

AN INTERACTIVE SYSTEM FOR MODELING

I. Galligani - L. Moltedo

Istituto per le Applicazioni del Calcolo "M. Picone", CNR

Rome, Italy

ABSTRACT

Recently some authors have proposed to introduce a pattern recognition approach in the modeling process, especially for the study of populations of species, within a compartmental representation, in aquatic ecosystems and for the water pollution control.

In order to implement this approach on a computer, it is necessary to develop "interactive systems" which are composed by a special language for modeling, a collection of data management procedures and a collection of numerical procedures. These systems give the possibility of integrating the data base handling techniques with mathematical methods for constructing models in an interactive manner in order to take into account the analyst's appreciation and understanding of the determining features of the prototype system during the different stages of the modeling process.

In this paper, we describe the main characteristics of such an interactive system with graphical facilities designed for a minicomputer which includes different algorithms for integrating ordinary differential equations. These algorithms have been chosen after an analysis which was not only oriented to the selection of the most significant methods but also to the study of their feasibility within a procedure which gives local and global error estimations. Some "standardization" problems in the implementation of this system have been taken into account.

INTRODUCTION

Various models have been proposed for the study of populations of species in ecosystems, within a compartmental representation, for the management of underground water reservoirs and for the improvement of the water quality. Most of them are lumped parameter models in which we have to identify the state-variables and the parameters. Direct and indirect methods have been developed for solving these identification problems. The indirect methods are essentially trial and error procedures which seek to improve an existing estimate of the variables and parameters in an iterative manner until the model response is sufficiently close to that of the real prototype system. This improvement generally is accomplished with the aid of empirical criteria or by formal mathematical procedures (gradient algorithms, random searches, etc.). However the indirect methods are effective only for a limited class of identification problems. They often break down, especially when the "starting" model is not a sufficiently close representation of the prototype system and the excitation-response data available from the observations of the reality are of low quality. Besides they do not adequately utilize the large amount of potentially-valuable and useful information contained in the excitation-response data available from experiments or observations of the prototype system. Thus some authors have proposed to introduce a pattern recognition or learning approach in the modeling process (see, for example [1] , pg.32 and [2]).

In order to implement this approach on a computer, it is necessary to develop "Interactive Systems"⁽⁺⁾ which are composed by:

- a special language for modeling;
- a collection of numerical procedures for solving classes of problems;
- a collection of data management procedures which provide a comprehensive set of data base management capabilities, including the ability to define new data bases, modify the definition of existing data bases,

(+) - We note that the word "system" is used in two different senses (both allowed due to common usage): physical system which we observe and computer software system.

retrieve and update values in these data bases and to extract from the data base those items that are used by the numerical procedures.

Indeed the advantages accrued by working on an interactive system with graphical capability are the very fast turnaround, the immediate graphical display, the simplicity of the control of the calculations and the possibility of various attacks on the problem and of various choices of the mathematical method. Especially for this "possibility", the analyst's appreciation and understanding of the prototype system's determining features may be taken into account during the different stages of the modeling process.

The construction of an interactive system with an effective high level language for modeling and with a well structured package for integrating data base handling techniques with mathematical methods is a very difficult task, especially when it is required to implement such a system on a minicomputer.

In this paper we describe the main characteristics of a "special" interactive system with graphical facilities designed to work on a minicomputer for solving ordinary differential initial value problems.

In this system we have restricted at the maximum the "descriptive" aspects of the modeling process, by designing the interaction to require only very simple actions by the user, which are anticipated by the system.⁽⁺⁾ The interaction by anticipation on a "special" system has been taken into consideration in order to attempt to solve to some extent the dilemma between the system's "effectiveness" versus the "simplicity of use". Notwithstanding this restriction, with this system it is possible to develop in reasonable computing time many "significant" lumped parameter models for different application areas, as dynamics of populations, environment, chemical kinetics, etc.

The system has been made self-helping and self-explanatory by giving the possibility to inquire optionally many tutorial displays.

The implementation of this system on a minicomputer allows the user to test simply and economically the above models in-house.

(+) - The interaction by anticipation allows the user to select a desired action rather than specify that action.

The system includes a rather great number of modern numerical procedures for solving initial value problems associated with ordinary differential equations. These algorithms take into consideration many different factors, as stiffness, cost of function evaluations, perturbation to linearity, estimation of the time-constants, etc., and are implemented within a procedure with automatic step-size and order determination.

The interactive system has been designed and written with portability in mind. For this the system has been written in Fortran by using the graphics package MINING [3]. With the MINING package a high degree of machine independence and an easement of programming have been achieved. (For example, the layout of each picture is done automatically; there are many automatic features, as "zooming", movement of items behind the "viewing window", etc.).

The machine-dependent subroutines on which MINING is based are constructed with some form of standardization.

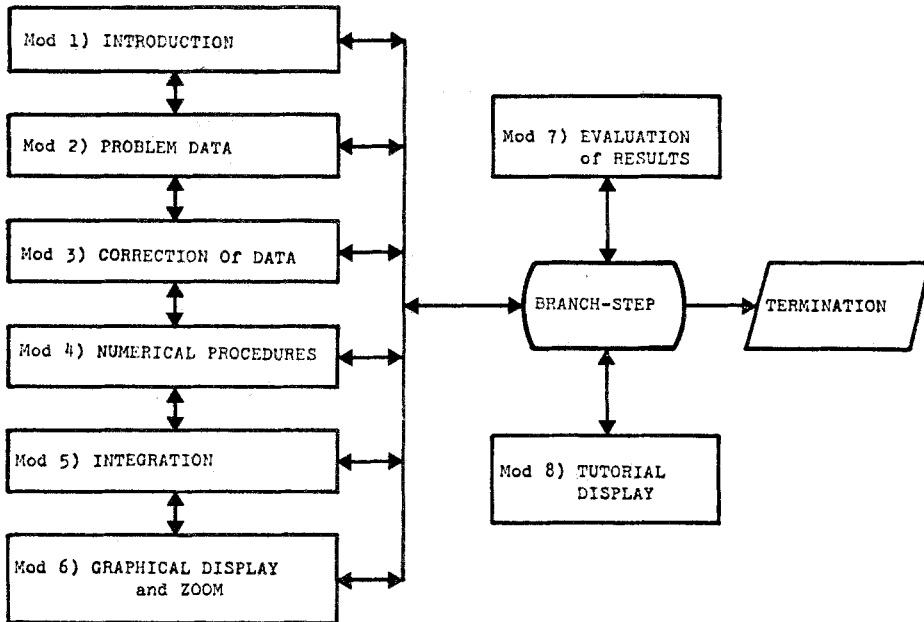
Besides all the numerical procedures reflect a same structure in order to make these programs relatively easy to maintain or to modify.

Recently some systems similar to that described in this paper (see, for example [4] and [5]) have been developed, but in none of these all the above features are taken into account at the same time.

2. THE INTERACTIVE SYSTEM

2.1 Structure of the System

The interactive system performs a certain number of logical distinct activities which can be depicted by the following flowchart. At each block of the flowchart, i.e. module of the system, corresponds a logical distinct activity.



The connection among the different modules is performed through the three commands CONTINUE, BACK and BRANCH-OUT, which transfer the program control to the next, previous and Branch-Step modules, respectively. The Branch-Step module plays a significant role in the management of the modules.

The numerical procedures included in the system solve the initial value problem, written in normal form:

$$\underline{y}'(t) = \underline{f}(t, \underline{y}(t)) \quad t \in (t_0, t_f] \quad \text{with} \quad \underline{y}(t_0) = \underline{y}_0 \quad (\text{A1})$$

In module 1, INTRODUCTION, some introductory remarks are given. These

remarks briefly summarize the flow of the program, give the built in limits on the number of equations, parameters, initial values, etc., and facilitate the user to verify the correct storage of the user's subroutines DE, JAC and FUNCT, which describe the function $f(t, y(t))$, the Jacobian matrix $P_y(t, y(t))$ and the objective function, respectively. A typical objective function is $\|y(t) - \tilde{y}(t)\|^2$, where $\tilde{y}(t)$ is an "estimate of the solution" $y(t)$. In a same job, it is possible to declare many quite different functions.

In module 2, PROBLEM DATA, the values of the variables appearing in $f(t, y(t))$ which are susceptible of variations during a parametrical study and the initial conditions of the problem (A1) are assigned.

The module 3, CORRECTION of DATA, allows corrections of some problem data, as modification of the values, deletion and addition of some data. This module is useful for parametrical studies of the problem.

In module 4, NUMERICAL PROCEDURES, the choice of the method is performed either automatically or by the user and the parameters of the method, as step-size, order, error bound, time-constants, etc., are assigned.

In module 5, INTEGRATION, the integration process is carried out. If there is an interruption in the integration process, some diagnostic messages are displayed; otherwise the user must assign the mesh-points, belonging to $[t_0, t_f]$, in which the solution $y(t)$ has to be represented.

In module 6, GRAPHICAL DISPLAY and ZOOM, the graphic displays of the solution $y(t)$ and, optionally, of the "estimate of the solution" $\tilde{y}(t)$ are performed.

In module 7, EVALUATION of RESULTS, the objective function is evaluated in correspondence of the current values of the problem data. In this module, it is possible to assign the values of an "estimate of the solution" $\tilde{y}(t)$ and to plot the values of the objective function in correspondence of different values of a same problem data.

The module 8, TUTORIAL DISPLAYS, contains many tutorial informations which may help the user to interact more profitably with the system.

With the module BRANCH-STEP it is possible to perform a set of different activities to allow initiation of a new problem or a new attack on the problem at hand. This module is reached in two ways: either in the

natural order of steps after Module 6 or by means the command BRANCH-OUT, which is available in all modules 1-8.

The module TERMINATION stops the process of solving different initial value problems, i.e. the job.

2.2 Numerical Procedures

The interactive system includes a rather great number of numerical procedures for solving initial value problems associated with ordinary differential equations.

These algorithms are based on well known methods which have been selected after a comparative experimental analysis characterized by the following main criteria:

- stability properties;
- cost of function evaluations;
- local truncation error expressions and possibility of obtaining good and not expensive global error estimates;
- reasonable criteria for choosing a proper step-size and order.

The results of this analysis have suggested to take into consideration the following algorithms.

- 1) An explicit Runge-Kutta method with a global error estimation based on the Stetter theorem [6]. For perturbed linear problems, including stiff ones, the Lawson formula [7] has been included. This method has been chosen in alternative to the explicit Runge-Kutta method with minimum truncation error bounds [8] and the explicit Runge-Kutta described in [9] §§ 2.8 - 2.10. We have also considered a variant of the formulas in § 2.10 suggested by Lawson for the error estimation and control.
- 2) A semi-implicit Runge Kutta method based on the A-stable Radau-type formula. An a-posteriori error estimate technique by Dahlquist [10] has been incorporated.
- 3) A multistep Adams-Bashforth-Moulton predictor-corrector method.
- 4) A method based on backward differentiation formulas developed by Gear.
- 5) The polynomial and rational extrapolation method by Bulirsch and Stoer.
- 6) The exponential fitted A-stable formulas by Liniger and Willoughby.

The implementation of the methods 3) and 4) is a modified version of the well known programs developed by Gear [11]. The implementation of

the methods 5) and 6) is a modified version of the programs included in [4]. The methods 2), 4) and 6), which are convenient for stiff problems, have been compared with the Jain's method [12]. While Jain's method is an A-stable of higher accuracy, it is costly to use. The method 5) is convenient for non stiff problems when function evaluations are not very expensive [13].

2.3 Implementation of the System

The interactive system has been designed for the minicomputer PDP11/40 with a Tektronix 4010 display with portability in mind. For this, it has been written in Fortran with the graphics package MINING [3]⁽⁺⁾; so the programming of the system was very easy, reaching a highly portable software. The machine dependent subroutines on which MINING is based are written in Fortran by using a Basic Graphics Package, which has been defined with some form of standardization by a CNR working group. The subset of subroutines of this package constructed for a display device with only Keyboard input (in the lowest common denominator approach [14]) is composed by:

- the "initialization and termination subroutines" INIT to establish communication between the user's program and the graphic display^(*) and FINIT to terminate the use of the display device;
- the "image generation subroutines" WIND to define the screen window, TPLOT to plot a dark or bright vector, CHOUT to display a character, EXTCHR to extract characters from a character string and ANSTR to display a character string;
- the "attention-related subroutines" CHIN to introduce by Keyboard a character and TINSTR to introduce by Keyboard a character string.

The subroutine CHOUT is used also to define the operating mode (alpha-numeric mode, graphic input mode, graphic plot mode, etc.) by addressing the proper control character.

(+) - Some few modifications to the original MINING package are carried out for its implementation on the minicomputer.

(*) - In this subroutine some "device-dependent" and "machine-dependent" informations must be made available.

The subroutines CHOUT, EXTCHR and CHIN are written in machine language. The subroutines TPLOT and WIND may be used in two versions: real and virtual tracking modes. Real tracking is intended to be performed on a "real screen" (i.e. the display screen) with the absolute coordinate system; virtual tracking is referred to a "virtual screen" (limited by the floating point precision of the computer) with an implicit cartesian coordinate system. This basic package may be extended with subroutines which take into account some special display features, as the cross-hair in the Tektronix 4010 display.

The numerical procedures of the system have been structured in a standard form, which reflect the pattern of the well known program DIFSUB by Gear [15]. They perform one integration step and can be subdivided into four main sections. The first concerns the initialization operations, the second the calculations for one step, the third the estimate of the accuracy of the result and the fourth the determination of the step-size and order for the successive step.

The formal parameter part (or "calling sequence") of the numerical procedures is standardized in the form suggested by Hull in [16].

3. SOME APPLICATIONS

a) An important problem in hydrology is to identify the parameters α_1 , α_2 and α_3 of the following non linear model describing the fluid flow in a river basin ($\varphi \equiv \varphi(t)$, $f \equiv f(t)$):

$$\frac{d^2 \varphi}{dt^2} + \alpha_2 \varphi^{\alpha_1} \frac{d\varphi}{dt} + \alpha_3 \varphi = f \quad t \in (t_0, t_f]$$

$$\varphi(t_0) = \beta_1 \quad \frac{d\varphi(t_0)}{dt} = \beta_2$$

An estimate $\tilde{\varphi}(t)$ of the solution $\varphi(t)$ is known.

If we put $y_1 = \varphi$ $y_2 = \frac{d\varphi}{dt}$ $y_3 = \alpha_1$ $y_4 = \alpha_2$ $y_5 = \alpha_3$, this problem may be formulated: to find a vector $\underline{y} \equiv \underline{y}(t) = (y_1 \ y_2 \ y_3 \ y_4 \ y_5)$ which extremizes the function $F(\alpha_1 \alpha_2 \alpha_3) = \int_{t_0}^{t_f} |y_1(t) - \tilde{\varphi}(t)|^2 dt$ along a trajectory of the equation set:

$$\begin{cases} \frac{dy_1}{dt} = y_2 \\ \frac{dy_2}{dt} = -y_1 y_2 y_4 - y_1 y_5 + f \\ \frac{dy_3}{dt} = \frac{dy_4}{dt} = \frac{dy_5}{dt} = 0 \end{cases} \quad (A2)$$

$$\begin{cases} y_1(t_0) = \beta_1 \\ y_2(t_0) = \beta_2 \\ y_3(t_0) = \alpha_1 \\ y_4(t_0) = \alpha_2 \\ y_5(t_0) = \alpha_3 \end{cases} \quad (A3)$$

This problem may be solved with the interactive system described in §2. by using the following iterative process.

Given an initial guess for the parameters α_1^0 , α_2^0 and α_3^0 , we solve the problem (A2) - (A3) and we calculate the objective function $F(\alpha_1^0 \alpha_2^0 \alpha_3^0) = \int_{t_0}^{t_f} |y_1(t) - \tilde{\varphi}(t)|^2 dt$. Now the parameters α_2^0 and α_3^0 are held fixed and we calculate for different values of α_1 belonging of the interval $[\alpha_1^0 - \delta\alpha_1^0 \quad \alpha_1^0 + \delta\alpha_1^0]$, $\delta\alpha_1^0$ is given, the objective function $F(\alpha_1 \alpha_2^0 \alpha_3^0)$. Let us plot $F(\alpha_1 \alpha_2^0 \alpha_3^0)$ as a function of α_1 alone. A minimum point on this curve, say at $\alpha_1 = \alpha_1^1$ will

correspond to a value of α_1 at which $\partial F / \partial \alpha_1 = 0$; thus the value α_1^1 minimizes F with respect to α_1 . Now the parameters α_1^1 and α_3^0 are held fixed and we calculate for different values of α_2 belonging to the interval $[\alpha_2^0 - \delta \alpha_2^0, \alpha_2^0 + \delta \alpha_2^0]$, $\delta \alpha_2^0$ is given, the objective function $F(\alpha_1^1, \alpha_2, \alpha_3^0)$. Let us plot $F(\alpha_1^1, \alpha_2, \alpha_3^0)$ as a function of α_2 alone. A minimum point of this curve, say at $\alpha_2 = \alpha_2^1$, minimizes F with respect to α_2 . Now the parameters α_1^1 and α_2^1 are held fixed and we calculate for different values of α_3 belonging to the interval $[\alpha_3^0 - \delta \alpha_3^0, \alpha_3^0 + \delta \alpha_3^0]$, $\delta \alpha_3^0$ is given, the objective function $F(\alpha_1^1, \alpha_2^1, \alpha_3)$. Let us plot $F(\alpha_1^1, \alpha_2^1, \alpha_3)$ as a function of α_3 alone. From this graph we can choose $\alpha_3 = \alpha_3^1$ such that minimizes F with respect to α_3 . All these actions form the first step and at the point $(\alpha_1^1, \alpha_2^1, \alpha_3^1)$ the objective function $F(\alpha_1, \alpha_2, \alpha_3)$ has been reduced but probably not yet minimized. With the new guess α_1^1 , α_2^1 and α_3^1 for the parameters, we repeat the above step to obtain the point $(\alpha_1^2, \alpha_2^2, \alpha_3^2)$ and so on until the relative changes in the parameters are small enough to claim convergence. Some useful convergence properties of this Gauss Sidel interactive minimizer are given by Elkin [17] . Sometimes the convergence of this process is slow: however it gives always a global view of the behaviour of the objective function with respect to the parameters.

b) The difficulty in ecosystem modeling is that the systems are so complex depending on so many parameters, that to treat them as a whole is a very awkward problem. The techniques of the multicompartmental analysis are very useful because they allow to construct a prototype model as a sequence of ecological models of increasing complexity. Generally, each of these models is described by a set of ordinary differential equations. By observing the difference between the compartmental evolution in the model and that in the system as determined by experiments, it is possible to identify the more significative parameters of the model. These models are utilized for "reducing" the experimental observations into a few parameters in order to simulate the behaviour of the system even in conditions not already experimented.

The interactive system described has been used to formulate some mathematical models for mass transfer in aquatic ecosystems as those deve-

loped in [18] . The transfer of a specific radionuclide in microecosystems in which biotic and abiotic complexity may be varied, has been analyzed. In this case a sequence of models may be formulated increasing the complexity of the system by increasing the number of system elements, the number of connections between them and the complexity of transfer functions (continuous linear, non linear, discrete).

c) An other use of the above interactive system was related to elaborate different variations of the Lotka-Volterra model which describes the dynamical behaviour of predator-prey populations in a closed ecosystem. Many improvements on this model have been suggested for the purpose of increasing its realism to reproduce population trends by fitting in the best way the measured values of the population-sizes available in a fixed period. The rationale for the construction of such a model is to propose initially between the predator species and the prey species the interaction mechanism of Lotka-Volterra. If this model reproduces well the available experimental observations, it may be considered a satisfactory description of the population trends; otherwise some deficiencies of this model must be corrected. A deficiency is the non existence of a saturation level of the prey species in the absence of the predator species: a population in a limited space cannot exceed some saturation level. An other deficiency is the non existence of a saturation effect due to the limited appetite of the predator species. An other deficiency is the non existence of a time-lag between a prey kill and a predator birth and a seasonal variation of the populations. An other deficiency is the non existence of an age-specificity. This age-specificity is particularly important because of the different vulnerability of the various age groups of prey (very young, adults, etc.) to predation.⁽⁺⁾

Each elaboration to the initial Lotka-Volterra model requires to identify the parameters which characterize the new model: this problem is solved by the interactive system with the Gauss Sidel minimizing process described in a).

(+) - When we correct this deficiency in the original model, we have to change not only the structure of the differential equations (as for the other corrections) but also the number of these.

REFERENCES

- [1] Sage A.P., Melsa J.L. : System Identification. Academic Press, New York (1971).
- [2] Karplus W.J. : System Identification and Simulation. A pattern Recognition Approach. Comp. Sc. Dept. Report, Univ. of California (1971), Los Angeles.
- [3] Smith L.B. : An example of a pragmatic approach to portable interactive graphics. Comput. & Graphics 1 (1975), 49-53.
- [4] Dalle Rive L., Merli C. : FALCON-A conversational polyalgorithm for ordinary differential equation problems. Information Processing 74, North-Holland Publ. Comp., Amsterdam (1974).
- [5] Edsberg L. : Integration Package for Chemical Kinetics. Stiff Differential Systems (ed. R.A. Willoughby). Plenum Press, New York (1974).
- [6] Stetter H.J. : Local estimation of the global discretization error. SIAM J. Numer. Anal. 8 (1971), 512-523.
- [7] Lawson J.D. : Generalized Runge-Kutta processes for stable systems with large Lipschitz constants. SIAM J. Numer. Anal. 4 (1967), 372-380.
- [8] Ralston A. : Runge-Kutta methods with minimum error bounds. Math. Comput. 16 (1962), 431-437.
- [9] Lapidus L., Seinfeld J.H. : Numerical Solution of Ordinary Differential Equations, Academic Press, New York (1971)
- [10] Dahlquist G. : Stability and error bounds in the numerical integration of ordinary differential equations. The Royal Inst. of Technology n.130, Stockholm (1959).
- [11] Gear C.W. : Numerical Initial Value Problems in Ordinary Differential Equations. Prentice Hall, N.J. (1971).
- [12] Jain R.K. : Some A-stable methods for stiff ordinary differential equations. Math. Comput. 26 (1972), 71-78.
- [13] Hull T.E. et al. : Comparing numerical methods for ordinary differential equations. SIAM J. Numer. Anal. 9 (1972), 603-637.
- [14] Caruthers L.C., Bergeron R.D.: Device independent graphics, SEAS 1974 Session Report.

- [15] Gear C.W. : The automatic integration of ordinary differential equations. Comm. ACM 14 (1971), 176-179
- [16] Hull T.E. : Numerical solutions of initial value problems for ordinary differential equations. Numerical Solution of Boundary Value Problems for Ordinary Differential Equations (ed.A.K. Aziz), Academic Press, New York (1975).
- [17] Ortega J.M., Rheinboldt W.C. : Iterative Solution of Nonlinear Equations in Several Variables. Academic Press, New York (1970).
- [18] Argentesi F., Di Cola G., Verheyden N. : Parameter estimation of mass transfer in compartmented aquatic ecosystems. Identification and System Parameter Estimation Part 1 (ed. P. Eykhoff), North-Holland Publ. Comp., Amsterdam (1973).