PRELIMINARIES

François E. Cellier
Institute for Automatic Control, The Swiss Federal Institute of Technology, Zurich

In recent years, simulation techniques have become more and more accepted as an invaluable tool for system analysis in a wide variety of different disciplines. Nowadays, hardly any scientific discipline could be mentioned which does not make use of simulation in one way or another. It would, however, be incorrect to conclude from this fact that the demands, as asked for by different disciplines, are necessarily the same throughout the wide spectrum of applications. It was, perhaps, one of the most tragic failures in simulation that in the early days of computerized simulation, a (well justified) enthusiasm concerning simulation results in engineering applications (so-called "hard" sciences) had led to the assumption that the previously so successful methodology could be used without modification to deal with biological or social systems (so-called "soft" sciences) as well. In fact, the demands are quite different as shall be shown with a very simple example.

Current simulation software provides for a very powerful behaviour generation, that is, given a model together with values specified for all parameters of the model, and given a particular experiment to be performed on that model, time histories (trajectories) of all state variables in the model are computed with good confidence and are displayed with high precision. This is certainly what is needed for most engineering applications (e.g. electrical network analysis). The models themselves have a high quality and, thus, the results from simulation studies performed on such models can be expected to be trustworthy. However, the same behaviour generation mechanism is entirely inacceptable for most soft science studies. The models obtainable there are of low credibility, the parameters are known at the best to some percent, and even the structure of the model is often more than questionable. Under such circumstances, it makes no sense at all to display results with 14 digits if we know that most probably not a single one of those is significant. However, the (in most cases rather credulous) user is inclined to validate the obtained results by the fact that 14 digits are displayed to him out of which not a single one is zero. In fact, one of the major reasons for the somewhat limited credibility which is assigned to simulation results in many domains lies in the credulity of their end users. As it is, in general, easier to change the software than to change its users, we should provide for means of adjusting the format of the displayed results to the application the software is used for. For instance, if the user would be asked to specify on input tolerances for all parameters of his model (which, in most cases, would not be too difficult to do), the software could perform an automated sensitivity analysis, and select dynamically an acceptable format for all output displayed. Even if no significant digits remain, it still makes sense to present results; however, rather than displaying a single curve, the simulation software should display a bandwidth (confidence interval) in which the results are expected to be found. Such a display could tell the user much more than a rather incidentally selected single curve. The so-called ill-definition of models from soft sciences, therefore, calls for a quite drastic modification of the modelling and simulation methodology. Obviously, the here mentioned example is just one among many aspects in this field.

Concerning biological systems in particular, some steps have meanwhile been taken towards a unifying methodology, and the chapter presented by *G. C. Vansteenkiste* discusses this issue in detail. To appreciate how deeply simulation techniques have influenced biological research already, the chapter presented by *J. R. Barrett and R. M. Peart* presents an impressive bibliography on just one among many biological topics, namely agriculture.

In biological sciences, the ill-definition of models results from among other things problems with taking appropriate measurements. As human beings (that is, the investigators) are to be considered as biological systems themselves, it is further not astonishing to notice that the time constants dominating the systems under study are frequently of the same order of magnitude in the investigator (that is a life-time). This fact certainly acts as a tremendous obstacle against the provision for reasonably accurate and validated models, in particular, if one considers that modelling (as we understand the term today) is an art which does not date back an entire life-span yet. However, I am fully convinced that reasonably accurate models for these types of systems would in theory exist even if neither we nor our direct descendants were able to determine them.

In the social sciences, the situation is complicated by some additional troubles. When human beings play an important role in the system under study, the systems often change their behaviour as they notice that they are modelled. I call this the keep-smiling effect (people tend to change their behaviour if they notice that a photographer is sneaking around trying to take their picture). This is certainly nothing new. At least since the days of Heisenberg, we know that it is impossible to observe a system without disturbing it. However, the disturbance may be somewhere in the twentieth decimal, and then we may easily neglect this disturbance, or it may be a significant change of the system behaviour as a whole. Unfortunately, in social sciences, the latter is usually the case, that is, the keep-smiling effect often is the dominating factor in the system behaviour. For this reason, I am rather doubtful that for all imaginable systems from social sciences, acceptable models would even in theory exist, independent of the question of whether we can determine them or not. For this reason, one must even be more cautious with models in the social sciences, and it is quite seductive to draw illegitimate conclusions from results of social system studies based on dubious models which have not been cautiously validated for their task. The contribution by *K. Leimühler* discusses these aspects in detail, and opens new perspectives as concerning possible means to overcome these difficulties. Certainly, it is one of the most challenging tasks in modelling today to investigate the problems resulting from ill-definition and to find new methodologies to overcome these problems. For this reason, an entire section of this book deals with the subject matter.

In addition to ill-definition, however, there exists a second problem in many soft science studies which is often not even easily separable from the previous. When modelling an electric resistor, for instance, it is very easy to postulate a model which is acceptable for a wide variety of experiments. Nevertheless, it is still just a model which should not be mixed up with the real system itself. If, for example, the experiment would include a variation of the surrounding temperature in a large range or would include very high frequences, the same model would no longer be acceptable. Fortunately enough, such experiments are seldom, and, for most applications, the simple model $U=R*I$ will do. In the soft sciences, however, it is for the most part much more difficult to separate out the importance of different potential input variables both in the time scale as well as in the space dimensions. This aspect is also discussed carefully in the chapter by *G. C. Vansteenkiste.* If one models the growing speed of a flower in spring, this shall depend on many influencing factors like temperature, photoperiod, air and soil humidity, nutrients in the soil, solar radiation, wind velocity, rain, microbial activities of the soil, and so on. Unfortunately, all those factors have effects of the same order of magnitude. For almost any kind of experiment, we should take them all into account. This calls for rather complex models with many input variables which are often only to a small extent observable and to an even smaller extent controllable. We call these models large-scale models. Quite obviously, such large-scale models are again more difficult to identify, making them even more ill-defined.

The second section of this book deals primarily with different aspects of modelling large-scale systems. The contribution by *K. Frohberg* discusses the example of the difficulties of world food supply. Typical keywords are hierarchism and regionalization, keywords which are also central to the discussions by *N. Müller* and by *A. Sydow. N. Müller* discusses hierarchical concepts of social systems, whereas *A. Sydow's* chapter is a little more technically oriented both in its presentation and selected applications.

Certainly, the best established methodology for large-scale system modelling is System Dynamics, a methodology we owe to J. Forrester. The presentation by *J. D. Lebel* discusses some aspects of System Dynamics models and, primarily important, contains an impressive bibliography with more than 700 entries on applications of this methodology.

As we have already seen, models must never be confused with reality. They just map some facets of reality (hopefully the ones we had in mind) into an abstract description. It is, therefore, of utmost importance to cautiously validate the obtained models for their respective tasks. This is usually a quite simple job when dealing with one or two differential equations and three or four parameters. It becomes, however, tedious and extremely difficult to perform when we deal with large-scale models. The chapter by *R. G. Sargent* surveys a large variety of aspects of validation and of means to achieve it. This chapter furthermore contains a discussion of the verification of simulation results: a topic which would be better included in the second part of this book; however, as the techniques used are very similar to those used in validation, it did not seem justified to split up the presentation of these topics into two separate chapters. Incidentally, the same holds true for the presentation by *A. Sydow* the first part of which deals more with large-scale system simulation rather than large-scale system modelling (and is somewhat related to the chapter by *E. Eitelberg*). However, in reality, it is often not so easy to clearly separate the different aspects out, and the contributions have therefore been placed where we felt that their major emphasis lay.

In many models, statistical aspects have an important role to play. As it was mentioned above for the case of models from soft science, results of simulation studies involving stochastic signals are not so easily interpretable. Again, one single curve doesn't tell very much. The question thus raises what possibilities exist to interpret the obtained simulation results correctly, and how the statistical experiments should be designed to allow (with minimum computation) an appropriate interpretation of the results. The chapter by *J. P. C. Kleijnen* surveys a variety of methods for this purpose.

Also previously postulated was the need for an automated sensitivity analysis. Obviously, this is just one among many potential fruitful extensions to the currently available simulation software. Current software always assumes that a model already exists, and restricts itself to pure behaviour generation. This ought no longer to be the case in the future. The computer should help the investigator both in the preparation of the model and in its validation. To distinguish from the currently available simulation software, we would rather want to call this "modelling software". The contribution by *T. I. Ören* lists a large variety of possibilities for such software. Obviously, not all of these possibilities shall be easily integratable into one system without making it so large, slow, and unmanageable that we had better forget about it all together. This is, however, not the intention of this contribution, as the author expressed clearly during his presentation. This contribution aims at systematizing possibilities for such systems from a methodological point of view.

As the models which are investigated tend to become larger and larger, it is quite obvious that there exists an increased risk of modelling errors which remain undetected due to the fact that the models we deal with are no longer very transparent. One possible answer lies in the hierarchism discussed above; another answer may be found by changing the model description mechanisms, e.g. by using graphical techniques. Two possible solutions are discussed for continuous systems by *J. J. van Dixhoorn,* and for discrete systems by *C. D. Pegden and A. A. B. Pritsker.* These two contributions conclude the first part of this book which deals with a diversity of topics in modern modelling theory.

The second part of this book is devoted to simulation issues. The large-scale aspect does not only influence modelling but simulation as well. It is, for instance, a quite general statement that large-scale models of continuous systems are stiff. These models require, therefore, particular techniques for integration. Again, partitioning mechanisms may provide one possible answer. However, this can in many instances not be done in the same way as described under the heading "modelling" as the partitioning should be done here for the benefit of the numerical algorithm, and not for the user of the model. It would, for instance, be very fruitful to partition the problem into a fast and a slow part. However, in many applications, it is not so easy to determine which equation belongs to the fast and which to the slow part. Actually, the eigenvalues of the Jacobian may tell something, but it is no trivial task to relate those eigenvalues back to the differential equations from which they were generated. Moreover, in a nonlinear case, it may well happen that the eigenvalues move around, and one and the same mode may belong for a time to the fast and then again to the slow subsystem. It is, therefore, one of the "hottest" research problems in numerical mathematics to develop an algorithm which would allow an automated partitioning of the differential equations to be done. The chapter presented by *M. B. Carver* discusses this matter and presents a partial solution to it. The problem must still be considered unsolved as no algorithm could be found which would work equally well for all application problems. Furthermore, the algorithm presented by *M. B. Carver* leaves some parameters for tuning to the user and it is not entirely clear how they should be properly chosen. This is, however, not at all meant to be a criticism, as the problem seems to be extremely difficult to solve; I am personally very sceptical as to whether a completely automated algorithm can be developed which works under all circumstances satisfactorily. The contribution by *E. Eitelberg* is somewhat related to the previous one. In this contribution, the aim is not an automated partitioning. Here, the partitioning is left completely to the user. However, the algorithm presented by *E. Eitelberg* allows for a modular simulation of once partitioned systems. This approach seems particularly fruitful for large-scale system simulation as models of this type are usually not developed once and then used, but they are steadily extended, modified, and adapted to changing demands. It is then more than welcome if previously determined structures can be maintained, and only those modules which underwent modifications have to be recompiled.

A very prominent class of large-scale models are those large sets of ordinary differential equations which result from applying the method-of-lines to partial differential equations (that is, to distributed system models). A quite large number of software packages have been developed in recent years to deal with a variety of classes of such systems (elliptic, parabolic, hyperbolic). One chapter by *W. J. Karplus* surveys those packages. It was felt that there exist many books already on the market in which the numerical methods as such are thoroughly discussed. However, only a very few surveys on the available software exist. This fact makes the contribution by *W. J. Karplus* particularly valuable as many users who are not mathematicians themselves may not have the skill to tackle their problems just after reading a book in which the numerical techniques are described. The subject matter is numerically extremely delicate, which makes the use of precut rather than home-tailered software recommendable wherever applicable. However, a word of caution is necessary as M. B. Carver pointed out in one of his software manuals describing the FORSIM system, one of the biggest dangers in using the FORSIM software is in the ease of its applicability. It is so easy to formulate a problem by use of this type of software that many people tend to use it without the necessary caution. The result may be a printout of very expensively computed and at the same time completely ridiculous results. The advantage of PDE software lies in the fact that it is very easy to experiment with the software. A change of just one parameter value may call for the execution of a completely different numerical algorithm. It also relieves the user from debugging large and intricate programs. It does not relieve the user, however, from understanding what he does. The chapter by *S. G. Tzafestas* surveys hybrid methods for the solution of this type of problem. The particular value of this contribution lies in the fact that it may open new perspectives as to the diversity of available algorithms for the solution of PDE problems. Although the chapter discusses hybrid computer solutions, the algorithms are described on such a level that the hybrid implementation in the end may be considered a matter of technicality. Also readers who have no hybrid installation at their disposal, and who have no intention at all of going hybrid may profit a lot from reading this contribution as the described methodology may lend itself as easily to other implementations.

The fifth and last chapter in this section dealing with large-scale system simulation discusses a slightly different topic from the previous ones. A fact which was often overlooked in the past is that large-scale models not only imply larger structures or more differential equations, but also more parameters, larger amounts of input, transitory, and output data of any kind. These large amounts of data may create a tremendous headache if the software does not provide for appropriate data-handling mechanisms. This suggests a comfortable link between the simulation software on one side, and a powerful data base management system on the other, a data base management system which has been adapted to the needs of simulation users. Such a solution is presented in the chapter by *C. R. Standridge and A. A. B. Pritsker.*

Although under separate heading, the next chapter also has to do with large-scale problems. As the systems under study, and thus also the models to be constructed, become more and more complicated, it is very essential that the simulation software is designed carefully and in a very systematic manner. If one looks into the manuals of currently available simulation languages, one finds explained in each of them how neatly the software may be used to solve the Van-der-Pol equation with it. Unfortunately, a software tool which is very appropriate for the Van-der-Pol equation, the pilot ejection study, and some large-fish-small-fish example is not necessarily also acceptable for large-scale studies. This can easily be shown with an example. Most of the currently available simulation languages do not provide for a mechanism for the declaration of variables. Such a mechanism would certainly not be attractive for a very short program, as it would make this short program unnecessarily long; short programs are very attractive, too. However, for a large-scale study, the same mechanism becomes vital as redundancy introduced in this way helps remarkably to detect all sorts of typing errors during an early stage of software development. If one has read the study once published by IBM in which IBM tried to relate the cost for removal of a software error to the time when it was detected (an error being removed during software development is assumed to cost one monetary unit; the same error when removed during software testing costs 20 monetary units; during production, it costs 120 monetary units), one is inclined to accept the discomfort of coding slightly longer programs, if, by these means, the probability to detect errors earlier increases. The chapter by *G. Rzevski* surveys some methodological aids for a systematic design of simulation software. As the topic is rather complex, not all possible means are mentioned; however, more may be found in the references cited in this chapter.

The final section in this book deals with simulation systems, that is, with complete ensembles of software and hardware for simulation purposes. The contribution by *R. Crosbie* summarizes the major ideas behind interactive and real-time simulation. The chapter by *W. Ameling* discusses architectural considerations which go into the design of special-purpose simulation hardware. In particular, when dealing with real-time simulation studies, conventional digital computers may be too slow (or too expensive, e.g. CRAY-1) for the task. Then, special hardware is required. One obvious solution may be to go hybrid, but other architectures exist as well. The idea common to all these solutions is to compute as many pieces of the algorithm as feasible in parallel.

One particularly attractive solution is the use of array processors. These devices are discussed in the second chapter by *W. J. Karplus.* As the architectural considerations which go into the design of array processors are meanwhile well established and have been discussed on many occasions, *W. J. Karplus* was asked to specially concentrate on the software issues. In a parallel processing system, each processor will, in general, require a program of its own. Those programs must then be synchronized to execute one complex task in parallel. So far, the weakest point in the use of such systems lies certainly in the programming aspects of it. How should the task be partitioned to yield optimal throughput? How do the programs for the single processors look like? How is the inter-processor communication realized? How do they share common data? This question is not at all solved to a satisfactory extent to date. In most cases, it means going back to old-fashioned assembly programming, and the time needed for development of software expands to entirely unacceptable numbers for most applications. A certain break-through has been achieved now in the use of array processors, although there is nothing more certain than the fact that the available solutions are not yet the final word to this discussion. The chapter by *W. J. Karplus* describes the current state-of-the-art, and is rather unique in that hardly any discussions of software issues for parallel processing devices can be found in the literature to date.

All of the presentations in this book have certainly (beside of their topic: simulation) another thing in common. This is the high standard of quality of the presented material both from the point of view of its contents as well as its presentation. During the reviewing and editing process, I tried to understand each contribution up to the last sentence. For this purpose, I fully relied on the text and did not try to dig out the references given in the contributions. As I believe I have achieved this goal, I am convinced that you, the readers, shall face no difficulty in doing the same.