symptoms of a single fundamental problem that lies in our assumptions about what models are and how they are used.

Why Models Go Wrong

While there is a general consensus in many policy areas that computer modeling is a difficult and troubled business, there is significant contention regarding the reasons for this difficulty. At one extreme, computer technologists will point to inadequacies of existing software tools, and suggest that more advanced technologies such as object-oriented simulation languages or expert systems could provide a fix. At the other extreme, critics of the use of models point to the difficulty of validating models in the social sciences and the propensity for models to obfuscate as much as they illumine. Examples of models being used to serve the interests of political and bureaucratic forces suggest that the problem lies in the lack of objectivity and scientific rigor on the part of modelers.

Yet another version of model skepticism holds that only small models are virtuous. All these points of view have their merit, but fail to capture the crux of the problem.

Technological fixes alone will not solve these problems. Computer science has provided numerous tools to facilitate the construction, maintenance, modification, and verification of large programs. These and future innovations may improve our ability to effectively produce computer programs implementing models, but they can only ameliorate, not solve, the problems enumerated above. For example, while a given model may be more tersely and understandably expressed in an appropriate language, there is a size threshold beyond which a program becomes difficult to understand regardless of the language used. Consequently, high-level languages and other techniques will not solve the general problem unless the reasons the models often become so large are addressed. Furthermore, no technological fix will eliminate the problems the developers of the ultimate combat model had with validation and sensitivity analyses.

Conversely, opposition to using computer models avoids the problems of model misuse by depriving analysts of a potentially very powerful tool. Similarly, there are problems whose inherent complexity cannot be wished away, and the dictum that only small or simple problems, or those admitting simple conceptual models, can be addressed by computer modeling means that computers may not be used to study just the phenomena for which their potential utility seems highest. This is counterintuitive to say the least.

By far the most telling evidence against all these explanations is the striking counterexample that there are very large models implemented with relatively primitive tools (e.g., FORTRAN) that have been enormously successful. Examples include the wide variety of mechanical and electrical engineering models that are used for computer-aided design systems. Such models consolidate existing theory and data into a package that can be experimentally validated to predict the behavior of interest in the systems they model. Once validated they can be used as surrogates for the actual system, in that the consequences of a proposed action can be evaluated by simply trying the action in the model.

Modeling efforts that become troubled typically involve models that cannot be validated experimentally. Validation may not be possible because the necessary experiments cannot be carried out, historical data are inadequate, theory is insufficiently mature to suggest models capable of making predictions, because cases of interest require initial conditions or boundary conditions that can only be guessed at, or because nonlinearities in the model cause even modest uncertainties in the inputs to produce substantial uncertainties in the results. Succinctly put, predictions of outcomes is only possible when we possess knowledge sufficient to make such a prediction. Various sorts of missing knowledge or inherent uncertainty can make the strategy of building a model that consolidates all we know and then using it as a surrogate for the target system unproductive. In particular, it is not possible to stage World War III several times to resolve questions of interest to combat modelers, nor to guess what the initiating circumstance might be.

Any model of the real world will certainly be only approximately correct, and no model can predict the behavior of a system to arbitrary precision. It is not some threshold limit on the accuracy, certainty, or precision in predicted outcomes that makes the critical difference between the applications that can be consolidatively modeled and those that cannot. Instead, the intended use for the model implies constraints on the accuracy required for it to be used as a surrogate for the actual system. Where the predictions of a best-estimate model can answer the question of interest, a consolidative approach to model development and use is preferred. Where the question to be addressed cannot be answered reliably by such a best-estimate model, the consolidative strategy for model development and use cannot be employed validly. The critical question that must be addressed is not simply the relative validity of models, but rather the appropriate strategy for using a model given its limitations.

Trying to Predict the Unpredictable

Consolidative modeling is the name I use for what many would consider the normal use of models. When adequate knowledge about both system characteristics and initial conditions exists, a model embodying this knowledge can predict system behavior reliably enough to be used for reasoning about the likely consequences of contemplated decisions. Such a computer model consolidates a large amount of information into a particularly useful form. A successful consolidative model can be used as a surrogate for the system itself, and once the model exists and is validated, much of the information used in its construction may be dispensed with.

Analytic strategies based on consolidative modeling typically have a two-phase structure: phase one constructs and validates the model, and phase two utilizes it by running particular cases. Such a model is a very powerful artifact, and much of the enthusiasm for modeling projects is based on the hope that a (predictive) consolidative model can be achieved.

Consolidative modeling is the design of models driven by what is known. Many of the great moments in the history of science, Newton's laws of motion, Maxwell's theory of electromagnetism, or Darwin's theory of natural selection, to name a few, are examples of the consolidation of previously determined facts into a single unifying model. Consequently, much of our scientific cultural heritage presupposes a consolidative research strategy. This paradigm for modeling a target system is such a deep part of our culture that many may regard it as obvious.

Its usefulness, however, depends on means being available to validate the resulting model. For predictive purposes, validation normally requires confirmation by experiment. The variety of means short of experimental validation that may be employed to improve model quality, including experimental validation of submodels where possible, determining model parameters from validated sources, ensuring the plausibility of outcomes (so-called face validity), or adding more realistic details to the model, cannot solve the fundamental problem facing the designers of the ultimate combat model. Where there are uncertainties that make strong validation impossible, the consolidative modeling paradigm will not lead to reliable results.

The fundamental error committed by the designers of the ultimate combat simulation was the use of the consolidative approach to modeling where it was inappropriate. For the questions they wished to answer, no best-estimate model could predict outcomes reliably enough to be used as a surrogate for the real world. Once they committed themselves to an untenable research strategy, success was impossible no matter how well the inappropriate model was crafted.

It is instructive to consider how the myriad problems that afflicted our fictional designers were generated by such a fundamental mistake. A central problem was the all-too-prevalent tendency to emphasize building a model over carefully thinking through an analysis. Emphasis on building a model makes sense if the model produced accurate predictions. Once such a tool was in hand, a wide variety of uses could then be entertained. However, in a context where reliable prediction is not possible, a model is of little use without an analytic framework that makes its outputs relevant. In such a context, a methodology that first develops a general purpose model and then considers possible applications runs a deep risk of expending large amounts of resources on a model that cannot be employed validly for any analysis.

Similar considerations pertain to difficulties with verification and validation. When models can be used for prediction, even a single experiment in which the model predicts system behavior successfully can provide a great deal of validation and verification. The issues of verification and validation become much more vexing when model outputs cannot be checked by experiment, and when the model is not expected to predict the details of any actual event.

Many of the problems encountered with the ultimate model were a consequence of its large size. The tendency for models to grow very large is often a symptom of the misapplication of consolidative modeling methodology. When a model can be used to predict, it can be validated through its predictions. Among competing predictive models, the most preferred is the simplest that makes sufficiently accurate predictions. However, when models cannot be validated in such a direct fashion, the quality of a model must be assessed by other means.

Often, judgments of model quality are based upon the degree of completeness (the inclusion of all factors and phenomena that might influence outcomes for at least some cases). In contrast with models of predictable target systems that are simplifications of reality, models of unpredictable systems often are attempts to copy the full complexity of the target system. The designers of such models have fallen prey to false reductionism: The belief that the more details a model contains, the more accurate it will be. This reductionism is false in that no amount of detail can provide validation, only the illusion of realism. By designing a model in the absence of an analytic strategy, the designers left themselves open to an unending process of adding ever more detail. Entertaining the myth that they were building a predictive model but without the backstop of a thorough validation, the fictional designers pursued various social definitions of model quality. The best military experts, and the best programmers could, perhaps, produce a best-estimate model, but the uncertainties of warfare made it certain that it would nonetheless be wrong. Furthermore, as model specifications and evaluation necessarily rested on subjective judgments, the eventual politicization of what was portrayed as a scientific endeavor was inevitable.

The designers of the ultimate combat model were, to some extent, victims of advancing computer technology. In earlier eras, limitations in machine speed made heroic modeling efforts impossible. As computers have become faster, our cultural biases toward consolidative modeling can now lead the unwary to pursue ever greater detail, producing models of unbounded complexity. Advanced computer graphics have further motivated the use of very detailed models. This detail can be useful for applications where consolidative strategies can succeed, or where realism is an end in itself, i.e., for entertainment or training applications. For analysis, however, often the need is to discover useful simplifying assumptions. Consolidative strategies resulting in highly detailed models often have not served this end.

To summarize, consolidative models make great screwdrivers, but they pound nails rather poorly. For problems involving practical barriers to experimental validation, significant uncertainties, or strong nonlinearities, a different approach is needed.

EXPLORATORY MODELING

If we do not wish to repeat the mistakes of the ultimate combat model, and do not wish to abandon computer modeling as a tool for understanding complex and uncertain problems, what are we to do? To answer this question, let us pursue our gedanken history a few more steps. Imagine the fictional designers of the ultimate combat simulation, called to testify before a congressional committee investigating the project.

Congressman: *And so, after spending zillions of taxpayer dollars, you produced a system that could not predict the details of even one battle.*

Designer: *That's true, but it was never intended to. If you had asked me before the project began, I would have told you that we would be simulating the general character of battle, not the individual details.*

Congressman: *If it doesn't get the answer right, what good is it? How can you justify the construction of this white elephant?*

Designer: *Models such as this one are useful, not because they predict the details of battle, but because building and using them improves our insight.*

When challenged, defenders of models that cannot predict system behavior will invariably fall back on an appeal to the model's utility in improving insight. This is, I believe, the correct answer. However, if we are to improve modeling for policy analysis, we must understand how it is that insight can be produced by such means.

How can we answer the skepticism of the model critics, like the fictional congressman above? If uncertainties are large, then with probability approaching unity, any model we construct will be incorrect in at least some details. How can valid conclusions be arrived at using "wrong" models?

When a model is not a veridical surrogate for the target system, the meaning of its outputs must be provided by a larger context. Typically, this context must be an analytic strategy that justifies its use. A few examples of possible strategies should suffice to make the skeptic's question seem less paradoxical.

One simple example is the use of models as existence proofs. Demonstrating a plausible model that has unexpected properties can usefully reorient an analysis, even when the model is unlikely to be correct in detail. This is because the analysis must confront the range of possible behaviors of the system, given what is known. The unexpected result expands this range. It creates new knowledge, whether or not it is true in the sense that it copies the actual system behavior.

A related use is hypothesis generation. A model can be helpful in suggesting an explanation for a puzzling fact, even if it is eventually proven wrong. Where no explanation previously exists, a model that suggests a plausible explanation can guide the search for other examples, or new data, or provide a basis for decision making superior to guessing if it represents all that is known.

A more complex example of this type is provided by analysis in situations where risk aversion is prudent. Here, an exploration that develops an assortment of plausible worst cases can be very useful for designing hedging strategies. This is true even if the models are not validated and their sensitivities are unknown. Indeed, validation and sensitivity analysis are concepts that are relevant primarily in the context of consolidative modeling. They are nonsequiturs in the context of exploratory modeling where issues of quality for exploratory modeling must be centered on ensuring the validity of the analytic strategy. Model-specific quality control issues are limited to verification that the model is plausible and that the software implements the model intended.

Other examples of exploratory research strategies include the search for special cases where small investments could (plausibly) produce large dividends, or extremal cases where the uncertainties are all one-sided, and *a fortiori* arguments can be used. In all these cases, individual model runs are not being treated as providing predictions or explicit answers to policy makers' questions. Instead, new information is being generated that can be helpful in making an informed policy decision. This new information was implicit in the prior knowledge that defined what was plausible. The role of modeling was to transform this implicit information into a more useful form.

Thus, the skeptic's question is actually a version of the more general one: "Of what use is partial information?" For some questions, partial information may be useless, but for many problems partial information can provide partial answers. For most policy problems, some decision must be made (at least the decision to do nothing), regardless of the level of uncertainty. Policy analysis requires understanding the implications of what is known, which for systems with insignificant nonlinearities may not be obvious. When dealing with complex systems, both what is known and what is uncertain may be best represented by computer models. Thus, computers can have a role in revealing the implications of what is known or believed and the possible consequences of what is unknown or uncertain.

Even quite simple models are capable of exhibiting behaviors that surprise their creators. Consequently, building and exercising models have the potential of revealing unanticipated implications of our knowledge and assumptions. Even when a model is not validated, it can serve as an inference engine, showing us where innocuous-appearing assumptions lead to predicted behaviors different from initial expectations. By throwing light on treacherous assumptions or revealing unrealized implications of existing information, computer modeling can perform an important service.

When used for exploratory modeling, the computer functions as a prosthesis for the intellect, supporting the discovery of implications of a priori knowledge, novel explanations of known facts, or unrealized properties of conjectures. This is a very different use for models than that of consolidating knowledge into a single package that can then be used as a surrogate for the target system. This difference also has significant implications for the methodology and the technology of model building and use.

Uncertainty and Exploratory Modeling

Uncertainty, or lack of knowledge, implies that there are many models that might plausibly represent the system of interest. Conversely, what is known constrains the set of models that can be considered plausible. To better understand the interaction between what is known and what is not, it is useful to consider the set or ensemble of all models consistent with what is known. This ensemble may often be of infinite size, and thus can never be constructed explicitly. Nonetheless, when thinking about research strategy, the concept can be helpful. And, within a fixed modeling framework, for particular purposes, the ensemble of plausible models may be represented and manipulated usefully.

When uncertainties are small, the entire ensemble may be adequately represented and reasoned about by means of a single example, as the differences among the properties of the plausible models will be similarly small. (The use of a single model to represent all plausible models is the consolidative modeling paradigm.) For other systems and purposes, however, the properties of members of the ensemble of plausible models may be sufficiently diverse that a large sample from the ensemble must be examined. It is the process of examining this larger sample that I call exploratory modeling. An exploratory analysis can involve a search for key examples, or may infer general properties of the entire ensemble of plausible models from the sample examined.

A concrete example may help to clarify this concept. The ensemble of plausible models is particularly easy to understand for the case of parameterized models. If a computer model contains N real-valued parameters, each with a plausible range, there is an implied virtual ensemble of models, one for each combination of parameter choices. This ensemble corresponds to a rectangular solid in an N-dimensional parameter space. (Constraints on the combinations of parameter values can produce more complex shapes.) Any point in this N-dimensional space corresponds to a particular model. Those within the boundary or envelope of plausible ranges are members of the ensemble of plausible models.

An exploratory analysis involves running models drawn from this ensemble. The process of selecting which model, that is, which combination of parameters, to run depends on the question being asked. For example, to determine the statistics of a model output across the ensemble, distributions for each parameter, such as uniform or Gaussian, can be asserted and samples from the ensemble can be chosen randomly to collect statistics. If, on the other hand, we wish to find the model that maximizes an output measure, a search strategy is required. Policy questions often motivate an interest in determining the regions of parameter space where certain properties are true. Thus, we might wish to discover the regions of the space for which an output measure is positive, or where the model trajectory is chaotic. This requires a different sort of adaptive sampling, one designed to determine the location of the boundary most efficiently.

It is useful to contrast exploratory modeling to the concept of sensitivity analysis. For any numerical computer program, sensitivity analysis is the process by which uncertainty in inputs is related to uncertainty in outputs (Miser and Quade 1985, Ronen 1988, Suri 1987, 1989). For special cases, model sensitivities can be determined analytically. More generally, sensitivities are estimated by running excursions. Typically this is done by varying parameters one at a time. For the N-parameter model discussed above, this corresponds to a run for each face of the rectangular solid, requiring $2N$ runs. These runs will be sufficient to estimate the model sensitivities only if the effects of parameters on the model are independent and monotonic. (The application of this approach to cases where these conditions are violated can be rationalized as an approximation if model variability is small over the range of parameter uncertainty.)

In the general case, the maximum divergence from the best-estimate case can occur in the interior of the rectangular solid, and not on a face at all. In such a situation, determining the plausible range of outputs requires sampling throughout the volume of the plausible range of inputs. Such an approach is best described as exploratory analysis, as the sampling strategy should be determined by the needs of the analysis and the available computational resources. Whatever it is called, a fine-grained combinatorial sampling of the volume of the plausible envelope requires many more than $2N$ cases. Simply extending sensitivity analysis to test at the vertices of the solid requires 2^N runs, which already is much greater than $2N$ for large N.

Large models like the ultimate combat model can have thousands of parameters and long run times, making even $2N$ excursions impractical. Consequently, for them even the simple version of sensitivity analysis is frequently not done. Such models are not designed with sensitivity analysis in mind; rather, they are designed as though the sensitivities were known a priori to be strongly bounded.

Here we see the failure of the ultimate combat model from a new perspective. In the inability to perform sensitivity analysis adequately, the inappropriateness of the consolidative paradigm is clearly revealed. If the uncertainties are small, and model behavior is well characterized, sensitivity analysis can be performed. However, when uncertainties are significant, running a best-estimate case and then doing an inadequate sensitivity analysis is a recipe for self-deception. Instead, a different strategy for sampling the ensemble of plausible models is required.

Representing the Ensemble of Plausible Models

Exploration need not be restricted to the values of numeric parameters, but can also be conducted across different sorts of nonparametric uncertainty. For example, members of the ensemble of plausible models might have differing numbers of variables, differences in data flow graphs, or computational algorithms. The modeler may know that X is determined from Y and Z, but be uncertain of the exact functional relationship. The modeler might know, for example, only that a function is continuously differentiable, bounded, and monotonic. The ensemble of such functions is not only infinite, it is of infinite dimension. Nonetheless, by representing such a function in terms of some basis, e.g., polynomials of arbitrary order, or Fourier series, the ensemble of possible functions can readily be searched across, or sampled from.

While the general case of representing and searching across nonparametric uncertainty presents fundamental research challenges, various special cases can be tractable. Often, plausible options can be enumerated, i.e., alternative submodels, equations, or rules. The ensemble to be searched can then be represented as the Cartesian product of the various enumerated ranges. Nonparametric uncertainties can be structured as graphs, trees, or lattices, for example, as sequences of possible decisions in a game-structured scenario.

In general, nonparametric uncertainty can be represented if a basis exists that is adequate to express the full range of possibilities, and if available knowledge can be expressed as constraints among admissible combinations of the basis elements. Where alternative bases exist, choosing among them can have significant consequences for the resulting exploratory analysis. For example, the choice between expressing an unknown function as an arbitrary polynomial or a Fourier series can have real consequences for both the ease with which constraints on that function can be expressed, and the properties of a search through function space structured by that basis. The choice of basis imposes a topology on the ensemble of models, affecting which models are viewed as similar to one another. This topology provides the foundation for any search or sampling strategy beyond random selection.

Whether the ensemble of plausible models can be represented explicitly or not, the problem of analysis using exploratory modeling can be conceptualized as the problem of how to select the limited number of experiments that can be run practically to best inform the question of interest. This selection process can be thought of as sampling if the goal is to infer properties of the ensemble, and hence, of the actual system, or as search if the goal is to find models with special properties. Typically, the distinction between search and sampling will quickly become blurred as insights gained in the early stages of analysis affect the selection of later experiments.

In general, a mathematically rigorous strategy for sampling will not be available. Instead, a sampling strategy may involve using human judgment to prioritize the investigation of the uncertainties involved. Consequently, the result of an exploratory analysis will typically not be a mathematically rigorous answer, but rather an imperfect image of the complete ensemble that improves gradually as more cases are run. Given a fixed analytic budget (in dollars, people, or time), the analysis must provide the most useful results possible based on what is known about the problem at-hand.

Types of Exploratory Modeling

There are three general types of applications where exploratory modeling can be used; they can be labeled data-driven, question-driven, and model-driven. Data-driven exploration starts with a data set, and attempts to derive insight from it by searching over an ensemble of models to find those that are consistent with the available data. Question-driven exploration searches over an ensemble of models believed to be plausible to answer a question of interest. Model-driven exploration involves neither a fixed data set nor a particular question or policy choice, but rather is a theoretical investigation into the properties of a class of models. For each of these types, it is possible to describe the exploratory modeling process in greater detail.

Data-Driven Exploratory Modeling. Data-driven exploration is used to reveal implications of a data set by searching through an ensemble of models for instances that are consistent with the data. While this process may not produce a "correct" model, structure in the data can be discovered by noting regularities in the modeling results.

For example, data modeling for econometrics is typically done by guessing a regression equation based on a priori theoretical considerations, fitting it to the data, and, if satisfactory results are not obtained, modifying the equation iteratively until an equation that explains a significant portion of observed variability is discovered. This process, sometimes referred to as specification search (Leamer 1978), has produced models of value for both forecasting and explanation, but also has pitfalls for the unwary. Standard statistical tests for goodness-of-fit do not take the (human-implemented) search process into account. Consequently, naive application of these tests may not protect against overfitting and other invalid results. (This danger is especially pronounced for computer-assisted and computationally intensive searches.) Consequently, conventional wisdom in econometrics and statistical modeling approves of data mining only when the ensemble of models is strongly constrained by a priori knowledge.

Examples of incautious exploratory data modeling have caused the general notion of search across models and some specific algorithms, e.g., stepwise regression, to fall into disfavor with much of the statistical community. However, the means exist to avoid these problems. For example, tests of statistical significance can be recalibrated to compensate for the search across the ensemble of models by means of the bootstrap or permutation methods (Efron 1982, Efron and Gong 1983).

The concept of search over an ensemble of models looking for good fits to the data applies equally well to other model formalisms. For example, there has been recent enthusiasm for modeling data by so-called "neural networks" or "connectionist" models. This approach is mathematically analogous to regression, with the network architectures defining nonlinear regression models (Geman, Bienenstock, and Doursat 1992, MacKay 1992). Data modeling with neural networks typically requires experimenting with a wide variety of network architectures and data representations. This human-mediated search process is seldom documented; rather, the final successful model is presented without the context of less successful variants. (Unfortunately, the use of statistically rigorous measures for goodness-of-fit are the exception in the literature of this field.)

Quite often, causal simulation models are calibrated by adjusting parameters so that model performance matches the available data. While this is also usually human mediated, and uses less than formal measures-of-fit, this process also can be best understood as a search through an ensemble of similar models for one with desirable properties.

In all these examples, search is most commonly done via human-implemented iteration that concludes with a single successful model. Increasing availability of computer power suggests opportunities for both machine assistance in the search process and producing more than one model as an output. When the data being modeled have associated uncertainties, a single model of the data can be misleading. For nonlinear models the goodness-of-fit may not be unimodal across the ensemble of models, making multiple outcomes of specification search useful. For at least some purposes, it may be more enlightening to specify an acceptable threshold for a figure of merit and produce a sampling of models that fit the data to within that threshold. Such an approach can provide the advantages of conventional data modeling without risking misplaced overconfidence in the results.

Question-Driven Exploratory Modeling. This type of analysis searches among an ensemble of plausible models to answer a question of interest or illuminate policy choices. Where uncertainties are significant, an exhaustive search across all plausible models typically will not be possible. The sampling strategy must therefore be designed to produce the maximum help in answering the question of interest from a limited number of computational experiments. Examples of possible strategies include: sampling randomly or uniformly across the ensemble (including nonparametric sources of uncertainty) to determine the range of plausible outcomes, searching for the worst cases to support the construction of risk-averse hedging strategies, listing plausible scenarios for which existing policies fail cataclysmically to generate needed contingencies and triggers, discovering bounding cases to support *a fortiori* arguments, and discovering boundary cases that reveal the conditions that would favor alternative options.

Consider a simplistic example in which we wish to answer the question: If we do X, do we win or lose? While the models involved may produce complex and voluminous outputs, for the purposes of this question, they may be boiled down to a single bit. To know the details of a particular scenario to many decimal places is not particularly helpful in answering the question. Instead, we need to know how the ensemble of plausible models divides up into those where we win (color them white) and those where we lose (colored black). Of course, if the ensemble is all one color (white or black), then the analysis (and the answer to the question) is particularly easy. Even when both outcomes are plausible, an analysis of the assumptions under which each may occur may be useful for making the actual decision to implement option X or not. Patterns across the ensemble can reveal unrecognized connections between the implications of different kinds of uncertainty that can suggest fruitful alternative strategies, prove helpful in prioritizing research to further constrain the range of plausibility, or support a more educated hunch on the part of the decision maker who must make this decision.

Our ability to discern patterns in outcome across the ensemble of plausible models depends on being able to define a topology on the ensemble such that similar models have similar outcomes. (The reader may find it helpful here to think again of the simple case of a parameterized model.) If for a given topology the white and black regions are thoroughly intermixed, e.g., form a fractal Cantor dust, (see Schroeder 1991), then little of use can be concluded from an exploratory analysis beyond estimating the relative probabilities of outcome given probability distributions for the uncertainties. At the other extreme, the boundary between the white and black regions could be very simple; for example, the boundary could be a hyperplane in the space of models. In such a case, the shape and position of the boundary could be estimated from a relatively small number of samples, as the size of the sample would scale with the area of the boundary rather than with the volume of the space. As the boundary between the regions becomes more complex, the process required to discover the boundary becomes similarly complicated. Similarly, more complex questions, such as determining the rank ordering among a list of alternative proposed actions, involve discerning multiple regions in model space.

Model-Driven Exploratory Modeling. This type of exploration investigates the properties of an ensemble of models without reference to a data set or policy question. Model-driven exploration can thus be viewed as an example of experimental mathematics. It will be useful for policy analysis whenever a new class of models is proposed to represent a system of interest. Properties of this class must be determined to assess how or whether such models might be useful.

An example of a model-driven problem that frequently occurs involves the relationship between different models that putatively represent the same phenomenon, in particular, two models at different resolutions. In the general nonlinear case, two such models will be effectively equivalent only for a limited range of state, only approximately, and only for a limited time. Determining the conditions for an aggregate model to be effectively equivalent to a higher-resolution model in general requires computational experiments. Another prominent instance of model-driven exploration occurred in recent developments in nonlinear dynamics, where chaotic trajectories were observed computationally before they were either explained analytically or demonstrated in actual systems. Determining whether a class of models is capable of exhibiting chaotic behavior will, in general, require exploratory analysis through computational experiments.

Question-Driven Exploration and the Size of Simulation Models. We have seen how the inappropriate use of consolidative modeling methodology can result in very large simulation models, leading to a host of problems. The growth in model size is driven both by the desire to build multipurpose models and by the substitution of high-resolution realism for experimental validation. Question-driven exploratory modeling avoids both of these routes to oversize models and thus can provide a route to solving many of the problems policy modeling has been prone to. This is because exploratory modeling allows for much greater flexibility in choosing appropriate levels of resolution.

The model that is built to answer a particular question should generally be the smallest (lowest resolution) that satisfies that purpose. Keeping the model as limited as possible minimizes problems with understandability and sensitivity analysis. As different questions are asked during the course of an analysis, models of different resolutions may be required. Addressing broad tradeoffs may require aggregated models of wide scope, whereas models for specific questions may require more focus and detail.

It must be recognized, however, that model resolution is a modeling artifact against which study conclusions are usually desired to be invariant. Model resolution is thus often a form of nonparametric uncertainty over which exploration may be desirable. When resolution can be varied simply by changing the mesh size of an array representation, such a requirement may not be particularly arduous. When increasing resolution requires breaking concepts down into subconcepts, however, labor-intensive modeling and programming may be required. Furthermore, higher-resolution models typically will have a greater number of uncertain parameters, increasing the dimensionality of the space to be explored. For both of these reasons, exploratory research strategies may need to confront issues regarding the scale of models and the range of exploration across resolutions.

Search strategies generally cannot visit every possible alternative, and must instead use heuristics to guide the search. Exploratory-modeling strategies likewise cannot guarantee absolute answers, but must endeavor to provide the most information for the resources available. Often human judgment must be used to focus attention on the aspects of modeling that appear most critical for the question at-hand. One possible approach to doing this is selective resolution where initial modeling is done with relatively aggregate models, and the results of this preliminary analysis are used to guide the selective use of higher-resolution models, with detail added only for the attributes that appear to have a large impact on the question of interest. In this way, the results of preliminary analysis with aggregated models can guide the allocation of resources in more detailed modeling. At the same time, by adding resolution only where necessary in the context of a specific question, the use of monolithic high-resolution models, with all their attendant difficulties, may be avoided.

TECHNICAL OPTIONS FOR SUPPORTING EXPLORATORY MODELING

We have seen how the decision to build the ultimate combat model led to many problems and produced little real contribution to policy analysis. What should its designers have done instead? It is apparent that the variety of needs posed by wide-ranging studies of systems with significant uncertainties cannot be supported by a single high-resolution model. Models built prior to identifying the questions of interest will seldom be ideal (or even adequate) to address them. Thus, the use of a large multipurpose model carries with it the enormous risk that the model will constrain the set of questions that are asked, creating a symmetric bias to the analysis. This is akin to looking for a lost quarter only where the light is good.

In the context of exploratory modeling, the models for any given study typically will be crafted for that study. Consequently, building a fixed model or models is not the way to facilitate a wide range of policy studies. For example, the desired level(s) of resolution in a model should be determined by the research strategy for dealing with uncertainty, which will vary between studies. And multiple alternative models or model variants may need to be examined to avoid being confused by artifacts of particular modeling choices.

In contrast to consolidative modeling, where a single model can be useful, exploratory modeling can only produce useful results through a constellation of alternative model outcomes. The product of consolidative modeling is an artifact, a single model presumed "correct" for the purposes of its use. Exploratory modeling is not a product but rather a process. How can this process be better supported? The goal of the ultimate combat model project was to build software prior to beginning analytic studies that would make those studies more effective. This goal can be achieved, but the software cannot be some putative ultimate model. Instead, support can best be provided through tools and broadly useful model pieces or constructs that can facilitate the exploratory modeling process. As existing technology to support modeling has evolved from the consolidative modeling tradition, it is designed to support the construction and use of single models. Consequently, the different requirements of exploratory modeling present opportunities for different tools to support the construction and use of multiple models.

The purpose of an exploratory modeling environment would be to allow users to navigate efficiently through the space of plausible models and model outcomes to construct lines of reasoning and to learn about the implications of both knowledge and hypothesis. There are two general needs that such an environment might address: assistance in managing the complexity of the exploratory modeling process, and support for more powerful and facile generation of new models. These two general categories can be subdivided further to produce the following list of needs:

- means for the analyst to understand and manage the complexity of the numerous models, cases, and relationships among them;
- means for visualizing the results of exploratory modeling;
- support for iterative and adaptive modeling;
- representation of the ensemble of models;
- automated search and sampling of ensembles of models.

Support for Managing the Complexity of an Evolving Analysis

An exploratory analysis can involve large numbers of computational experiments. The cognitive complexity of managing such an analysis can be a significant burden for the researcher. The ability of the analyst to keep track of the myriad details of model characteristics, interrelationships, cases, histories, implications, status, and outcomes could be enhanced greatly by an appropriate software environment.

To support an analytic process based on performing computational experiments, an exploratory modeling environment could provide an automated laboratory notebook with a page for each computational experiment. In analogy to a physical laboratory notebook used for physical experiments, each page would record all the information needed to reproduce the experiment, records of the results of the experiment, and annotations explaining why the experiment was performed, what was learned from it, and how these results relate to the global research strategy. These pages would form the conceptual center for a data facility that would serve as an electronic record of the evolving chain of reasoning that constitutes the analysis. Such a facility would be more than a data base, as it would contain not only data but also computer models, model runs, model outputs, human notations, and all needed relationships among these entities. Such a facility would in fact be a hypermedium (Barrett 1988, Wurman 1989) for modeling.

Such a history would be a record of all modeling experiments, including the model variants, and data going into any modeling experiment as well as its outcome. The ultimate environment would have attributes of brainstorming and outlining tools, data base facilities, version control systems, and general purpose modeling environments. Such an environment would assist the user in keeping track of an evolving analysis involving the construction of multiple models and model variants, case runs, changing assumptions used, and tentative conclusions drawn.

In addition to documentation of individual experiments, such a notebook should provide the means of documenting the research strategy that dictates the selection of the experiments, and the relationships among the results that support study conclusions. Because of these relationships, individual modeling experiments can gain contextual significance or meaning that they would not have in isolation. In this way, the use of multiple models in the analysis of a problem can provide a novel sort of software modularity compared with the use of single monolithic programs.

A monolithic model must represent all of the complexities of the system as procedural (algorithmic) computer programs. By using multiple models, some portion of this complexity can be exported outside the models, where it can be represented as declarative information inherent in the relationships among models, cases, and data bases. This is beneficial in that it may allow for smaller models that may be easier to understand and because declarative information implies opportunities to devise more powerful tools than are feasible to manipulate procedural code.

Visualizing the Results of Exploratory Modeling

Individual model runs can produce voluminous data; multiple runs can produce astronomical quantities. Developing intuition based on the results of modeling requires adequate means for viewing these results. The deluge of data that can be generated makes it impossible for users to quantitatively examine more than a fraction of it. If insight is to be generated from these outputs, means must be available to easily view the data for various purposes. Means for viewing the results of exploratory modeling would be useful both for presenting final results of the analysis to the consumer and for providing a powerful means for the analyst to improve his or her intuition.

With the advent of raster graphics, entire fields of variables can be converted to color images. Information conveyed in this way undergoes a qualitative change because it utilizes the tremendous pattern-recognition capabilities of the human eye-brain system. An environment for exploratory modeling should include capabilities for visualizing data harvested across multiple cases. Some of these graphical tools might resemble the displays available in statistical-data modeling packages. Thus, in the context of a question-driven exploratory analysis where models had been sampled across N dimensions of uncertainty, one might display a two-dimensional slice or projection of the N-dimensional point cloud of model outcomes. This view of the results could be manipulated by such devices as sliders or handles that would allow the user to explore the results graphically.

Support for Iterative Modeling

Exploratory modeling is often accomplished by human-mediated iterative testing and revision of a current model, resulting in an evolving genealogy of experiments. This process can be supported in a number of ways. One approach is based on the use of high-level languages that support WYSIWYG (What You See Is What You Get) modeling. The goal of WYSIWYG modeling is made possible by the flexibility of exploratory modeling to obey constraints on the size of models, the availability of high-level programming languages designed for understandability and modifiability (see, for example, Allen and Wilson 1988, Shapiro et al. 1985, 1988), and the use of interactive computer software environments, which allow easy inspection and manipulation of model source code, parameters, and outputs. These ends can also be sought through open model architectures that promote transparency and ease of revision, modeling tools and languages that support full object orientation, end-user readability of model code, highly interactive direct manipulation interfaces, strong configuration management, and rapid execution.

New models need not always be built from scratch. It may be possible to construct modeling environments for specific policy areas that incorporate baseline model components to allow model construction through combining submodels from libraries of model components, varying parameters, and model revision on the margin. Instead of building megamodels to support a variety of studies, modeling environments could be constructed that incorporate baseline models, and other tools to aid in model construction, so that the process of building numerous model variants is made tractable.

Constructing new models by combining model components can be facilitated through standards for model interfaces. Interface standards would also facilitate construction of hierarchic ensembles of models and standard tools for viewing their behavior. The definition of such interfaces is a challenging problem that can be eased somewhat by having model components interact through a common data facility with an associated standard data dictionary.

Representing the Ensemble of Plausible Models

While better support for the iterative modeling process would be useful, exploratory modeling will necessarily be constrained if programming is required for each model variant generated. To eliminate this constraint, tools are needed that will allow new models to be generated with less labor and greater transparency than modifying the program code directly. To construct such tools, it is necessary to represent explicitly the space of models from which variants are to be drawn.

Representing a space of computational experiments involves both defining a unique descriptor for each experiment, and providing a transformation from descriptors to model instances. For example, a list of parameter values together with a parameterized model defines a model instance. Relative to the space of modeling experiments generated by all possible parameter combinations, a particular list of parameters can be considered to be a descriptor for a corresponding experiment, and the parameterized model an implementation of the transformation from descriptors to executable experiments. The parameterized model is thus best thought of as a model schema in that it is associated with a large number of potential model instances.

More complex ensembles of models involving unbound functions, alternative submodels, and other sorts of nonparametric uncertainty, can similarly be represented by a combination of descriptors that define syntactically the assumptions that vary across the ensemble together with a model schema that encodes the knowledge that does not. In this way, it may be possible to program at the schema level with the program defining a space of models, and all particular models being generated from the schema as a result of specifying a particular model descriptor. With appropriate tools for the generation of model instances, this would both ease the process of model revision, and free modeling methodology from its present bias toward best-estimate models.

Current methodology is built upon a powerful metaphor, that a computer program is a model. Despite this metaphor's power, it can be misleading. The creative process in the minds of policy analysts typically results in conceptual models that leave many details unspecified. For example, the analyst might hypothesize that X depends upon Y and Z, without identifying the functional form of the dependency, let alone the exact values of all parameters. Such a conceptual model is best represented by a model schema that corresponds to a multitude of particular models, and not by any representative single model.

Current technology requires the modeler to focus on one particular model out of the range of possible instances in order to begin implementing his or her ideas. Typically, neither the program implementer, author of the conceptual model, nor the customer for the final policy recommendation will have any idea how much bias in the answer is introduced by this early design decision. Narrowing the focus to a single model should preferably be postponed until the latest possible time, ideally until the specification of a particular computational experiment.

Support for the Search Across Models

Once the ensemble of interesting computational experiments has been represented, a variety of tools for generating individual experiments become feasible. Iterative development could be supported by direct-manipulation user interfaces that provide the user with the ability to select particular experiments without manipulating source code directly. More speculatively, once the ensemble of models is represented in the computer, full automation of the generation of instances becomes an interesting option, as this would allow for computationally intensive research strategies. With such a mechanism, composite computational experiments become possible, where the analyst would specify an ensemble and a sampling strategy, generate thousands to billions of individual experiments, and then examine the results. Sampling strategies could include random Monte Carlo approaches, automatic sensitivity analysis (for a related idea, see Rothenberg, Shapiro and Hefley 1990), or adaptive search to identify special cases (i.e., boundaries, thresholds, or extremal results). Such composite experiments have effectively infinite inherent parallelism, large grain size, and small interprocess communication requirements, making them ideal for exploiting massively parallel supercomputers, or networks of workstations.

CONCLUDING REMARKS

The computer's capabilities for rapidly performing many more arithmetic or logical operations than the human mind gives it a prominent role in addressing problems of great complexity. We are still early in the process of understanding how best to design computer systems so that human capabilities are enhanced, not eclipsed, and the strengths of the computer utilized while its liabilities are minimized. The inappropriate use of computer models can clearly result in invalid conclusions. Questionable research strategies that involve the use of computer models for policy analysis are all too frequent, and more critical scrutiny of models needs to be encouraged. However, in this paper I have tried to demonstrate that in the context of a properly conceived research strategy, computational experiments can be a very useful tool.

In numerous scientific fields, research methodologies based on computational experiments have often been controversial when first introduced. In time, the increasing availability of computational power typically results in growing experimentation with exploratory modeling approaches and the credibility of this research has grown as increasing numbers of workers in these various fields become computer adept (Strauss 1974, Campbell et al. 1985, Rose and Dobson 1985, Anderson 1988, Lipman, Marr and Welsh 1989). We can anticipate that the use of exploratory modeling to break trail for more traditional science is likely to become increasingly important.

Due to the complexity of the systems of interest and the abundance of problems for which no model can be validated experimentally, no disciplines seem better disposed to benefit from the exploratory approach to computation than do operations research and the policy sciences. Increasing sophistication among policy researchers with this approach could not only contribute significantly to improved analysis of complex and uncertain problems, it could also provide a modicum of protection against being fooled by our own models.

ACKNOWLEDGMENT

This paper is an extensively modified version of a previous RAND note (Bankes 1992). That previous work was supported through the Arroyo Center, the U.S. Army's federally funded research and development center (FFRDC) for studies and analysis, operated by RAND under contract MDA903-91-C-0006.

REFERENCES

ALLEN, P., AND B. WILSON. 1988. Modeling Qualitative Issues in Military Simulations With the RAND-ABEL Language. In *Proceedings of the 1988 Winter Simulation Conference*, Association for Computing Machinery, San Diego, Calif.

ANDERSON, A. 1988. Learning from a Computer Cat. *Nature* **331**, 657–658.

BANKES, S. 1992. Exploratory Modeling and the Use of

Simulation for Policy Analysis, N-3093-A, RAND, Santa Monica, Calif.

BARRETT, E. 1988. *Text, Context, and Hypertext, Writing With and for Computers.* The MIT Press, Cambridge, Mass.

CAMPBELL, D., J. CRUTCHFIELD, D. FARMER AND E. JEN. 1985. Experimental Mathematics: The Role of Computation in Nonlinear Science. *Commun. ACM* **28,** 374–384.

EFRON, B. 1982. *The Jackknife, the Bootstrap and Other Resampling Plans.* SIAM, Philadelphia, Penn.

EFRON, B., AND G. GONG. 1983. A Leisurely Look at the Bootstrap, the Jackknife, and Cross-Validation. *Am. Statist.* **37,** 36–48.

FREEDMAN, D. A. 1981. Some Pitfalls in Large Econometric Models: A Case Study. *J. Bus.* **54,** 477–500.

GEMAN, S., E. BIENENSTOCK AND R. DOURSAT. 1992. Neural Networks and the Bias/Variance Dilemma. *Neural Computation* **4,** 1–58.

GOELLER, B. F. 1984. Guidelines for Constructing Policy Analysis Models. P-6975, The RAND Corporation, Santa Monica, Calif.

HODGES, J. 1991. Six (Or So) Things You Can Do With a Bad Model. *Opns. Res.* **39,** 355–365.

LEAMER, E. 1978. *Specification Searches: Ad Hoc Inference With Nonexperimental Data.* John Wiley, New York.

LIPTON, R., T. MARR AND J. D. WELSH. 1989. Computation Approaches to Discovering Semantics in Molecular Biology. *Proc. IEEE* **77,** 1056–1060.

MACKAY, D. J. C. 1992. A Practical Bayesian Framework for Backpropagation Networks. *Neural Computation* **4,** 448–472.

MEADOWS, D. H., AND J. M. ROBINSON. 1985. *The Electronic Oracle: Computer Models and Social Decisions.* John Wiley, Chichester, U.K.

MEADOWS, D. H., J. RICHARDSON AND G. BRUCKMANN. 1982. *Groping in the Dark: The First Decade of Global Modelling.* John Wiley, Chichester, U.K.

MISER, H. J., AND E. S. QUADE (eds.). 1985. *Handbook of Systems Analysis: Overview of Uses, Procedures, Applications, and Practice.* North-Holland, New York.

MISER, H. J., AND E. S. QUADE (eds.). 1988. *Handbook of Systems Analysis: Craft Issues and Procedural Choices.* North-Holland, New York.

QUADE, E. S. 1980. Pitfalls in Formulation and Modeling. In *Pitfalls of Analysis*, G. Majone and E. S. Quade (eds.). John Wiley, Chichester, U.K., 23–43.

QUADE, E. S. 1985. Predicting the Consequences: Models and Modeling. In *Handbook of Systems Analysis: Overview of Uses, Procedures, Applications, and Practice*, H. J. Miser and E. S. Quade (eds.). North-Holland, New York, 191–218.

RAIFFA, H. 1982. Policy Analysis: A Checklist of Concerns. International Institute for Applied Systems Analysis, Laxenburg, Austria, 82–92.

RONEN, Y. (ed.). 1988. *Uncertainty Analysis.* CRC Press, Boca Raton, Florida.

ROSE, D., AND V. DOBSON. 1985. Methodological Solutions for Neuroscience. In *Models of the Visual Cortex*, D. Rose and V. Dobson (eds.). John Wiley, New York, 533–560.

ROTHENBERG, J., N. Z. SHAPIRO AND C. HEFLEY. 1990. A 'Propagative' Approach to Sensitivity Analysis. In *Proceedings of the AI, Simulation and Planning in High Autonomy Systems Conference*, B. Zeigler and J. Rozenblit (eds.). IEEE Computer Society Press, Los Alamitos, California, 10–16.

SCHRAGE, M. 1989. Why Beauty of Scientific Models Is Often Only Skin Deep. *Los Angeles Times,* December 21, 1989, p. D1.

SCHROEDER, M. 1991. *Fractals, Chaos, Power Laws.* W. H. Freeman, San Francisco.

SHAPIRO, N., H. E. HALL, R. H. ANDERSON AND M. LACASSE. 1985. The RAND-ABEL Programming Language, History, Rationale, and Design. R-3274-NA, The RAND Corporation, Santa Monica, Calif.

SHAPIRO, N., H. E. HALL, R. H. ANDERSON, M. LACASSE, M. S. GILLOGLY AND R. WEISSLER. 1988. The RAND-ABEL Programming Language: Reference Manual. N-2367-1-NA, The RAND Corporation, Santa Monica, Calif.

STOCKFISCH, J. A. 1975. Models, Data, and War: A Critique of the Study of Conventional Forces. R-1526-PR. The RAND Corporation, Santa Monica, Calif.

STRAUSS, C. M. 1974. Computer-Encouraged Serendipity in Pure Mathematics. *Proc. IEEE* **62,** 493–495.

SURI, R. 1987. Infinitesimal Perturbation Analysis for General Discrete Event Systems. *J. ACM* **34,** 686–717.

SURI, R. 1989. Perturbation Analysis: The State of the Art and Research Issues Explained via the GI/G/1 Queue. *Proc. IEEE* **77,** 114–137.

WURMAN, R. 1989. *Information Anxiety.* Doubleday, New York.