WHAT IS A COMPUTER-BASED MATHEMATICAL MODEL?

SAUL I. GASS

College of Business and Management, University of Maryland, College Park, MD 20742

(Received May 1983)

Abstract—Many researchers feel that one's ability to formulate a mathematical model is more art than science. Even so, modern applied mathematics is centered about the role of the mathematical model. Texts are now available and courses are being taught that introduce the basic principles and methods of mathematical modeling. This paper emphasizes that the resolution of most complex modeling problems, especially those in the policy area, are solved using a computer-based model. Thus, there is a need to stress the often overlooked model development steps of verification, validation, and evaluation. This paper reviews some of the recent research in these areas, and notes that important research in these topics is required to put mathematical modeling on a more scientific basis.

In his questioning paper “What is a Computer Program?” Gemignani [16] raises the issue of our apparent inability to state an unambiguous definition of a computer program. Such a definition would have to satisfy the needs of the scientific, legal and related professions. It is a most challenging dilemma. But, to my mind, the more important concern to mathematicians is the associated problem of defining what constitutes a computer-based mathematical model.

The introduction of core courses in mathematical modeling appears to be the trend of the 1980's. This is attested to by the report of Protomastro and Hallum [31], the work of Spanier [35,36] and the many recent texts dealing with the subject; see, for example, Bender [2], Burghes and Wood [4], Dym and Ivey [6], Maki and Thompson [26] and Roberts [33]. As we instruct our students in this field, we need to have them develop a proper appreciation for how such modeling is done in practice. The textbook “toy” examples and the pencil-and-paper approach to solution procedures do not convey the fact that most applications of mathematical models are rather large and complex entities that can only be solved by using a computer—which implies the need to develop a computer-based model. With the increased use of computers in the classroom, this aspect of problem-solving can be introduced by hands-on case studies. But the requirements of the curriculum and skills of the students (and faculty) leave little time to develop fully the implications of relying on a computer-based mathematical model to solve a problem. In this note we describe some of these concerns and those raised by the title of this paper, and indicate directions recent research has taken to alleviate them.

A “standard” description of how we go about developing a mathematical model is conveyed by the seven stages exhibited in Fig. 1, Burghes and Wood [4]. With respect to this diagram, the authors explain: “The lefthand column represents the real world, the right-hand column the mathematical world, and the middle column the connection between these two worlds where firstly the problem is simplified and formalized and secondly the mathematical solution is translated back into the real world situation.” Note the feedback aspects implied by the dashed lines, i.e., model development is an iterative process. An expanded view of the modeling process, one that is more detailed and exact with respect to actual practice, is given in Fig. 2, U.S. GAO [38]. Here the steps of Fig. 1 are included, but the emphasis is on the translation of the mathematical statement into a verified and validated model that is solved using a computer.
Model verification is the process of (1) ensuring that the computer program, as written, accurately describes the model as designed, (2) ensuring that the program is properly mechanized on the computer, and (3) ensuring that the program as mechanized does run as intended, Fishman and Kiviat [8], House and McLeod [23].

Verification tests fall into two somewhat indistinct categories: (1) experiments to debug the logic of the computer program and (2) checks to demonstrate the correctness of the numerical and data procedures as carried out by the program. There is general agreement that for large and complex computer programs that it is impossible to demonstrate that a program has been completely verified, i.e., there will always be bugs ready to exhibit their ugly behavior. Verification tends to be the domain of programmers, with some assistance from analysts in devising numerical tests. Some computer scientists advocate replacing verification with the concept of program reliability, i.e., the program works for all practical purposes even though it is not perfect, DeMillo et al. [5]; see also Griffin [19,20] and his concept of program level of confidence. How to measure computer program reliability and, as we shall see, confidence in a model are open research questions. It comes as a blow to one's scientific sensibilities that we sometimes must use an imperfect tool (a computer program) to solve a problem, and there is a chance that the results may not be reproducible by others. For example, I know of an instance in which a FORTRAN compiler resulted in a forecasting model producing an answer different from the one obtained using the previous compiler. The mathematical model/computer program relationship is even more frustrating. Mitchell and Wilson [28] give an example of slightly different programmed versions (both "correct") of the same algorithm that gave significantly different results with the same data.

Model validation tests the agreement between the behavior of the model and the real world system being modeled, Fishman and Kiviat [8]. How to measure the validity of a model is
BASIC STEPS IN THE MODELING PROCESS

- Describe Problem
- Isolate System
- Adopt Supporting Theory
- Formulate Model
- Analyze Data Requirements
- Collect Data
- Develop Computer Program
- Debug Computer Program
- Develop Alternative Solutions
- Evaluate Model Output/Results
- Present Results/Plans
- Develop Model Maintenance Procedures
- Transfer System to Users

**VERIFICATION**

Computer model verification examines whether the computerized model "runs as intended" depending on available documentation. Verification may have to examine several of the steps in the modeling process.

**MODEL VALIDATION**

Validation examines the correspondence of the model and its outputs to perceived reality. Depending on the available documentation, validation may have to examine or assess many of the steps in the modeling process.

Fig. 2. Basic steps in the modeling process from U.S. GAO [38].
an open question: The answer depends on the real world aspect being analyzed, the type of model being used, who is asking the question, and who will interpret the answer. Models of the physical world or in-beings physical entities (e.g., oil refineries) can usually be checked for physical validity, Emshoff and Sisson [7], Orden [30]. Validity here means that the model results can be shown to replicate the physical world, past experiences, or to work within the real world environment. But what of models of non-existent systems or of future situations (e.g., a transportation system for a metropolis in the year 2000, a new automated factory design, or a land use plan) in which the past may not be a good predictor of the future? For such first-time or future models validity is superseded by the concept of model credibility, as defined by the decision maker (Hermann [22], Emshoff and Sisson [7]). There is still a deeper concern for those models of problem situations (current or future) that include a human behavioral content (e.g., traffic flow, econometric forecasting, or national energy policy planning) in that we really do not understand the “behavioral laws” that make the real world go around. This is especially true for problems that involve public policy and business policy issues. As noted by Strauch [38], “...problems encountered in policy analysis run the gamut from well-defined to highly ‘squishy’. A well-defined problem is one that can be given a clear-cut, well-defined formulation, amenable to rigorous analysis. A squishy problem is one with the property that any clear-cut, well-defined formulation of it will look like an unambiguous representation of the substantial problem only so long as we don’t lean on it, or question it too carefully or deeply. ... Problems with any significant degree of behavioral or political content tend to be squishy.” During the past decade mathematical models have been used extensively to analyze almost all areas of public and business policy, e.g., energy, multi national corporations, pollution, water resources, military, welfare, performing arts, housing, education. Not all, in fact only few, such models can be said to have been successful, i.e., used by their sponsors as decision aids (Gass and Sisson [15], Brewer and Shubik [3], Fromm et al. [9], and Greenberger, Crenson and Crissey [17]). A major reason for this lack of success has been the question of model validity.

As with program verification, analysts do not believe that a model can be completely validated. Quade [32] states: “A particularly dangerous myth is the belief that a policy model can be fully validated—that is, proved correct. Such models can, at best, be invalidated.” With respect to the modeling process of Fig. 1, Burghes and Wood [4] note: “In many cases, particularly in the social sciences, it is difficult to apply box 6 (validate model) at all, and we move straight from box 5 to box 7.” How can the use of such a model then be justified, especially if it is one to assist you in making a decision?

The process of model validation is not precise, especially for squishy and policy problems. Most relevant literature on validation is due to simulation researchers, see Hermann [22], Naylor and Finger [29], Van Horn [39], and Schellenberger [34]. When you develop a new mathematical model, especially one to be used as a decision aid, you should bear in mind the needs of validation (as expressed by these authors) and make the validation process a continual part of the model developmental process (as shown in Fig. 2). A most challenging part of any model formulation and its testing on a computer is the construction of mathematical procedures and techniques that “prove” validity. Here the analyst must develop tests of validity that the analyst and users think are important. Some such tests, many of which have been of continuing interest to mathematicians, concern model sensitivity analysis, robustness, and statistical measures of confidence. Analytical measures of sensitivity and confidence depend, of course, on the model’s structure and assumptions. Recent research includes, among others, that of Labys [24], Mitchell and Wilson [28], Balci and Sargent [1], McKay [27], and Harris [21]. Although most of this work has concentrated on validation of energy models, there is no reason why it cannot be carried over to models in general. A key question, given that there are no model-based analytical representations for sensitivity, is how to devise a limited (and a usually costly) set of computer runs of a model that yields the most information. Which parameters
What is a computer-based mathematical model?

are varied and in what combinations? What if the model passes information to another model, and so on? How is sensitivity measure here?

Most mathematical models of policy problems are decision aiding models (as defined in the Operations Research (OR) sense). Verification and validation are, of course, important issues for users of such models. But beyond these concerns, there is the need for making the OR-based methodologies and their models more useful to a decision maker. This has led to the development of the concepts and processes of model evaluation (or assessment) and model confidence (or credibility).

Model assessment has been defined to include (1) verification, validation, and quality control of the usability of the model and its readiness for use and (2) investigations into the assumptions and limitations of the model, its appropriate uses, and why it produces the results it does, Greenberger and Richels [18]. Assessment is a process by which interested parties, who were not involved in a model's origins, development and implementation, can assess the model's results in terms of its structure and data inputs so as to determine, with some level of confidence, whether or not the results can be used in decision making, Gass [10, 11]. There are many proposed methods for carrying out a model assessment, Wood [40].

All of them tend to rely on quantitative techniques whenever possible, but the final assessment statement is usually a qualitative judgmental summary. Whether the process can be made more rigorous is an open question. What should a model's assessment report card state? First, we emphasize that model confidence is not an attribute of a model, but is an information-based opinion (that relies on the assessment report) or a judgment made by a particular user for a given decision environment. Thus, the level of confidence may vary from user to user. Second, confidence in a model evolves by a joint effort between the model developers and a designated user. A decision model without a designated user (which implies specific use) has no basis upon which a confidence statement can be made. An approach to quantifying the assessment information into a user's confidence statement is given in Gass and Joel [14]. We stress that there is no proven way for doing this. The nature of model confidence measures is also an open question. Research on model assessment and confidence has concentrated on energy models used by the Department of Energy, Gass [12]. But the requirements for the successful use of policy models in other areas of government and business call for the application of similar concepts.

Well, what is a computer-based mathematical model? In one sense it is whatever is represented by the computer program and input data that resides in a particular computer system. That is the easy answer. But is that programming system what the analysts had in mind? What the programmers thought the analysts wanted and what the programmers thought they wrote? What a user needs to solve the user's problem? Is the programming system reliable? Has it been verified? Has it been validated? Can it be run on a different computer? Can it be used for a new application that requires parameter changes? And so on and so on! We need ways of answering all these questions before we can answer the question posed by this paper's title.

REFERENCES

32. Quade, E. S., “Pitfalls in Formulation and Modeling,” in Majone and Quade [25].